

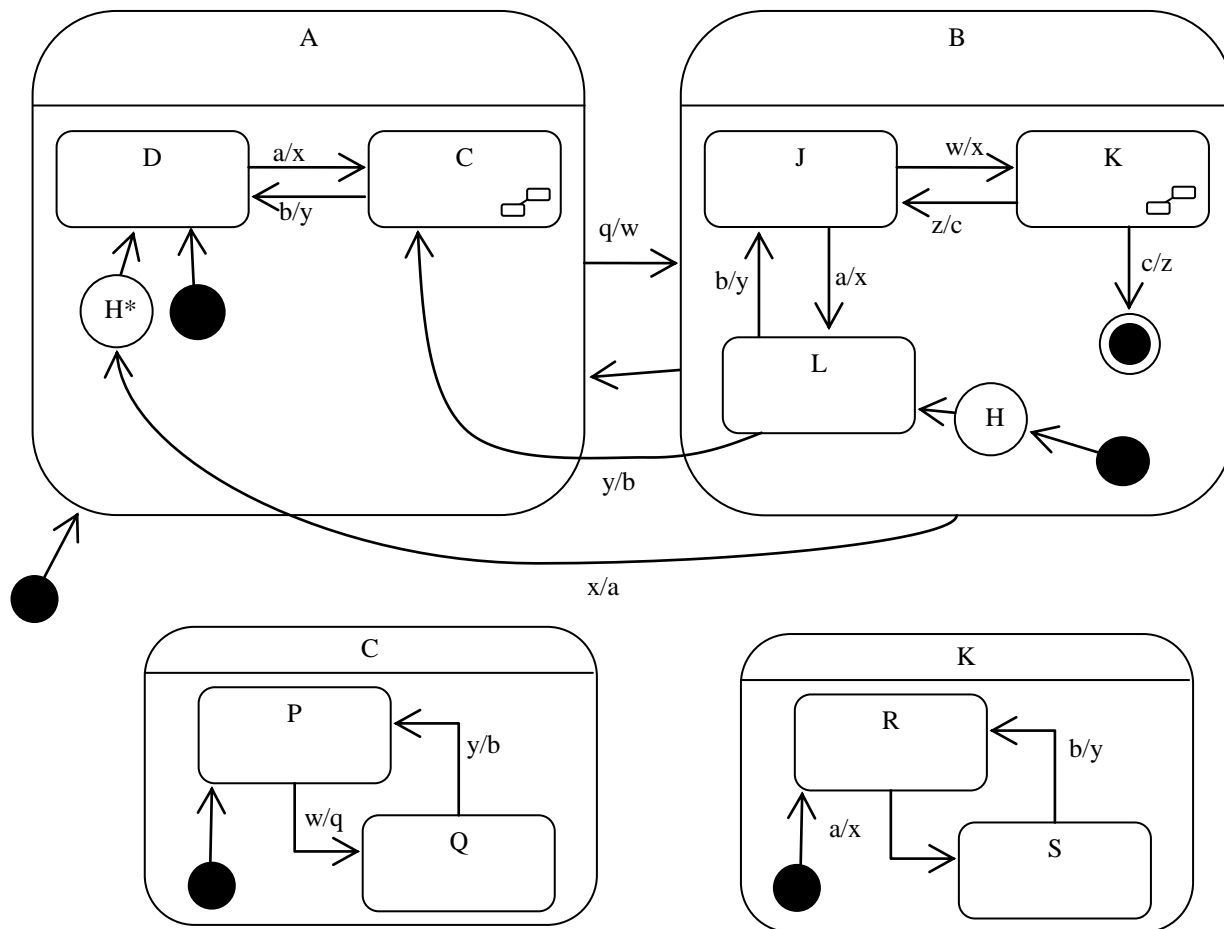
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2010. december 21.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. A következő UML állapotdiagram alapján minősítse az állításokat! Csak a rubrikába tett jelzést vesszük figyelembe! (8 pont)

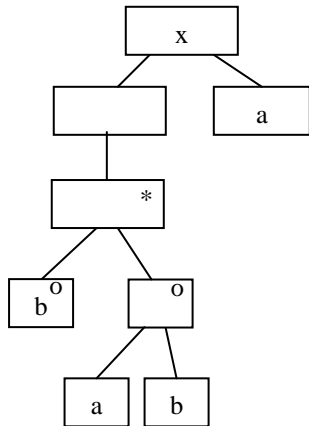


Igaz	Hamis	Állítás
<input checked="" type="checkbox"/>	<input type="checkbox"/>	L állapot után közvetlenül következhet Q állapot
<input type="checkbox"/>	<input checked="" type="checkbox"/>	C állapotból elérhető egy lépésben S
<input checked="" type="checkbox"/>	<input type="checkbox"/>	K állapotból csak „c” és „x” esemény hatására léphetünk át A állapotba
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q állapotból „y” esemény hatására átlépünk D-be
<input type="checkbox"/>	<input checked="" type="checkbox"/>	L állapot után csak C és J következhet egy lépésben

A kezdés után az **a, w, q, b, x** esemény-szekvencia hatására

Igaz	Hamis	Állítás
<input type="checkbox"/>	<input checked="" type="checkbox"/>	pontosan kétszer fut le a „q” tevékenység
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Q állapotba kerülünk
<input type="checkbox"/>	<input checked="" type="checkbox"/>	érintettük az K állapotot

2. Adott az alábbi Jackson-ábra.



```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x          ①      >
  <!ELEMENT a          (#PCDATA)>
  <!ELEMENT b          (#PCDATA)>
]>
<x>
  <a>a</a>
  <b>b</b>
  <b>-x</b>
  <b>x</b>
  <a>-x</a>
</x>
```

- Jelölje meg, hogy melyik elv jelenik meg a szerkezetben egynél többször ! (2 pont)
- szekvencia (sorrend)
 - szelekció (választás)
 - iteráció (ismétlődés)
 - egyik sem

- Az alábbiak közül ① helyén melyik válasz írja le helyesen a Jackson-ábrát ? (4 pont)
- $(a^*, b \mid (a, b))$
 - $(((a, b) \mid b)^*, a)$
 - $(b^*, (a, b) \mid a)$
 - $((b, (a, b)^*), a)$
 - $((b \mid (a, b))^*, a)$
 - $((b \mid (a, b)^*), a)$
 - $(((a, b) \mid b^*), a)$
 - egyik sem

- A baloldalon álló XML adatszerkezet (2 pont)
- mint XML szerkezet szintaktikai hibás
 - megfelel a Jackson-ábrának (érvényes)
 - nem felel meg a Jackson-ábrának (érvénytelen)
 - érvényessége nem dönthető el

3. Kockázatelemzés (risk analysis) során minden kockázathoz hozzárendeljük (4 pont)

probability **seriousness (effect)**

4. Egy osztály megvalósít egy absztrakt adatstruktúrát. Az osztály metódusai az adatstruktúra műveletei. Milyen a metódusok közötti kohézió ? (2 pont)

- funkcionális
- strukturális
- kommunikációs
- procedurális
- temporális
- egyik sem, hanem

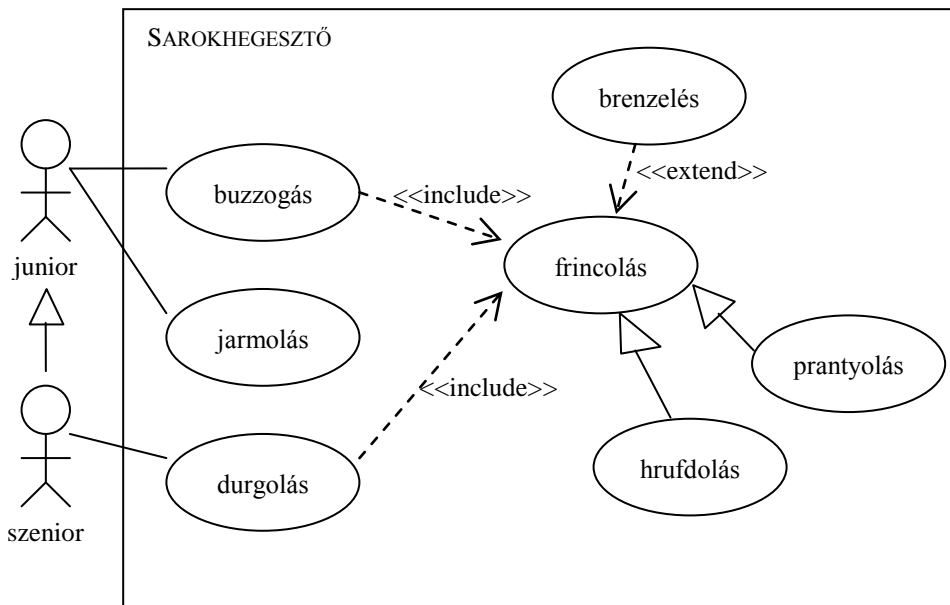
Egy egyszerű társalgó (chat) program úgy működik, hogy a bejelentkezett felhasználó által küldött üzeneteket a rendszer minden másik bejelentkezett felhasználónak továbbítja. Milyen a szoftver architektúrája ? (2 pont)

event based implicit invocation (eseményvezérelt) ..

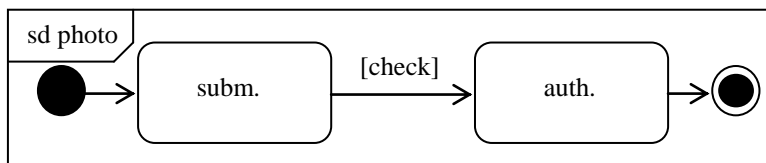
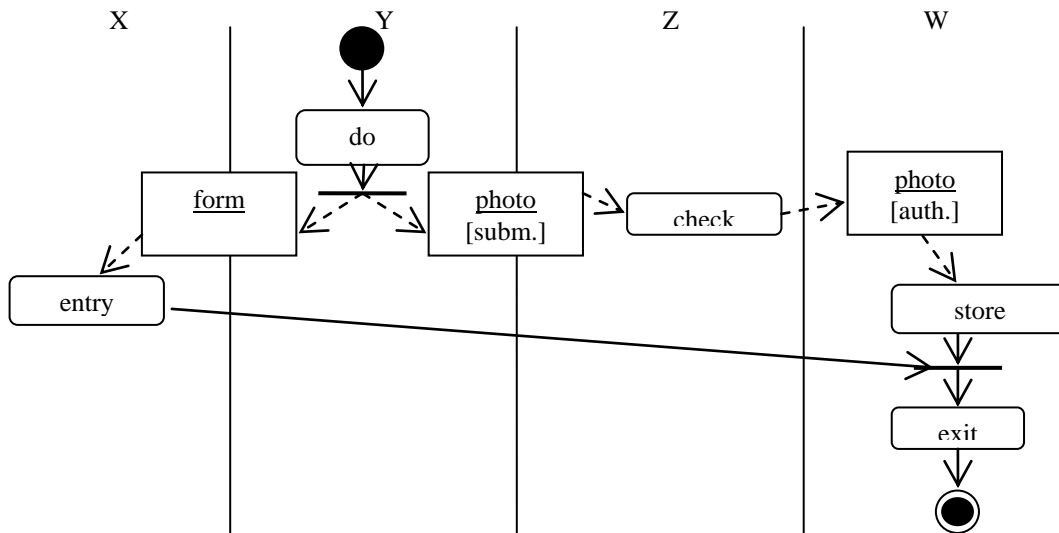
A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Készítsen UML2 use-case diagramot az alábbi leírás alapján! (7 pont)

A sarokhegesztővel a junior furga buzzogni, és jarmolni tud. A buzzogáshoz be kell kapcsolni a frincolást. Ennek két módja van: a prantolás és a hrufdolás. A herkentyű hibás beállítások esetén frincolás közben brenzel is kicsit. A szenior furga a fentiek mellett a durgolás funkcióhoz is hozzáfér, amihez szintén be kell kapcsolni a fenti frincolás funkciót.



6. Legyen adott az alábbi – object flow-val kiegészített – aktivitás-diagram (activity diagram) ! Rajzolja meg azon objektumok UML2 state-chartját, amelyeknek az ábra alapján több állapota is van! (4 pont)

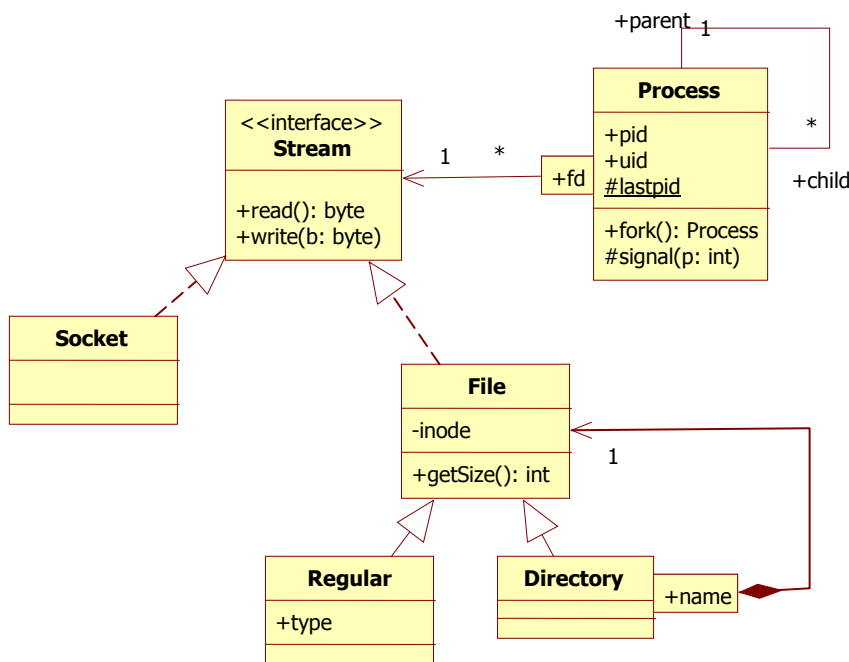


7. Definiálja, hogy mi a "konfiguráció" a konfiguráció menedzsmentben ! (3 pont)

a selection of configuration items designating a state of the project

8. Készítsen UML 2 osztálydiagramot (class diagram) az alábbi leírás alapján! Használja a kövéren szedett kifejezéseket! Ahol lehet, adja meg a paraméterek, attribútumok, stb típusát is! (8 pont)

Az OOniX operációs rendszerben **folyamatok**, **fájlok** és hálózati kapcsolatok (**socket**) vannak. A folyamatoknak van azonosítója (**pid**), tulajdonosa (**uid**). Egy folyamatból a **fork()** metódussal lehet újat létrehozni. Minden folyamat ismeri a közvetlen **őst** és a **gyerekeit**. A fájloknak van egy senki más által nem látható **inode** száma, és lekérdezhető a **méretük**. A fájlok többfélék lehetnek: **könyvtárak**, amelyek más fájlokat tartalmazhatnak (a **nevük** alapján), **reguláris** fájlok, amiknek van **típusa**, stb. Minden fájl egy könyvtár része. A folyamatok egyformán kezelhetnek socketet és fájlt is, de csak egy közös interfészt (**stream**) látnak belőlük, amiken bájtokat lehet **olvasni** és **írni**. Az ilyen objektumokról a folyamatnak van egy listája, aminek az elemeit fájlleíróval (**fd**) azonosítja. A folyamatok létrehozásakor egy (a folyamatok számára közös) számláló (**lastpid**) növekszik, ez lesz az újonnan létrehozott folyamat azonosítója. A folyamatoknak más folyamatok tudnak üzenni a **signal()** üzenet meghívásával (egy darab egész típusú paramétere van). A lastpid és a signal csak folyamatból (és esetleges leszármazottjából) látható.



9. A Rational Unified Process (RUP) követelmény szakaszában milyen típusú use-case-eket készítünk ? Mi ennek a típusú use-case-nek a fő jellemzője ? (4 pont)

high level, (some expanded), essential
technológia- és implementáció-független

Eredmények értékelése:

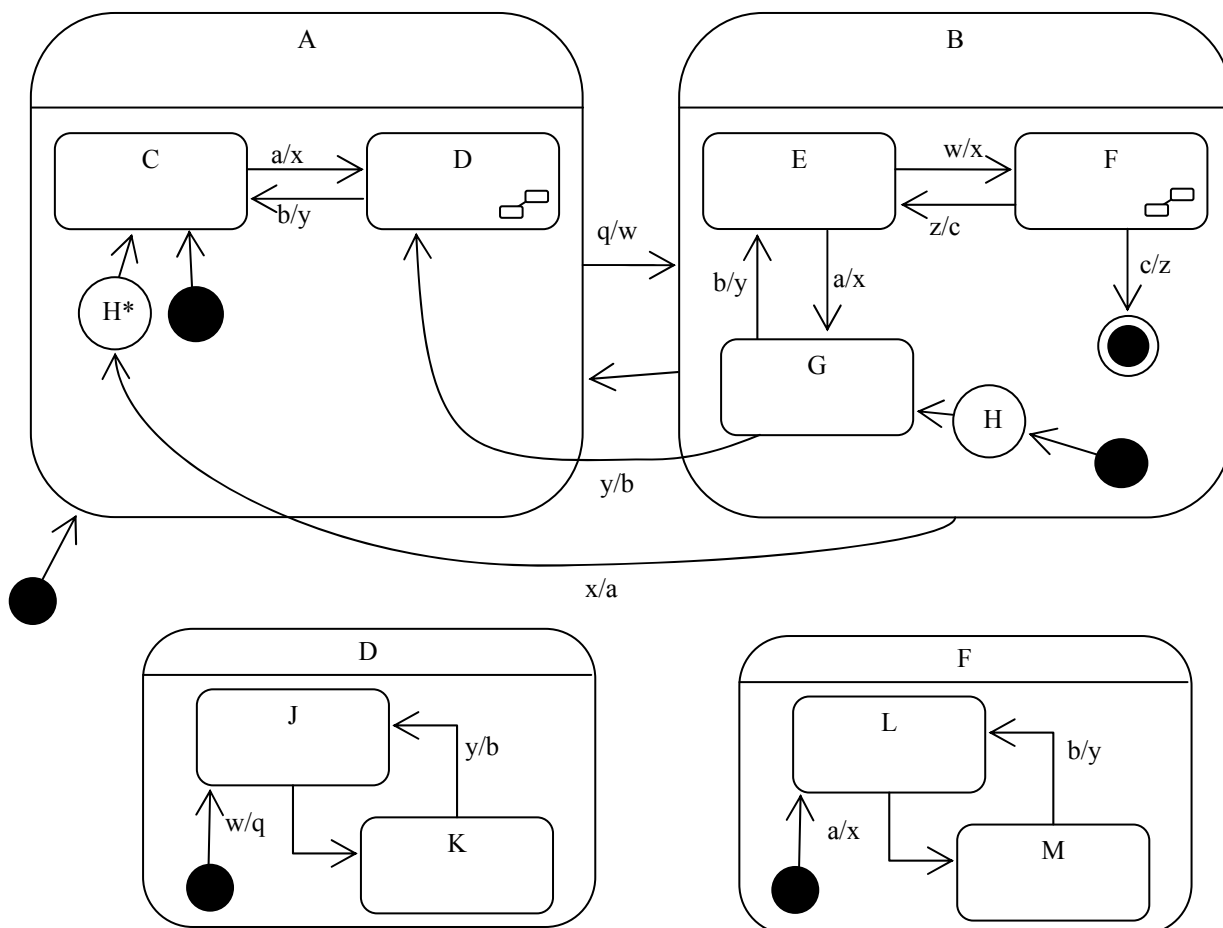
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2010. június 1.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. A következő UML állapotdiagram alapján minősítse az állításokat! Csak a rubrikába tett jelzést vesszük figyelembe! (8 pont)



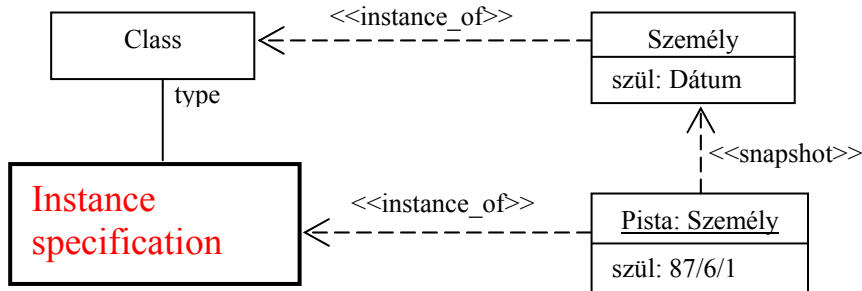
Igaz	Hamis	Állítás
<input type="checkbox"/>	<input checked="" type="checkbox"/>	G állapot után csak D és E következhet egy lépésben
<input type="checkbox"/>	<input checked="" type="checkbox"/>	D állapotból elérhető egy lépésben M
<input checked="" type="checkbox"/>	<input type="checkbox"/>	G állapot után közvetlenül következhet K állapot
<input type="checkbox"/>	<input checked="" type="checkbox"/>	K állapotból „y” esemény hatására átlépünk C-be
<input checked="" type="checkbox"/>	<input type="checkbox"/>	F állapotból csak „c” és „x” esemény hatására léphetünk át A állapotba

A kezdés után az **a, w, q, b, x** esemény-szekvencia hatására

Igaz	Hamis	Állítás
<input checked="" type="checkbox"/>	<input type="checkbox"/>	K állapotba kerülünk
<input type="checkbox"/>	<input checked="" type="checkbox"/>	érintettük az F állapotot
<input type="checkbox"/>	<input checked="" type="checkbox"/>	pontosan kétszer fut le a „q” tevékenység

rossz válasz esetében pontlevonás, de total ≥ 0

2. Nevezze meg az alábbi diagramon vastag vonallal rajzolt UML2 elemet ! (Emlékeztető: *Pista* az UML modell eleme, nem pedig a *Személy* futási időben létrejött példánya !) (3 pont)



3. Adott az alábbi dekorált XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE a [
  <!ELEMENT a ((b,(c?, d*))|c)>
  <!ELEMENT b ANY>
  <!ELEMENT c (#PCDATA)>
  <!ELEMENT d (#PCDATA)>
]>
<a><b><a>
<b><c>b</c>
</b><c>d</c> ①
<c>a</c></a>
```

Mi állhat ① helyében, hogy az XML érvényes legyen ? (3 pont)

-
-
- </c><a/>
-
- <c/>
- értelmetlen a kérdés, mert a DTD nem jól formált

4. A mellékelt állapotgéppel specifikált programot az interpreter architektúra szerint kívánjuk megvalósítani. Az állapottábla alapján specifikálja a

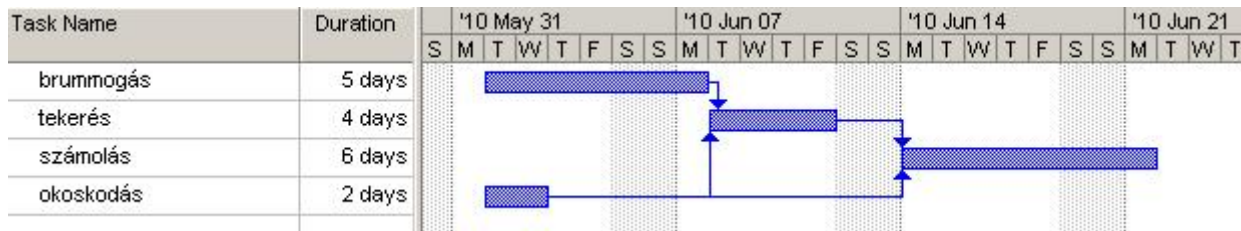
	X	Y	Z
①	③/A	③/A	②/C
②		③/B	②/C
③	①/A	③/A	

“control state” értékeit ! (2 pont): [①|②|③]

“pseudocode” utasításait ! (4 pont): ③/A, ②/C, ③/B, ①/A

5. Egy ma (június 1., kedd) induló projekt feladatait, azok hosszát és függéseit a mellékelt táblázat foglalja össze. Rajzoljon Gantt-diagramot ! (szombaton és vasárnap nem dolgozunk !) (2 pont)

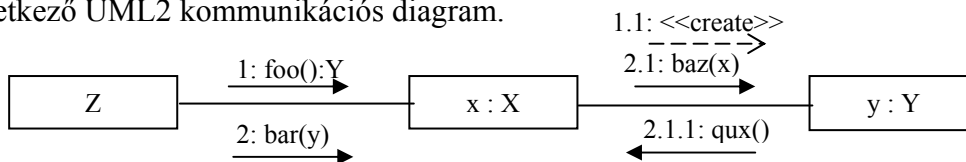
szám	feladat	nap	függés
1	brummogás	5	
2	tekerés	4	1, 4
3	számolás	6	2, 4
4	okoskodás	2	



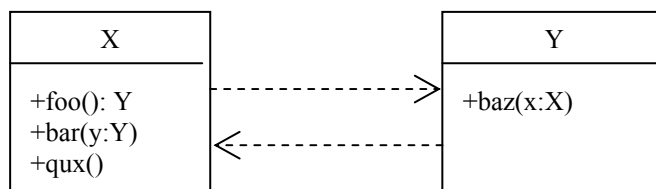
Hányadikán kezdhetjük legkésőbb az „okoskodás”-t, ha a véghatáridőt tartani akarjuk (2 pont)? jún 4.-én..

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Adott a következő UML2 kommunikációs diagram.



Feltételezve, hogy a fenti diagramon szereplő objektumok osztályainak nincsenek – a diagramból nem kiolvasható – további metódusai, közöttük nincs más egyéb kapcsolat (pl. öröklés), az alábbi ábrát korrekt UML2 osztálydiagrammá alakítva ábrázolja az osztályokat a metódusok szignatúráival együtt, valamint a két osztály közötti kapcsolatot ! (3 pont)

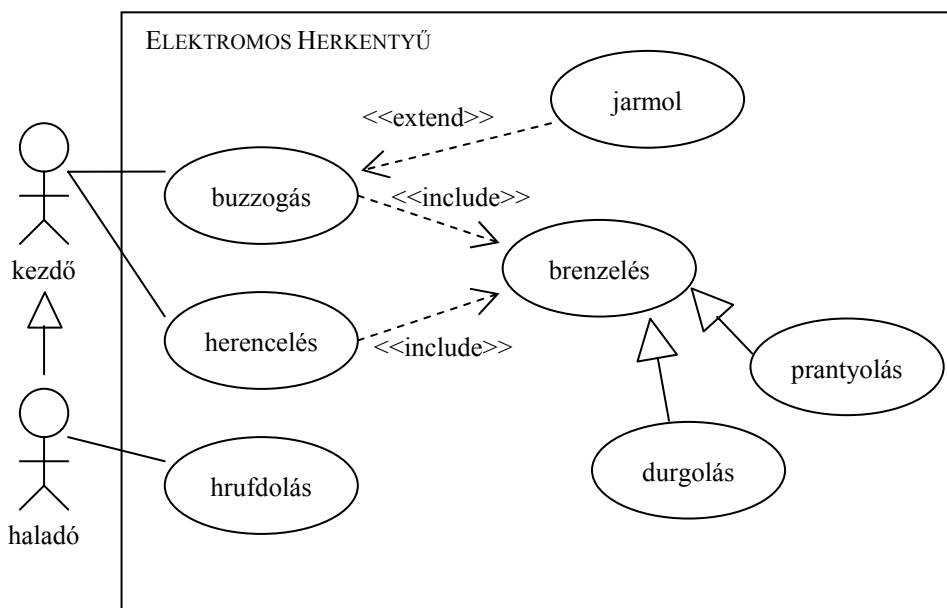


Mi a kapcsolat az *operáció* és *metódus* között ? (2 pont)

a metódus implementálja az operációt.....

7. Készítsen UML2 use-case diagramot az alábbi leírás alapján! (7 pont)

Az elektromos herkentyűn a kezdő kukor buzzogni, és herencelni tud. Mindkettőhöz be kell kapcsolni a brenzelést. Ennek két módja van: prantnyolással vagy durgolással. A herkentyű hibás beállítások esetén buzzogás közben jarmol is kicsit. A haladó kukor a fentiek mellett a hrufdolás funkcióhoz is hozzáfér.

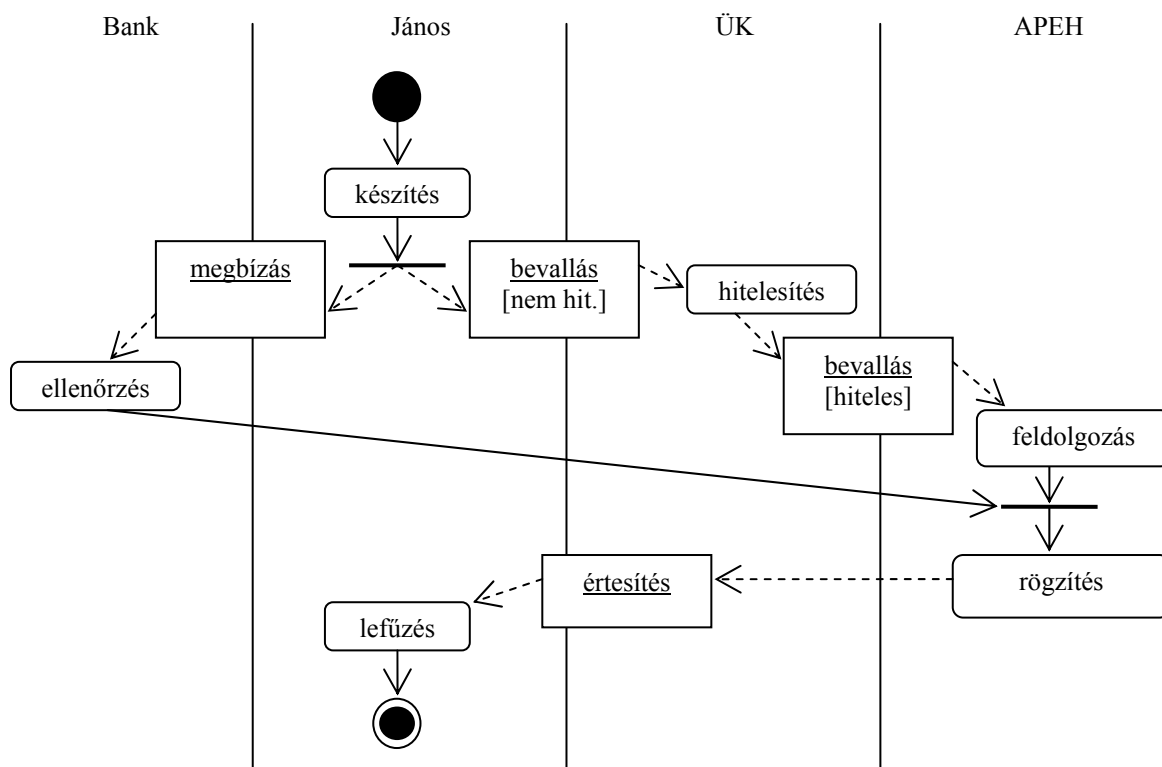


8. Mi a “fan-out”? (3 pont)

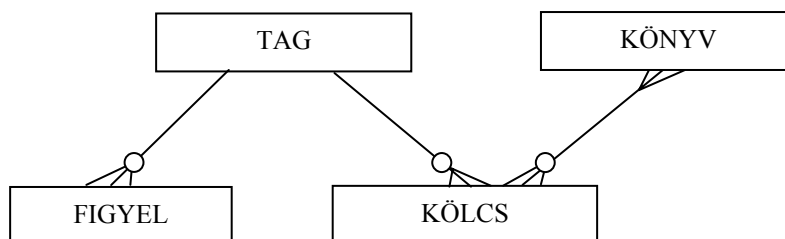
- egy adott modul (osztály) döntési hatáskörébe tartozó modulok (osztályok) száma
- egy adott modulban (osztályban) használt paraméterek száma
- egy adott modul (osztály) által használt modulok (osztályok) száma
- egy adott modul (osztály) vezérlési hatáskörébe tartozó modulok (osztályok) száma
- egy adott modult (osztályt) használó más modulok (osztályok) száma

9. Készítsen UML 2 aktivitás-diagramot (activity diagram) az alábbi leírás alapján! Jelölje az action-object flow-t is! Használja a kövéren szedett kifejezéseket! (8 pont)

Regenkurt **János** vállalkozó adóbevallást **készít**. A **bevallást** feltölti az ügyfélkapura (**ÜK**), és ezzel párhuzamosan a **bankjánál** átutalási **megbízást** ad az adóhátralék befizetésére. Az ügyfélkapu **hitelesíti** az adóbevallást, és továbbküldi az **APEH**-nek, ahol **feldolgozzák**. A bank **ellenőrzi** az átutalási megbízást, majd a pénzt átutalja az APEH-nek. Amikor az APEH feldolgozta a bevallást és megkapta az átutalást, akkor **rögzíti** az állapotot, és **értesítést** küld Jánosnak, hogy minden rendben. Ezt az értesítést János **lefűzi**. Ezzel az adóbevallás véget ér.



10. A könyvtári tagokkal (TAG) kapcsolatosan tároljuk minden egyes kölcsönzés adatát (KÖLCS), valamint a tagnak késedelem esetén (SMS-ben, levélben, e-mailben) küldött figyelmeztetést (FIGYEL). Egy kölcsönzés alkalmával több könyvet (KÖNYV) is kölcsönözhet a tag. A kölcsönzésre vonatkozó adatot a könyv visszavétele után 60 nappal töröljük, de a figyelmeztetési adatokat megőrizzük a tag törléséig. Jelölje a multiplicitást és opcionálitást az alábbi ER diagramon ! A csirkeláb és karika jelölést használja ! (3 pont)



Eredmények értékelése:

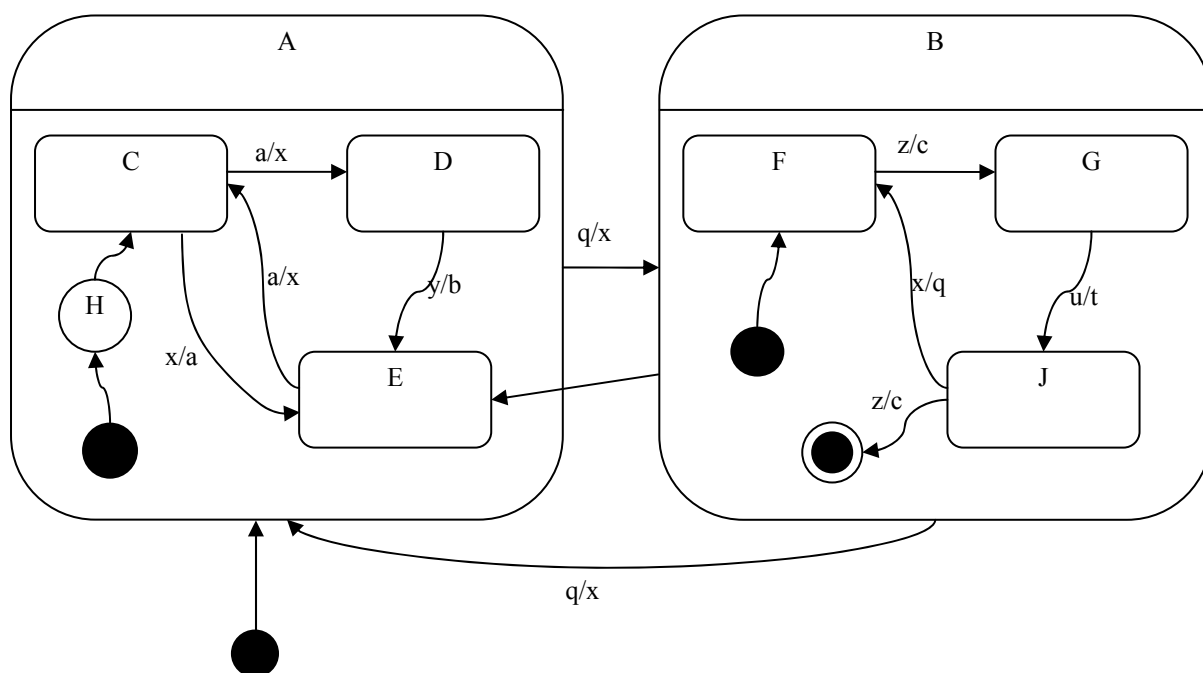
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2010. május 26.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. A következő UML2 állapotdiagram alapján minősítse az állításokat! (8pont)



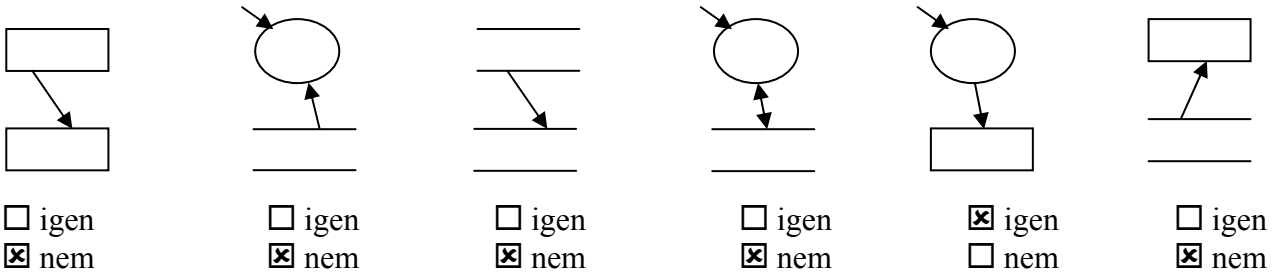
Iga z	Hamis	Állítás
<input checked="" type="checkbox"/>	<input type="checkbox"/>	D állapotból 2 lépésben visszaérhet D-be
<input type="checkbox"/>	<input checked="" type="checkbox"/>	F állapotból „q” esemény hatására H állapotba kerül
<input checked="" type="checkbox"/>	<input type="checkbox"/>	B-ből A-ba való váltáskor végrehajtható a „c” tevékenység
<input type="checkbox"/>	<input checked="" type="checkbox"/>	E állapotból egyetlen esemény hatására csak a C állapot következhet
<input checked="" type="checkbox"/>	<input type="checkbox"/>	J állapotból egyetlen esemény hatására E állapot következhet

A kezdés után az **x, q, z, q** esemény-szekvencia hatására

Iga z	Hamis	Állítás
<input type="checkbox"/>	<input checked="" type="checkbox"/>	C állapotba kerülünk
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Kétszer lefut az „x” tevékenység
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Érintjük a J állapotot

rossz válasz esetében pontlevonás, de total ≥ 0

2. Adja meg, hogy helyesek-e a következő adatfolyamokra illetve context diagram részletek ! (6 pont)



rossz válasz esetében pontlevonás, de total >= 0

3. Készítsen az alábbi DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben pontosan egy darab j elem van ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

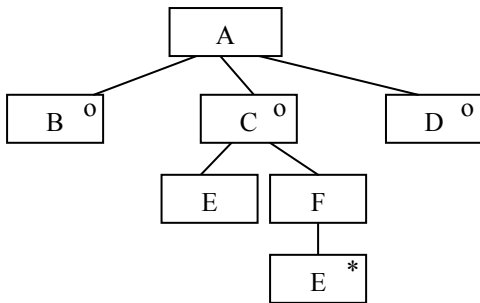
```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE f [
  <!ELEMENT j (#PCDATA)>
  <!ELEMENT h (#PCDATA)>
  <!ELEMENT g (f, j)>
  <!ELEMENT f (g | h)>
]>
    
```

```

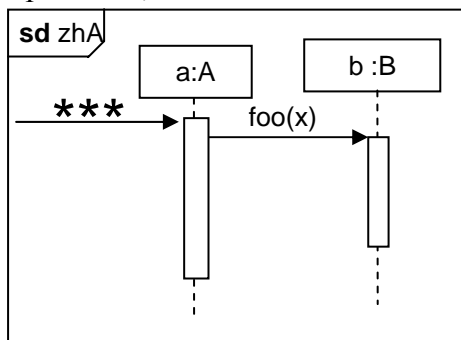
<f>
  <g>
    <f>
      <h>hhh</h>
    </f>
    <j>jjj</j>
  </g>
</f>
    
```

4. Az alábbi JSD ELH diagram alapján készítsen állapottáblát ! Az állapotokat a bevezetett jelölésrendszer szerint, számokkal jelölje ! Az ① legyen az induló állapot ! (4 pont)

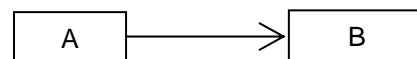


	B	D	E		
①	③	③	②		
②			②		
③					

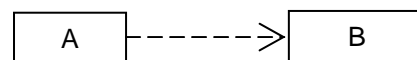
5. Tételezze fel, hogy az alábbi (zhA nevű) UML2 szekvenciadiagramon szereplő objektumok osztályai között nincs más egyéb – a diagramból nem kiolvasható – kapcsolat (pl. öröklés) ! Rajzolja be (4 pont) az A és B között kapcsolatot, ha



*** = bar(x)



*** = qwx(b)



A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

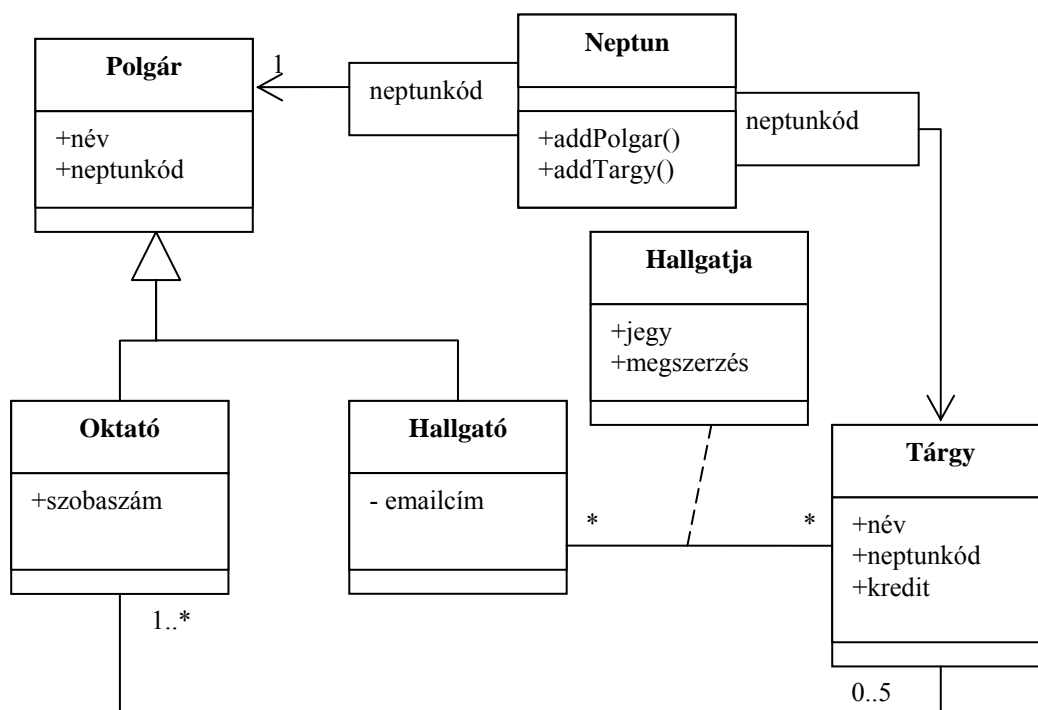
6. Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe ? (3 pont)

komponensek: **eseményforrás, eseményfigyelő**

szerepek: **az eseményfigyelők beregisztrálják magukat a forrásoknál, akik náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek**

7. Rajzoljon UML2 osztálydiagramot az alábbi történet alapján! Jelölje a számosságokat is! (8 pont)

Az egyetemi polgárokat, akiknek nyilvános a neve és a neptunkódja, a Neptun rendszer tartja nyilván, mégpedig a polgáronként egyedi neptunkód alapján. Polgár az oktató és a hallgató is. Az oktatónak nyilvántartjuk a szobaszámát, a hallgatónak az emailcímét (privát adat). A Neptunban tároljuk a tárgyakat is egyedi neptunkóddal. A tárgyaknak szintén ismerjük a nevét és neptunkódját, valamint az érték kapható kreditek számát. Egy hallgató több tárgyat is felvehet (hallgatja), egy tárgyra több hallgató is járhat. Ezen kívül egy adott hallgató egy adott tárgyra kapott jegyét és a megszerzés évét is nyilvántartjuk. Egy tárgyat legalább egy oktató oktat, egy oktatónak pedig lehet több tárgya is (maximum 5), de van akinek egy sincs. A Neptunba fel tudunk venni új tárgyat és polgárt.



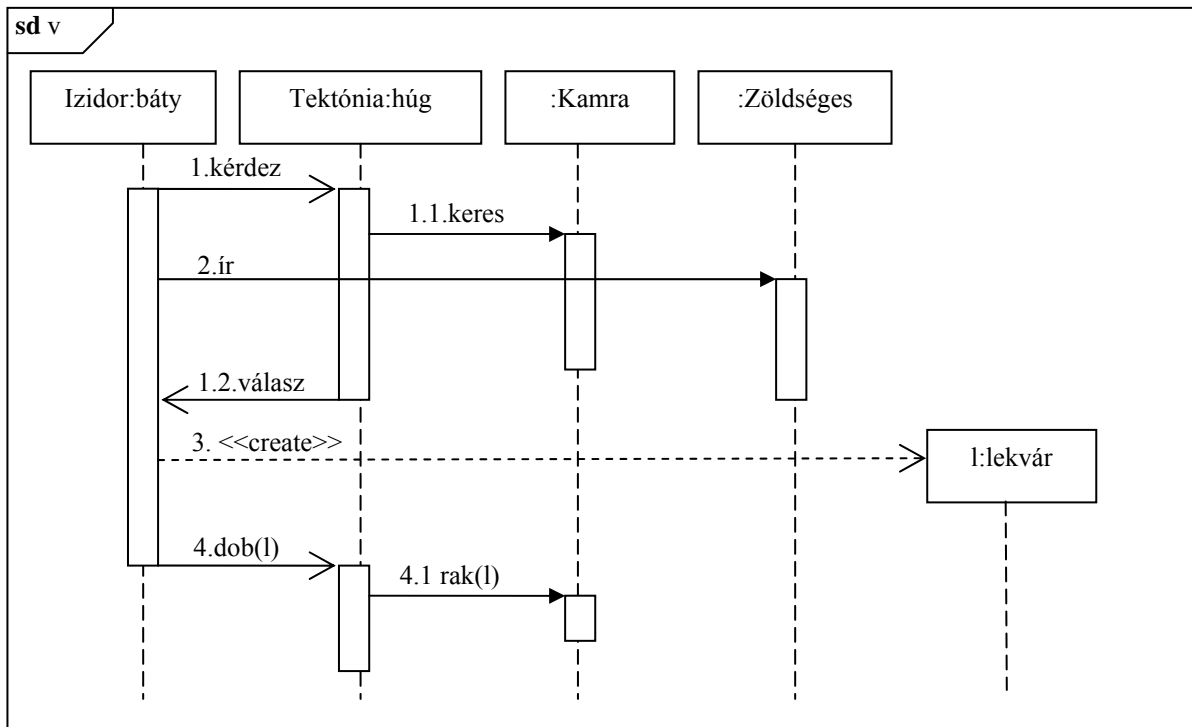
8. Jelölje az állítások mellett 1-5-ig, hogy minimálisan melyik CMM szinttől igazak ! Ha az állítás nem értelmezhető, akkor tegyen X-et ! (4 pont)

5	Számszerű visszacsatolás segíti az állandó folyamatfejlesztést
X	A termék életciklusának tervezése beépült a folyamatfejlesztés részfeladatai közé
5	Innovatív ötletek és technológiák segítik az állandó folyamatfejlesztést
X	A vezetők képesek a termék minőségének közvetlen ellenőrzésére

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

9. Készítsen UML2 szekvencia-diagramot az alábbi történet alapján! Ne feledkezzen el a hierarchikus számozásról sem! (7 pont)

Izidor a zöldségesnél rájön, hogy lekvárt akar főzni, ezért SMS-ben megkérdi húgát, Tektóniát, hogy van-e otthon befőző cukor. Tektónia átkutatja a kamrát, és talál cukrot, amiről (szintén SMS-ben) értesíti bátyját. Izidor eközben vicces szöveget ír a zöldséges hátára, amíg a választ meg nem kapja, majd hazamegy, és megfőzi a lekvárt. A kész lekvárt választ sem várva odadobja húgának, és elsiet. Tektónia a lekvárt beteszi a kamrába.



10. A RUP (Rational Unified Process) egyik munkafolyamatában (workflow) szerződés (contract) készítését javasolja illetve írja elő. Melyik munkafolyamatban esedékes szerződés készítése? Kik között kell szerződést készíteni? Milyen fontosabb pontjai vannak a szerződésnek? (4 pont)

Munkafolyamat (workflow): **analízis**

Szerződő felek: **operációk és az operációk felhasználói**

Szerződés fontosabb pontjai: **Responsibilities, Pre-conditions, Post-condition, Types, Crossrefs,**

Exceptions, Output

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

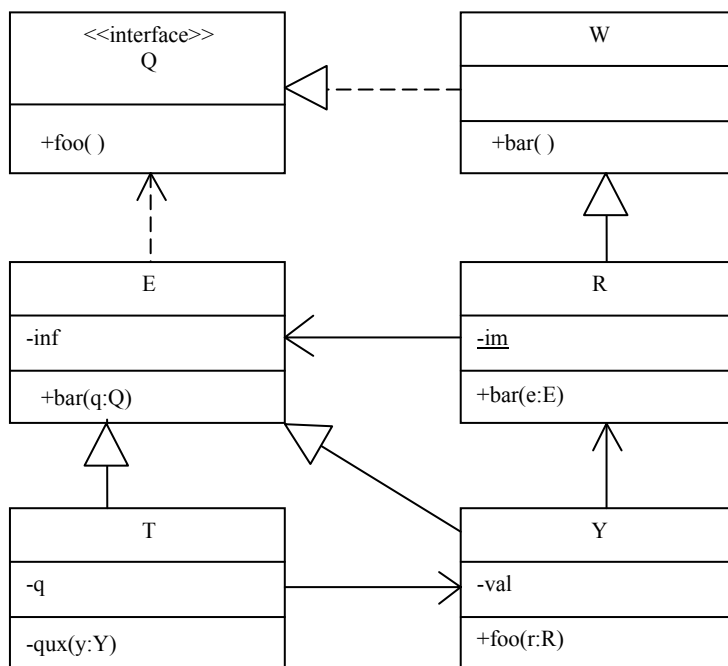
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2010. január 26. 12:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)

- [C] Y **bar(q:Q)** metódusa kaphat paraméterül R objektumot, mert Y függ R-től.
- [B] T **qux(y:Y)** metódusa módosíthatja a paraméter **val** attribútumát, mert a metódus privát.
- [E] E bárhol helyettesíthető R-rel, mert azonos az interfészük.
- [C] Y **foo(r:R)** metódusa nem módosíthatja a paraméter **im** attribútumát, mert az attribútum statikus.
- [E] E **bar(q:Q)** metódusa kaphat E objektumot paraméterül, mert az E megvalósítja a Q interfészt.
- [E] E **bar(q:Q)** metódusa nem hívhatja meg egy paraméterül kapott W **foo()** metódusát, mert W-nek nincs ilyen szignatúrájú metódusa.
- [B] R **bar(e:E)** metódusa nem kaphat paraméterül Y objektumot, mert az Y-R asszociációban csak Y hívhatja R-t.
- [B] R nem valósítja meg a Q interfészt, mert van olyan szignatúrájú metódusa, ami nem szerepel a Q metódusai között.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Jelölje, hogy a szoftverfejlesztés „spirális modellje”-nek egyes szektoraihoz rendelt tevékenységek mely fogalmakkal jellemezhetőek ! (4 pont) Rossz válasz esetén pontot veszít !

szektor	analízis	dialízis	tézis	szintézis	protézis
1.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Adott az alábbi dekorált XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x      (#PCDATA|b|c) ① >
  <!ELEMENT b      ANY>
  <!ELEMENT c      (#PCDATA)>
]>
<x>
  ②
  <x>januar<c>26</c>
  </x>vizsga</b><b><b/>/x</b>
</x>
```

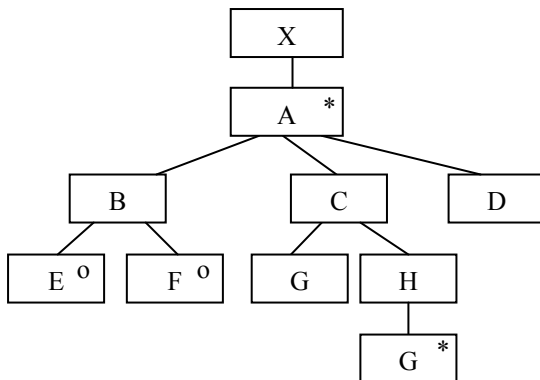
Mit írna ① helyére, hogy a DTD jól formált legyen ? (2 pont)

- semmit
- *
- +
- + vagy *
- egyéb:

Feltételezve, hogy a DTD jól formált, mi állhat ② helyében, hogy az XML érvényes legyen ? (2 pont)

- semmi
- /x
- /x
-
- x

4. Az alábbi JSD ELH diagram alapján készítsen állapottáblát ! Az állapotokat a bevezetett jelölésrendszer szerint, - az előfordulás ideje szerint növekvő - számokkal jelölje ! Az ① legyen az induló állapot ! (5 pont)



	D	E	F	G	
①		②	②		
②				③	
③	①			③	

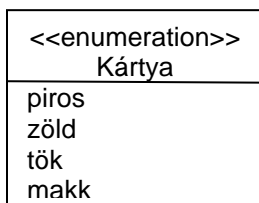
5. Sorolja fel a Rational Unified Process (RUP) életciklus modelljében szereplő „támogató munkafolyamatokat” (supporting workflows) ! (3 pont)

- konfigurációs menedzsment
- menedzsment
- környezet

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

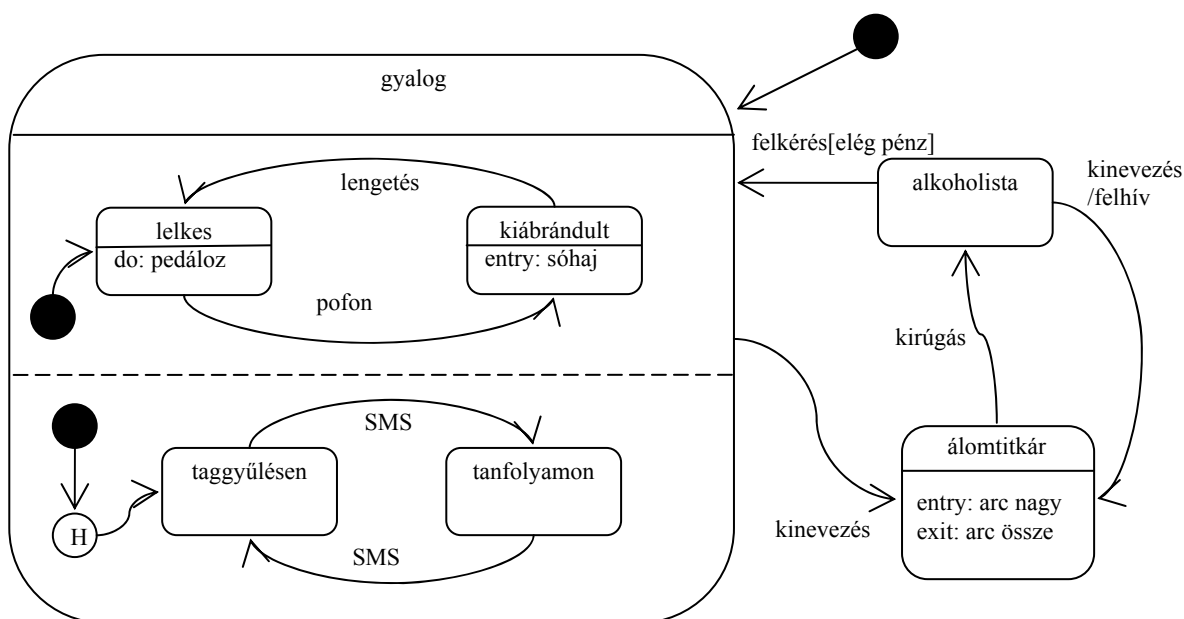
6. Definiálja UML2-ben az alábbi felsorolást ! (3 pont)

Kártya = [piros | zöld | tök | makk]



7. Rajzoljon UML 2.0 állapotábrát (state chart) az alábbi történet alapján! (7 pont)

A Stupiditas nevű szervezet tagja *gyalogként* (másként *paraszt*) kezdi pályafutását. Először nagyon *lelkes*, ilyenkor folyton pedálozik. Ha nagy pofont kap, *kiábrándul* (és elsóhajtja magát). Némi állami támogatás belengetésével ismét lelkes lesz (és pedálozik). Mindeközben (vagyis hogy éppen lelkes vagy kiábrándult), csak két összejövetelen lehet megtalálni: *taggyűlésen* és *tanfolyamon* (a taggyűlés az első). Ha az egyiket SMS-t kap, átmegy a másikra. Mikor kinevezik *álomtitkárnak*, akkor maga mögött hagyja a gyalogos életet (hiszen nagy fekete autót is kap). Álomtitkárként először az *arca* lesz nagy. Amikor kirúgják „állásából”, az *arca* összemegy és *alkoholista* lesz. Ekkor felkérésre, ha elég pénzt kap (az alkoholizmust levetkőzve) ismét *gyalog* lesz. Itt mindenképpen lelkesen azon az összejövetelen folytatja, ahol utoljára *gyalogként* megfordult. Az alkoholizmusból egy újabb álomtitkári kinevezés is kigyógyítja. Ilyenkor felhívja anyukáját.



8. Adja meg, hogy az alább felsorolt célok a JSD melyik lépését jellemzik ! Ha egy célhoz nem rendelhető JSD lépés, akkor azt „NEM”-mel jelölje ! (4 pont) Rossz válasz esetén pontot veszít !

Az ELH-kat implementáló állapotgépek specifikálása

NEM

A modell-processzek hálózatának bővítése funkció processzekkel

function

Az ELH-k alapján processz-hálózat felvétele

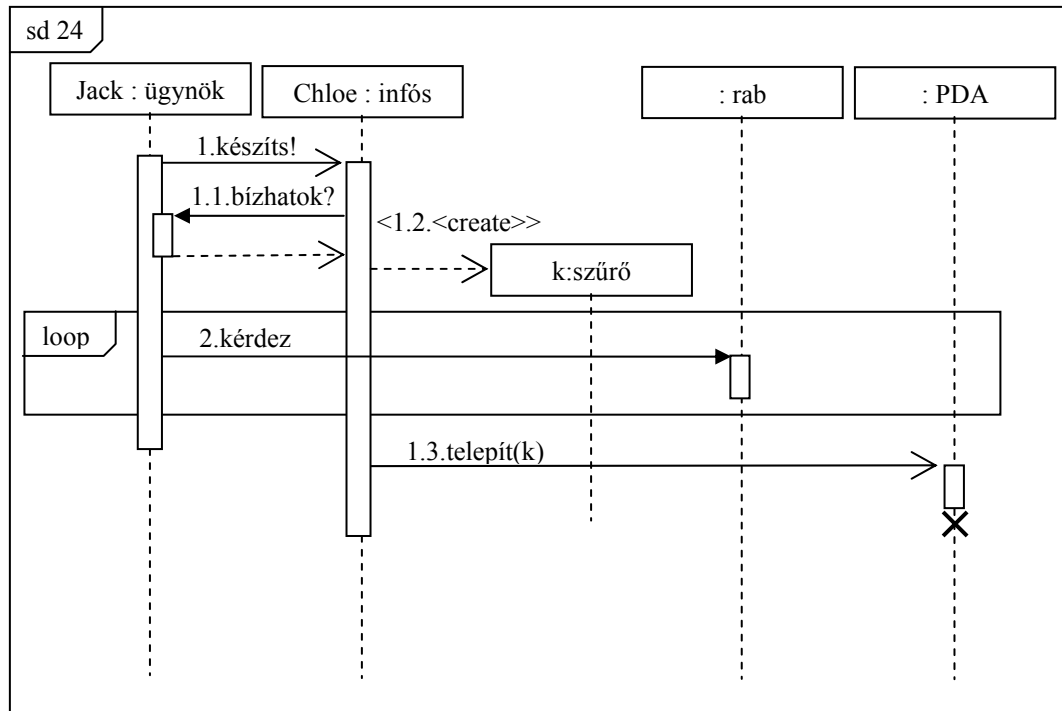
kezdeti modell

Az entitások funkcióinak implementálása

NEM

9. Készítsen UML2 szekvencia-diagramot az alábbi történet alapján! Ne feledkezzen el a hierarchikus számozásról sem! (8 pont)

Jack Bauer, a terroristák veszedelme visszatér. Információra van szüksége, ezért készített egy Kalman-szűrőt Chloe-val. Chloe, mielőtt nekilátna, megkérdi Jack-et, hogy megbízhat-e benne, majd elkezd legyártani a szoftvert. Közben Jack az egyik rabot kihallgatja, és többször is megkérdezi, hogy hol vannak a fegyverek. A kihallgatás végére készül el a szűrő, amit Chloe elkezd feltelepíteni Jack PDA-jára, de közben másra is figyel. A PDA a telepítés hatására tönkremegy.



10. A Rational Unified Process (RUP) tervezési szakaszában milyen típusú use-case-eket készítünk ? Mi ennek a típusú use-case-nek a fő jellemzője ? (4 pont)

valóságos (real)
a felhasználói felület elemeire hivatkozik

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

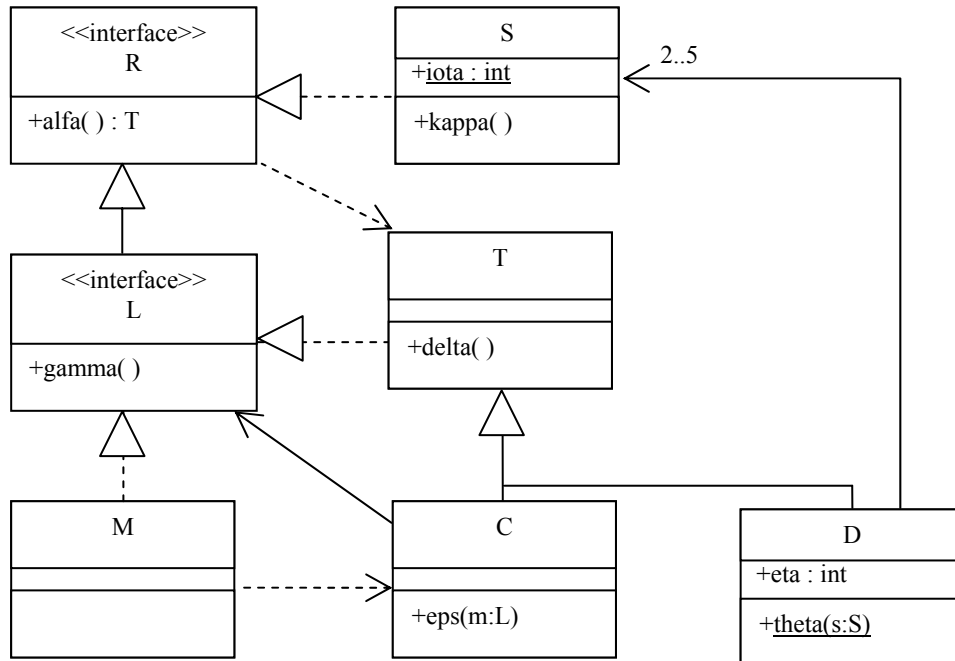
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2010. január 12. 13:30

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [A] **M** **alfa():T** metódusa visszaadhat **C** objektumot, mert **C** függ **M**-től.
- [C] **D** **theta(s:S)** metódusa nem kaphat paraméterül **C** objektumot, mert **S** és **C** is megvalósítja az **R** interfészt.
- [E] **C** **eps(m:L)** metódusa nem hívhatja meg a paraméter **gamma()** metódusát, mert az utóbbi metódus protected láthatóságú.
- [B] **D** **theta(s:S)** metódusa nem módosíthatja a paraméter **iota** attribútumát, mert a **theta(s:S)** statikus.
- [E] **S** nem valósítja meg az **alfa():T** szignatúrájú metódust, mert **S** nem függ **T**-től.
- [B] **D** **theta(s:S)** metódusa legfeljebb 5-ször hívható meg, mert **D** objektum legfeljebb 5 **S**-sel állhat asszociációban.
- [B] **M** bárhol helyettesíthető **C**-vel, mert mindketten megvalósítják az **R** interfészt.
- [D] **T** osztálynak van **alfa():T** szignatúrájú metódusa, mert **T** megvalósítja az **R** interfészt.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. A C osztály megvalósít egy adatszerkezetet (pl. stack). C metódusai az adatszerkezeten értelmezett műveleteket realizálják (pl. push, pop, empty). Milyen típusú a kohézió C osztályon belül ? (3 pont)

kommunikációs

3. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

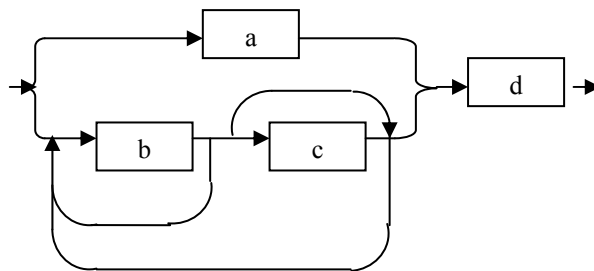
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a|(b+,c?)+), d)>
  <!ATTLIST b d CDATA #REQUIRED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
  <!ELEMENT d (#PCDATA)>
]>
<x>
  <b d="nem d">bbbb</b>
  <d><![CDATA[</d>
  <b> bbbb </b>
  <d> ]]></d>
</x>
```

- IGEN
- NEM

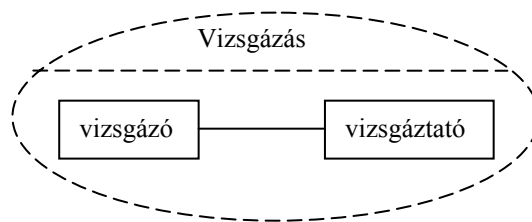
Hiba ? :

.....

Rajzolja fel a DTD-vel definiált adatszerkezet szintaxis gráfját. A DTD esetleges hibáját hagyja figyelmen kívül ! (3 pont)



4. A **Vizsgálás** funkció a **vizsgáló** és a **vizsgáztató** szerepeket megvalósító objektumok együttműködéseként valósul meg. Rajzoljon UML2 kollaborációt (collaboration) a szerepek feltüntetésével ! (3 pont)



5. A Rational Unified Process (RUP) követelménykezelő (Requirement) munkafolyamatában melyik UML modellt, és melyik két diagramtechnikát alkalmazzuk ? Hibás válasz pontszámcsökkentő. (5 pont)

modell: **use-case**

diagramok: **use-case diagram, szekvenciadiagram**.....

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Milyen integrációs (vagy tesztelési) stratégia esetében használunk teszt ágyat (test bed) ? (2 pont)

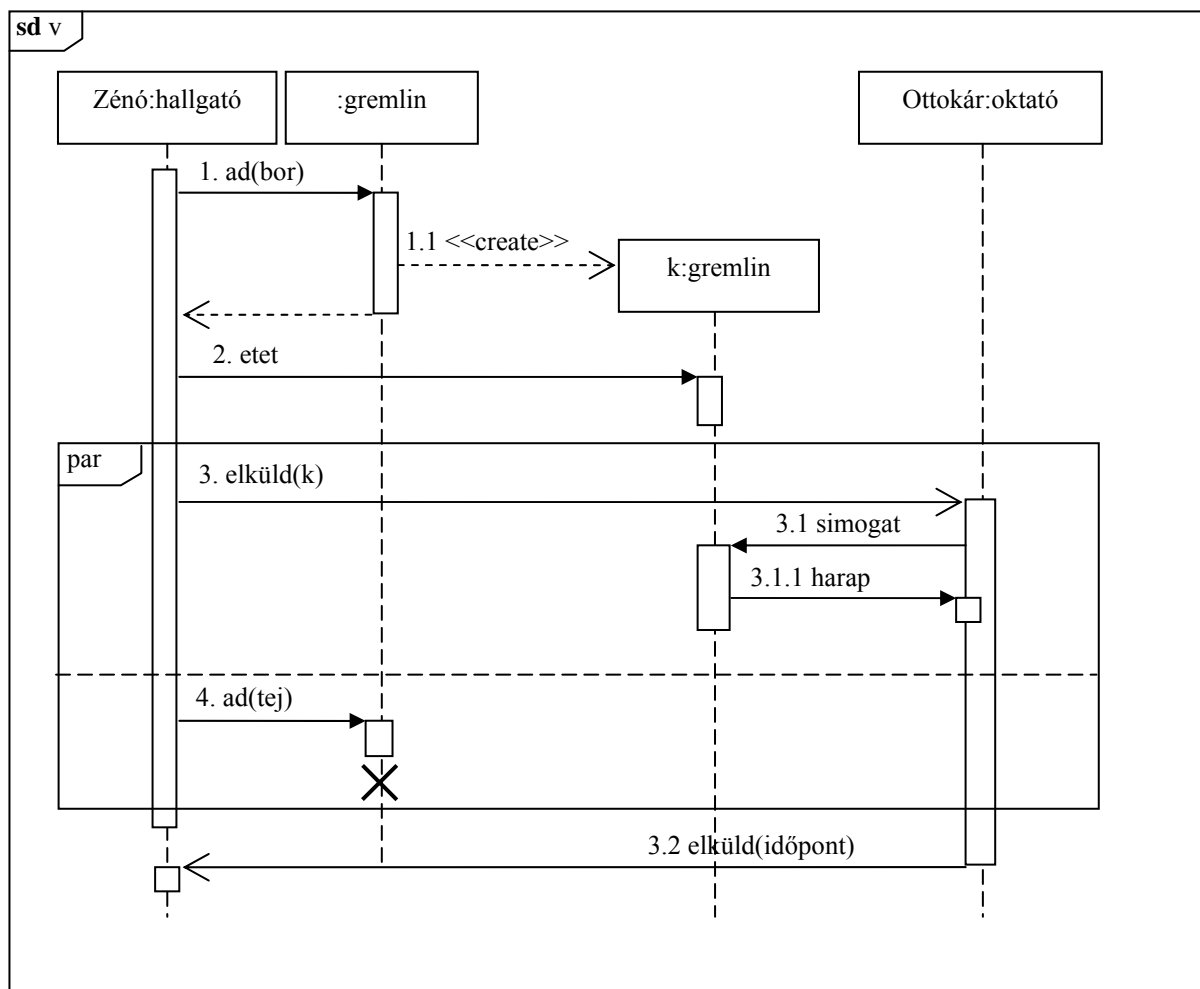
- **bottom-up**

Mi a funkciója a teszt ágyaknak (test bed) ? (2 pont)

- **a tesztelendő alrendszerek kezeléséhez keretet ad**

7. Készítsen UML2 szekvencia-diagramot az alábbi történet alapján! Ne feledkezzen el a hierarchikus számozásról sem! (8 pont)

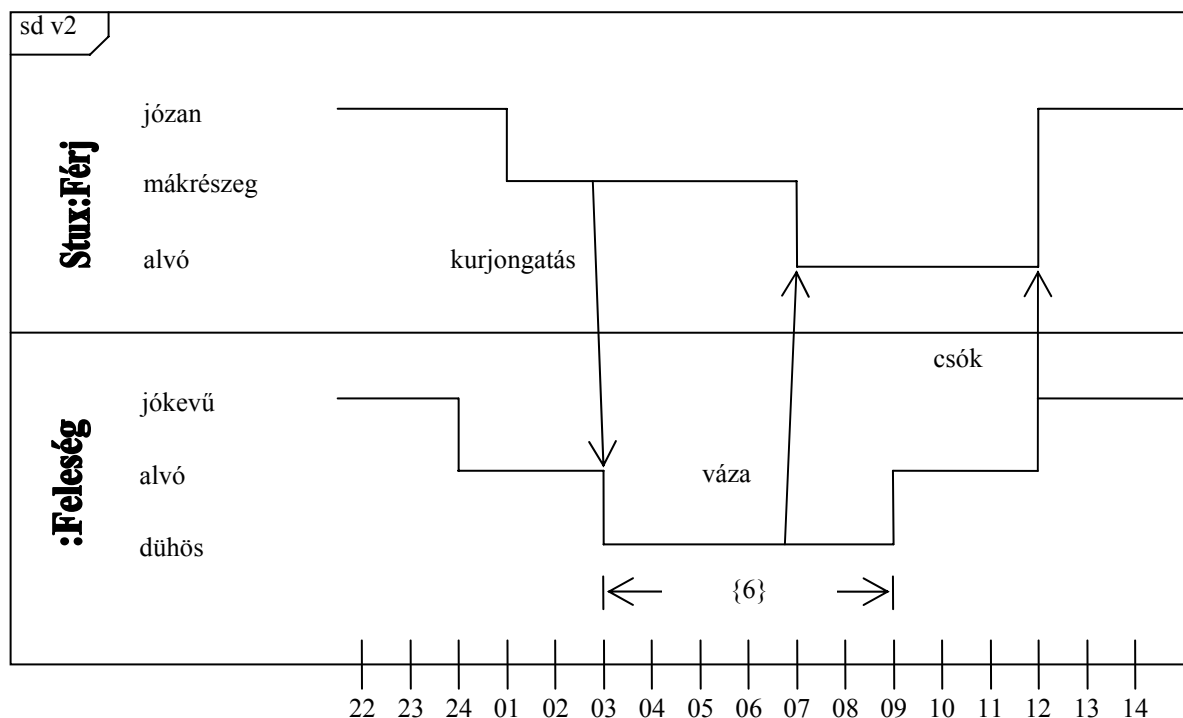
Zénó a karácsonyra kapott gremlinjének véletlenül bort ad, mire az egy kisgremlinnek ad életet. Zénó nagyon megőrül, és egyszer megeteti a kisgremlint, majd elküldi oktatójának, Gyíkarcú Ottokárnak házi feladat helyett. Ottokár megsimogatja a kisgremlint, aki simogatás közben megharapja. Közben Zénó az eredeti gremlinnek tejet ad, aki ebbe belepusztul. Végül Ottokár elküldi Zénónak a pótleadás időpontját.



8. Nevezze meg sorrendben a CMM (Capability Maturity Model) felső három szintjét ! (3 pont)

3. definiált (defined).....
4. irányított (managed).....
5. optimalizált (optimizing).....

9. Az alábbi történet alapján rajzoljon UML 2.0 időzítési diagramot (timing diagram)! (7 pont)
 Stux este 10-kor józanul ment el otthonról, egyedül hagyva jókedvű feleségét, aki 12-kor elaludt. Stux hajnali 1-re lett mákrészeg, és hajnali 3-ra ért haza. Ekkor a kurjongatásra felesége dühösen ébred, és hat órán át dühös is marad. 7-kor felesége egy vázát vág hozzá, amitől Stux elalszik. Miután az asszony kidühöngte magát, elalszik. Mikor délben jókedvűen felébred, megcsókolja férjét, aki józanul ébred.



10. Jellemezzünk egy (x,y) párokból álló listát – ahol x egy kulcs, y egy 0-nál nagyobb egész érték – az alábbi műveletekkel! Adja meg az algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket. (4 pont)

- NEW()** új (üres) listát hoz létre.
ADD(l,(x,y)) az l listához kapcsolja az x,y párt, ha x nem szerepelt a listán. Ha a listán már van x , akkor a hozzá tartozó értéket az új y -nal helyettesíti, ha az nagyobb a listán szereplőnél.
VALUE(l,x) megadja az l listán az x kulcshoz tartozó y -t. Ha a listán a megadott x nem szerepel, akkor az eredmény 0.
MAX(l) a listában szereplő legnagyobb y . (üres lista esetén nulla)

$VALUE(NEW()) = 0$ 0.5 pont
 $MAX(NEW()) = 0$ 0.5 pont
 $VALUE(ADD(l, (x,y),z)) = \text{if}(x == z \ \&\& \ VALUE(l,x) < y) \ y \ \text{else} \ VALUE(l, z)$ 2 pont
 $MAX(ADD(l, (x,y))) = \text{if}(MAX(l) < y) \ y \ \text{else} \ MAX(l)$ 1 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

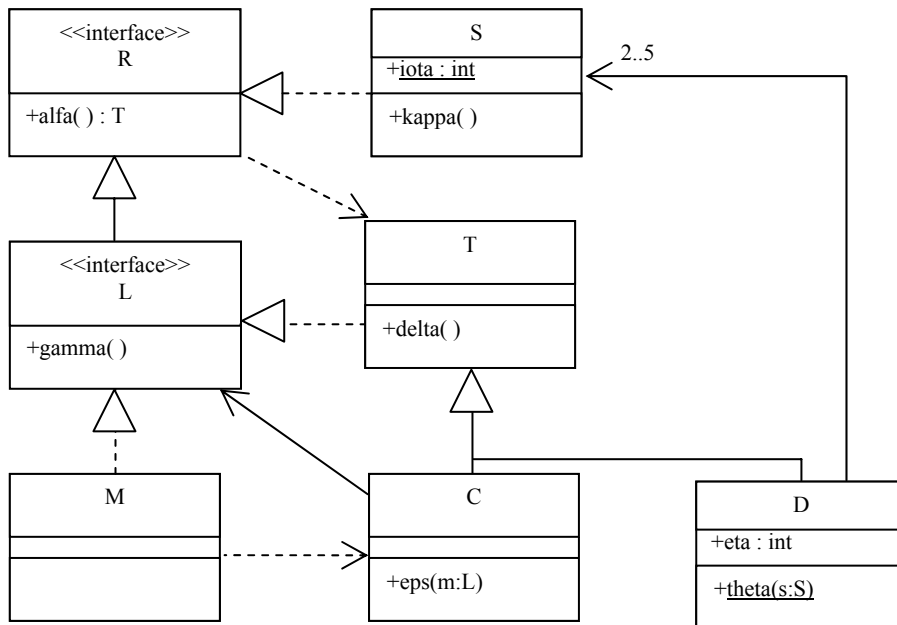
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2010. január 12. 12:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B]** M bárhol helyettesíthető C-vel, mert mindketten megvalósítják az R interfészt.
- [E]** C eps(m:L) metódusa nem hívhatja meg a paraméter gamma() metódusát, mert az utóbbi metódus protected láthatóságú.
- [C]** D theta(s:S) metódusa módosíthatja a paraméter iota attribútumát, mert a theta(s:S) statikus.
- [E]** T osztálynak nincs alfa():T szignatúrájú metódusa, mert T nem valósítja meg az R interfészt.
- [A]** M alfa():T metódusa visszaadhat C objektumot, mert C függ M-től.
- [B]** D theta(s:S) metódusa kaphat paraméterül C objektumot, mert S és C is megvalósítja az R interfészt.
- [E]** S nem valósítja meg az alfa():T szignatúrájú metódust, mert S nem függ T-től.
- [B]** D theta(s:S) metódusa legfeljebb 5-ször hívható meg, mert D objektum legfeljebb 5 S-sel állhat asszociációban.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. A szoftverfejlesztés „spirális modelljé”-nek a 2. szektorában mi a megoldandó feladat ? (3 pont)

- specifikálás
- célok kijelölése
- tervezés
- implementálás
- validálás
- kockázatok becslése

3. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

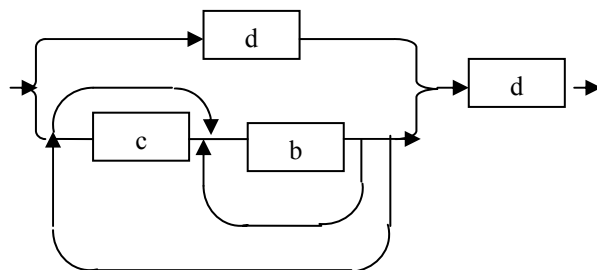
```
<!DOCTYPE x [
  <!ELEMENT x ((d|(c?,b+)+), d)>
  <!ATTLIST b d CDATA #IMPLIED>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
  <!ELEMENT d (#PCDATA)>
]>
<x>
  <d>b="Vizsga" </d>
  <d><![CDATA[ </d>
  <x>vizsga</x>
  <d>]]></d>
```

- IGEN
- NEM

Hiba ? :

.....

Rajzolja fel a DTD-vel definiált adatszerkezet szintaxis gráfját. A DTD esetleges hibáját hagyja figyelmen kívül ! (3 pont)



4. Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe ? (3 pont)

komponensek: eseményforrás, eseményfigyelő
szerepek: az eseményfigyelők beregisztrálják magukat a forrásoknál, akik a náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek

5. Az alábbi táblázat felsorolja az UML2-ben definiált konkurencia szemantikákat. Egy embernek több telefonja van. Miközben az egyikben beszél, egy másikon (konkurens) hívás érkezik. A táblázatba írja be, hogy a különböző szemantikák szerint a konkurens hívás előfordulhat-e, és mi történik az éppen zajló beszélgetéssel és az új hívással ! (5 pont)

szemantika neve	előfordulhat-e?	mi történik ? éppen zajló beszélgetés / konkurens hívás
konkurens (concurrent)	<input checked="" type="checkbox"/>	félbeszakítja / azonnal felveszi
sorrendi (sequential)	<input type="checkbox"/>	folytatja
őrzött (guarded)	<input checked="" type="checkbox"/>	folytatja / felveszi, ha befejezte

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

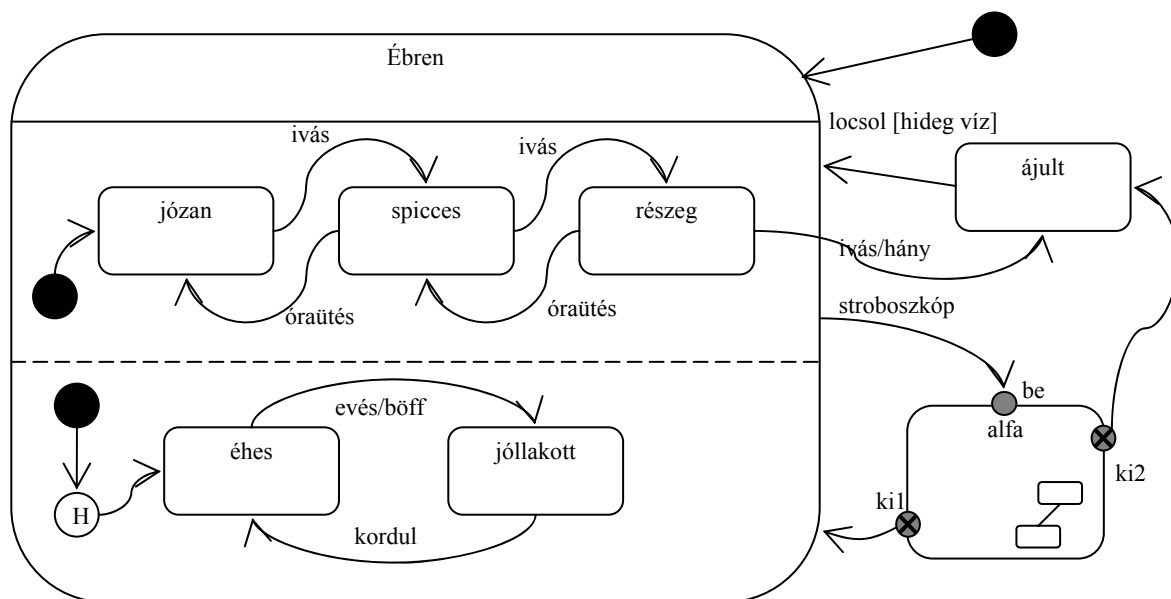
6. Az üzleti folyamatok lefutásának két esete az (1) orkesztráció (orchestration) és a (2) koreográfia (choreography). Mi a két megközelítés közötti különbség? (4 pont)

(1) orkesztráció (orchestration) – egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.

(2) koreográfia (choreography) – a folyamat nincs központosítva, minden résztvevő a dolgát elvégezve az általa ismert következő résztvevő(ke)t aktivizálja.

7. Rajzoljon UML 2.0 állapotábrát (state chart) az alábbi történet alapján! (7 pont)

Stux ébren háromféle hangulatban lehet: józan, spicces, részeg. Ezen kívül (szintén ébren) lehet éhes vagy jóllakott. Értelemszerűen, ha józan és iszik, akkor spicces, ha megint iszik, részeg lesz. Óraütésre visszafele változik. Ha éhes és eszik, akkor elbőffenti magát és jóllakott lesz. Akkor éhez meg, mikor megkordul a gyomra. Ha részeg, és még iszik, akkor elhányja magát és elájul, amely állapotban sem az éhséget, sem a jóllakottságot nem érzi. Ha ájult, akkor addig marad így, amíg le nem locsolják, de csak hideg vízre reagál. Ájultából kelve mindig józan lesz, de az éhsége vagy jóllakottsága nem változik. Mindezekon kívül le tud menni alfába. Ennek az állapotnak a belsejéről annyit tudunk, hogy összetett, de többet nem. Egy entry (be) és két exit pontja (ki1, ki2) van. Éber állapotból kerülhet ide, ha stroboszkópba néz. Hogy hogyan jön ki belőle, arról csak annyi bizonyos, hogy a ki1 pontból józanul és éhesen jön elő, a ki2-n keresztül pedig elájul. Az életét állítólag józanul és éhesen kezdte.

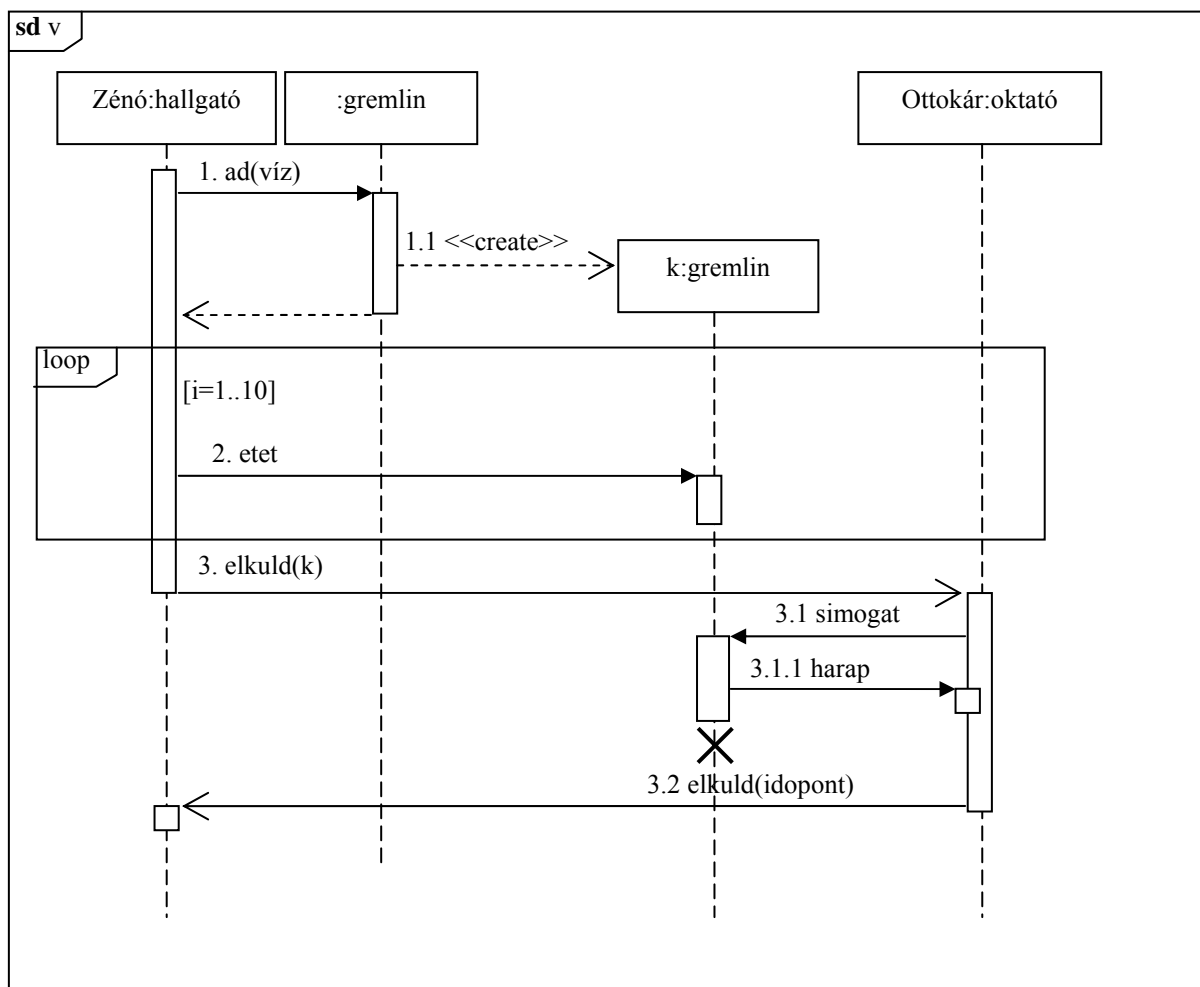


8. Az UML2-ben a gyűjteményeknek (kollekcióknak) két fontos tulajdonsága van: rendezettség (ordered) és egyediség (unique). Írja be a táblázatba a kollekciók nevét! (4 pont)

rendezett	egyedi	név
nem	nem	Bag
nem	igen	Set
igen	igen	Ordered Set
igen	nem	Sequence

9. Készítsen UML2 szekvencia-diagramot az alábbi történet alapján! Ne feledkezzen el a hierarchikus számozásról sem! (8 pont)

Zénó a karácsonyra kapott gremlinjének véletlenül vizet ad, mire az egy kisgremlinnek ad életet. Zénó nagyon megőrül, és 10-szer megeteti a kisgremlint, majd elküldi oktatójának, Gyíkarcú Ottokárnak házi feladat helyett. Ottokár megsimogatja a kisgremlint, aki simogatás közben megharapja, de Ottokár vitriolos véréből el is pusztul. Ottokár elküldi Zénónak a pótleadás időpontját.



10. Mire szolgál a verziókezelésben alkalmazott „modify-update-merge” stratégia ? Röviden írja le a működésének lényegét ! (3 pont)

mire: termékekhez történő konkurens hozzáférés szabályozása

lényeg: párh. check-out, check-in-kor összefésülés, konfliktusfelold...

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

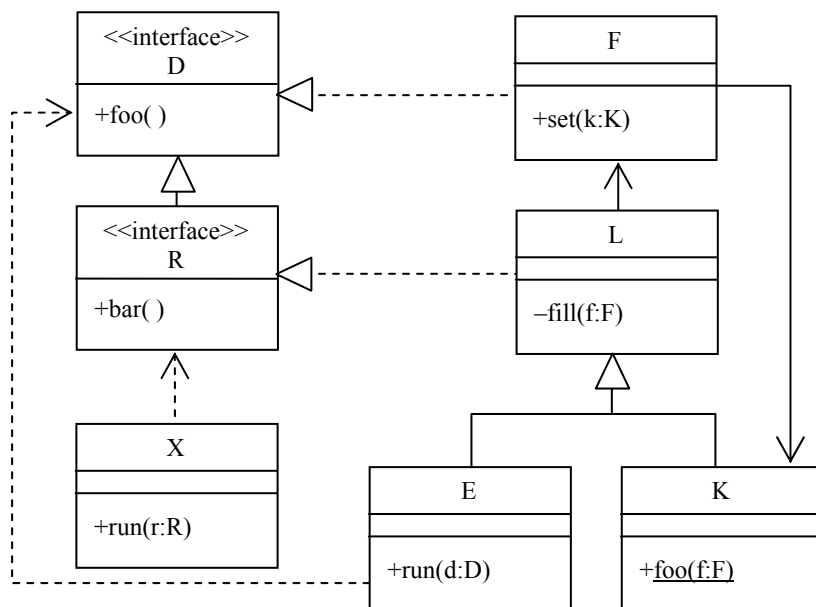
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2010. január 5. 13:30

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



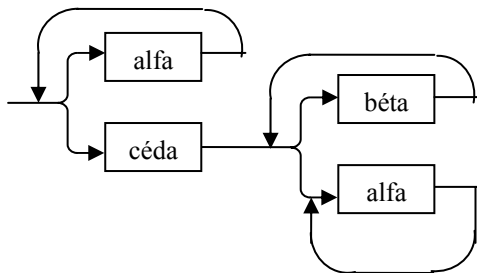
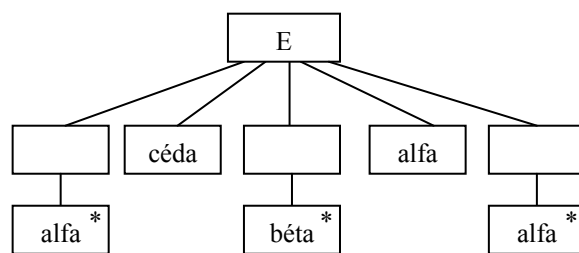
- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B] X run(r:R) metódusa kaphat paraméterül F osztályú objektumot, mert X függ R-től.
- [E] K-nak nincs foo() szignatúrájú metódusa, mert K-t nem lehet példányosítani.
- [E] L bárhol helyettesíthető F-fel, mert mindketten megvalósítják az R interfészt.
- [B] L nem helyettesíthető E-vel, mert L-nek van privát metódusa.
- [B] X run(r:R) metódusa nem kaphat paraméterül K objektumot, mert K-nak van statikus metódusa.
- [B] K foo(f:F) metódusa nem hívhatja meg a paraméter foo() metódusát, mert az utóbbi metódus nem statikus.
- [B] E bárhol helyettesíthető K-val, mert van közös ősük.
- [C] F set(k:K) metódusa nem hívhatja meg egy paraméterül kapott K fill(f:F) metódusát, mert K függ F-től.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Az E entitás élettörténetét az alábbi állapottáblával írhatjuk le (**betű** a kezdőállapot). Rajzolja fel az élettörténetet JSD ábrán és szintaxis gráffal ! (6 pont)

	alfa	béta	céda
betű	betű	--	szám
szám	egyéb	szám	--
egyéb	egyéb	--	--



3. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE a [
  <!ELEMENT a ((b|(c?, d+)), c)>
  <!ATTLIST d d CDATA #REQUIRED>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
  <!ELEMENT d (#PCDATA)>
]>
<a><b/><c/></a>
```

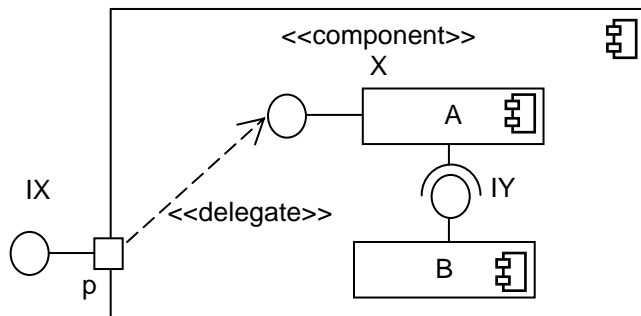
- IGEN
 NEM

Hiba ? :

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszer-kezetet, amelyben van d elem ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<a>
  <d d="xxx">dddd</d>
  <c/>
</a>
```

4. Legyen egy X komponensünk, amely p portján megvalósítja az IX interfészt. Ennek a komponensnek a felépítéséhez felhasználjuk az A komponenst, amely realizálja az IX interfészt, de szükséges felhasználnunk egy B komponenst is, amely megvalósítja az A által elvárt IY interfészt. Rajzoljon UML2 komponens diagramot ! (3 pont)

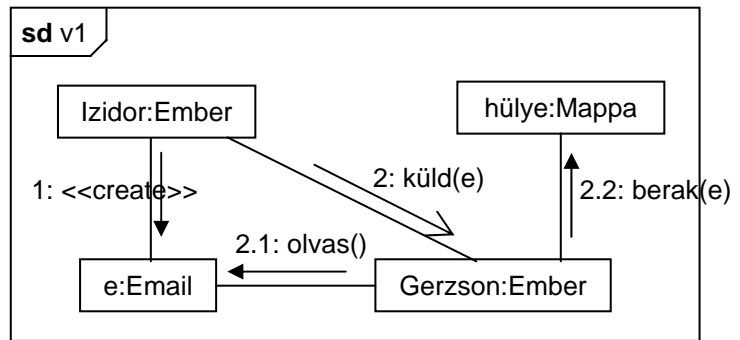


5. A specifikáció célja a követelményeknek eleget tevő rendszer formális leírása. Milyen három fontos nézőpontból készítjük a leírásokat ? (3 pont)

- funkcionalitás** **szerkezet (struktúra)**
dinamika (viselkedés)

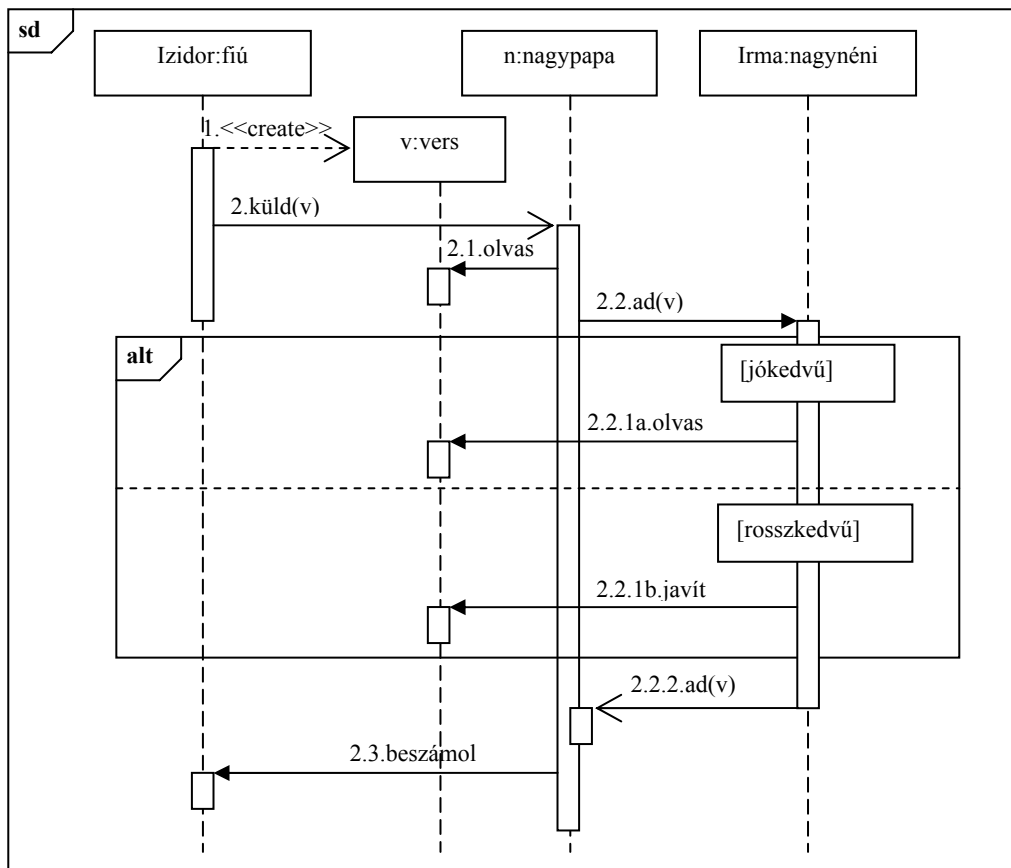
A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Izidor ír egy e-mailt, amit elküld Gerzsonnak. Gerzson a levelet elolvassa, majd beteszi a „hülye” mappába. Rajzoljon UML2 kommunikációs diagramot ! Alkalmazzon hierarchikus számozást ! (3 pont)



7. Készítsen UML2 szekvencia-diagramot az alábbi történet alapján! Ne feledkezzen el a hierarchikus számozásról sem! (8 pont)

Izidor verset ír ajándékként vidéken élő nagynénjének, Irmának. Az ajándékot a nagypapának küldi azzal, hogy adja át a nagynéninek. A nagypapa kíváncsi, és elolvassa a verset. Ezután odaadja Irmának, és várja a hatást. Ha a nagynéni jókedvű, akkor a verset felolvassa, ha rosszkedvű, akkor aláhúzza benne a nyelvtani hibákat. Mindezek után a verset visszaadja a nagypapának, és elsiet. A nagypapa ezután felutazik Pestre, és a történekről beszámol Izidornak.

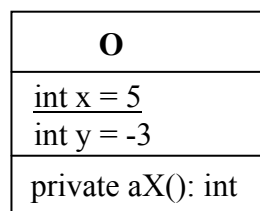


8. Elkészítjük az alábbi O osztály két példányát, $o1$ -et és $o2$ -t. Ezt követően végrehajtjuk a következő műveleteket:

- $o2.x = -3; o1.x = 4;$
- $o1.y = o2.x + 4;$
- $o2.y = o2.x + o1.y;$

Mennyi lesz a változók értéke ? (4 pont)

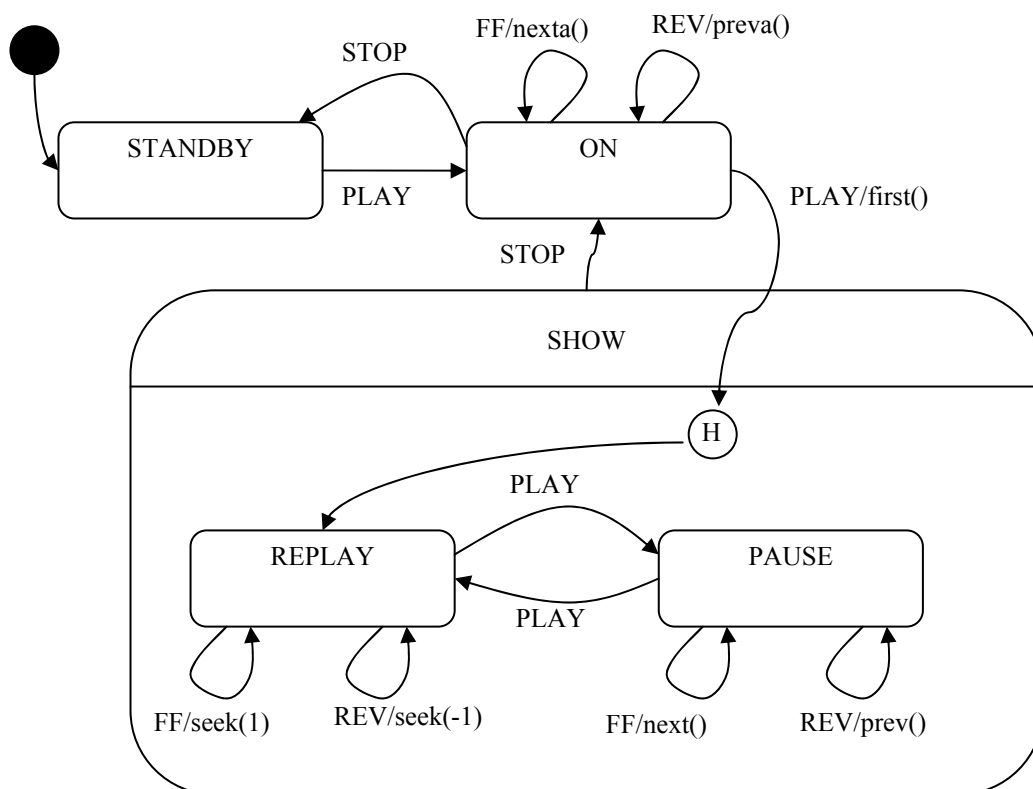
$o1.y = 8$ $o2.y = 12$



9. Rajzoljon UML2 state-chartot (állapot-diagram) az alábbi történet alapján ! (7 pont)

A Dárebéz MP3 lejátszón 4 gomb van: PLAY, STOP, FF, REV. Amikor elemet teszünk bele, akkor STANDBY állapotba kerül. PLAY hatására kapcsol be (ON). Ekkor az FF és a REV gombokkal lehet előre- és hátralépni az albumok között. PLAY megnyomására lejátszó (SHOW) módba kerül, amikor vagy az aktuális album első számát kezdi lejátszani (REPLAY), vagy szünetelteti a lejátszást (PAUSE). Hogy melyiket csinálja, az attól függ, hogy utoljára melyiket végezte SHOW módban (ha még egyiket sem, akkor REPLAY az alap). Ha REPLAY alatt nyomkodjuk az FF és a REV gombokat, a számon belül tekerünk előre vagy hátra 1 mp-nyit. Ha PAUSE módban nyomkodjuk őket, akkor a számok között ugrálhatunk. STOP hatására ismét ON-ba kerülünk, újabb STOP-ra STANDBY-ba. SHOW állapotban a PLAY gombbal lehet megállítani (PAUSE) és újraindítani (REPLAY) az aktuálisan játszott számot.

A lejátszó mp3-API-ja a következő függvényeket ismeri: *seek(int x)*: x mp-et előre megy; *next()*, *prev()*: következő, előző számot választja; *nexta()*, *preva()*: következő/előző album; *first()*: album első számára áll.



10. Hogyan értelmezzük az UML2-ben a szekvenciális konkurenciát ? (2 pont) A magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

callers must coordinate outside the object so that only one flow is in the object at a time.

Sorolja fel az UML2 által definiált egyéb konkurencia szemantikákat ! (2 pont)

guarded, concurrent

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

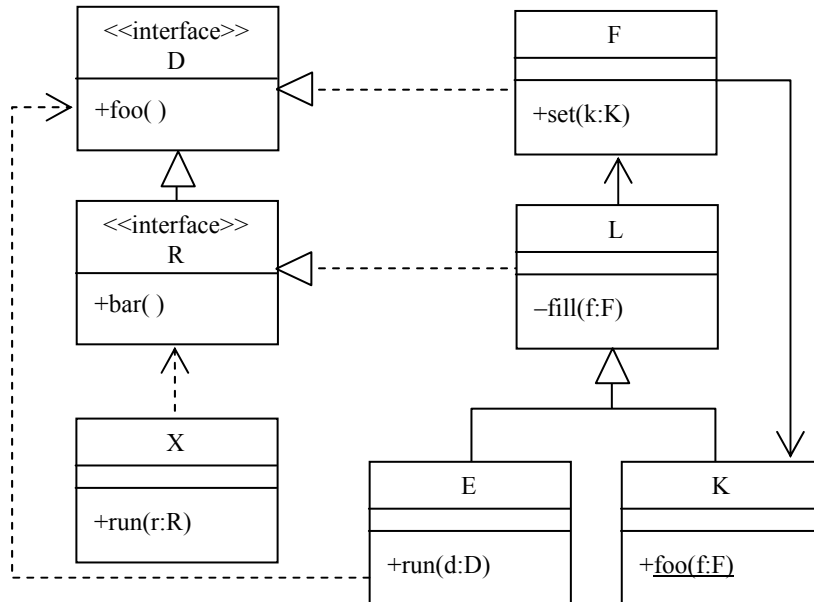
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2010. január 5. 12:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



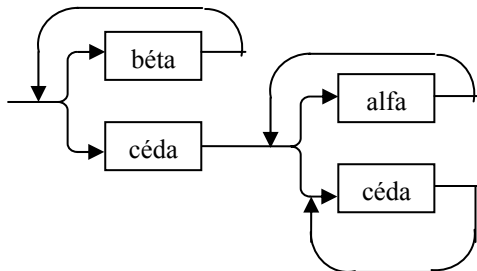
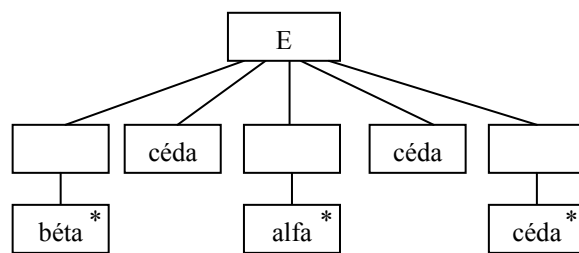
- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B] E bárhol helyettesíthető K-val, mert van közös ősük.
- [B] L bárhol helyettesíthető F-fel, mert mindketten megvalósítják a D interfészt.
- [B] L nem helyettesíthető E-vel, mert L-nek van privát metódusa.
- [B] F set(k:K) metódusa meghívhatja egy paraméterül kapott K fill(f:F) metódusát, mert K függ F-től.
- [B] X run(r:R) metódusa kaphat paraméterül F osztályú objektumot, mert X függ R-től.
- [E] K-nak nincs foo() szignatúrájú metódusa, mert K-t nem lehet példányosítani.
- [B] X run(r:R) metódusa nem kaphat paraméterül K objektumot, mert K-nak van statikus metódusa.
- [B] K foo(f:F) metódusa nem hívhatja meg a paraméter foo() metódusát, mert az utóbbi metódus nem statikus.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Az E entitás élettörténetét az alábbi állapottáblával írhatjuk le (betű a kezdőállapot). Rajzolja fel az élettörténetet JSD ábrán és szintaxis gráffal ! (6 pont)

	alfa	béta	céda
betű	--	betű	szám
szám	szám	--	egyéb
egyéb	--	--	egyéb



3. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE x [
  <!ELEMENT x ((b|(a+,c+)), a)>
  <!ATTLIST b d CDATA #REQUIRED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <b>d="f"</b>
  <a/>
</x>
```

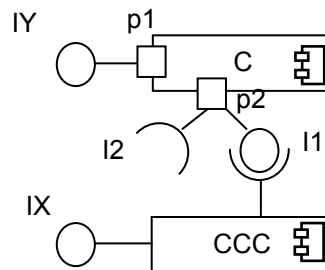
- IGEN
 NEM

Hiba ?: **-nek nincs paramétere**

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van c elem ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <a>aaa</a>
  <c>aaa</c>
  <a>aaa</a>
</x>
```

4. Legyen egy C komponensünk, amely a p1 portján megvalósítja az IY interfészt, a p2 portján megvalósítja az I1 interfészt és várja az I2 interfészt. Van egy CCC komponensünk is, amely az I1 interfészen keresztül kapcsolódik a C komponenshez. A CCC komponens megvalósítja az IX interfészt is. Rajzoljon UML2 komponens diagramot ! (3 pont)

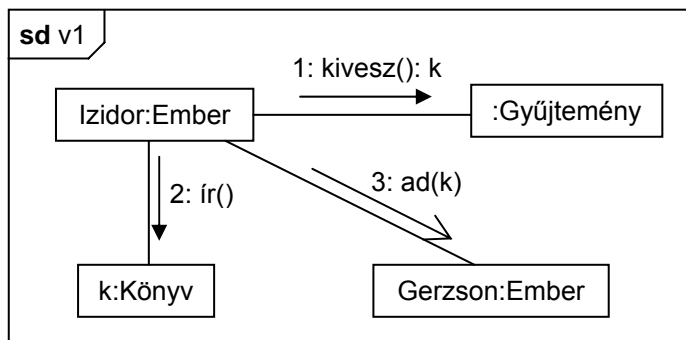


5. Adja meg, hogy egy szoftver elem (dokumentáció, kód) felülvizsgálata (review, inspection) során minden egyes azonosított probléma vagy hiba esetén a felülvizsgálók mit rendelkeznek (mi van a jegyzőkönyvben ?) ! (3 pont)

- kijelölik a felelőst
- előírják a szükséges tevékenységet (pl. további vizsgálat, átdolgozás)
- meghatározzák a hiba típusát és a probléma súlyát

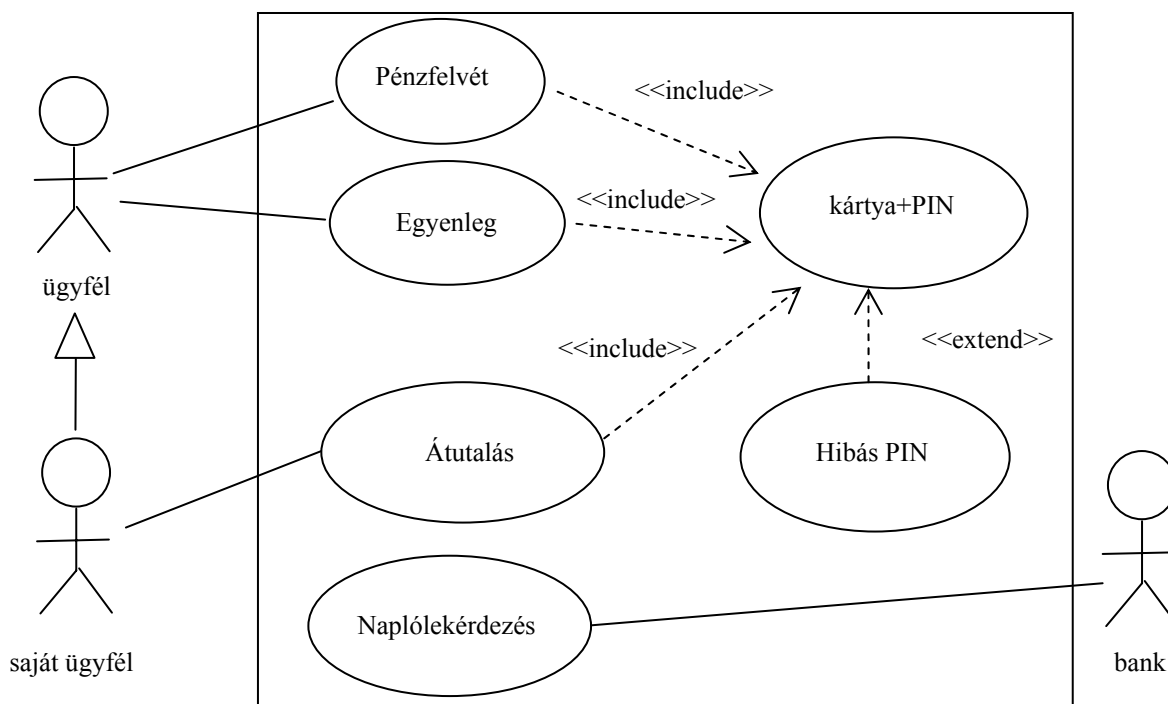
A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Izidor kiveszi a könyvgyűjteményéből kedvenc könyvét, ajánlást ír bele, majd odaadja Gerzsonnak. Rajzoljon UML2 kommunikációs diagramot ! (3 pont)



7. Rajzoljon UML2 use-case diagramot az alábbi történet alapján! (7 pont)

A Management Optimal Bonus (MOB) Bank automatáival pénzt lehet felvenni, számlaegyenleget lehet lekérdezni, és a bank saját ügyfelei pénzt utalhatnak a bank vezetőségi bónusz programja számára. Mindezen funkciók eléréséhez be kell helyezni a kártyát és meg kell adni a 4 jegyű azonosítót (PIN). Ha ez háromszor egymás után nem sikerül, az automata a kártyával elérhető teljes összeget a bank jutalomkeretére utalja. Ezen kívül a bank lekérdezheti az automata naplófájlját.



8. Elkészítjük az alábbi C osztály két példányát, c1-et és c2-t. Ezt követően végrehajtjuk a következő műveleteket:

```

c2.a = 8; c1.a = -2;
c1.b = c2.a + 4;
c2.b = c2.a + c1.b;

```

Mennyi lesz a változók értéke ? (4 pont)

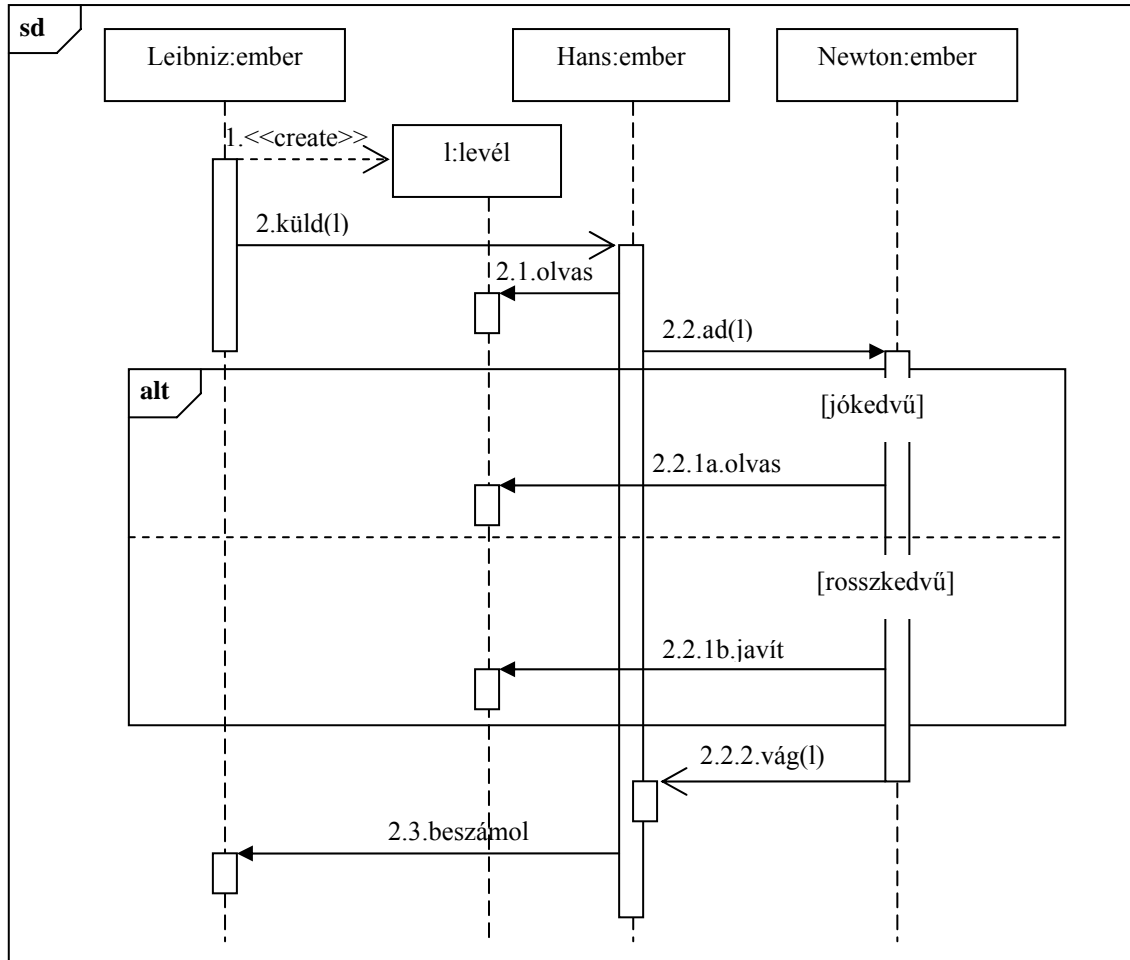
c1.b = 2

c2.b = 0

C
int a = 12
int b = 5
private aX(): int

9. Készítsen UML2 szekvencia-diagramot az alábbi történet alapján! Ne feledkezzen el a hierarchikus számozásról sem! (8 pont)

Gottfried Wilhelm Leibniz szeretne tálkozni Sir Isaac Newtonnal. Ír egy latin nyelvű levelet, amelyben differenciálszámítással kapcsolatos eredményeit ecseteli. A levelet hű barátjának, az éppen Angliában tartózkodó Hans Georg von Hirscheissenfeldnek küldi azzal, hogy adja át Newtonnak. Hans kíváncsi, és elolvassa a levelet. Ezután találkozik Newtonnal, és odaadja neki a levelet. Ha Newton jókedvű, akkor a levelet elolvassa, ha rosszkedvű, akkor a levélben aláhúzza a nyelvtani hibákat. Mindezek után a levelet hozzávágja Hanshoz, és elsiet. Hans hazautazik, és az eseményről beszámol Leibniznek.



10. Felsoroltunk szoftverrel kapcsolatos "hibákat". A hibák mellett jelölje be, hogy az melyik kategóriába tartozik! (4 pont) Rossz válasz esetén pontot veszít !

	bug	failure	error	fault
Hiányzik a "synchronized" kulcsszó	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A ciklusfeltétel hibás	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Elmaradt a kritikus kódok felülvizsgálata	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Alábecsültük a példányosítás erőforrás-igényét	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sok objektumnál nagyon lassan kapjuk az eredményt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Az első futásnál hibás eredményt kapunk	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

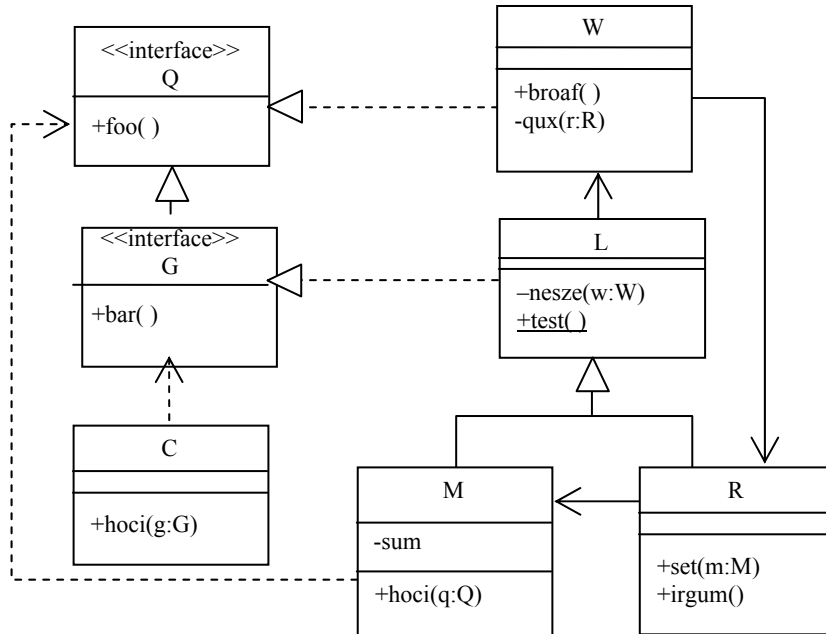
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2009. június 18. 13:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



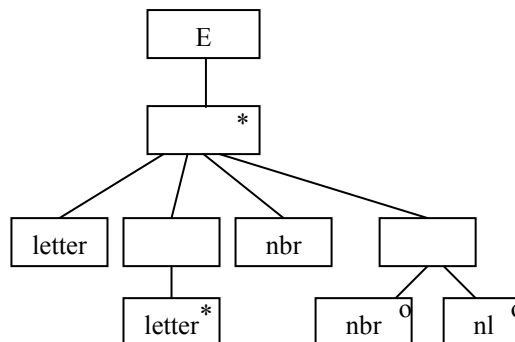
- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [E] **M hoci(q:Q)** függvénye meghívhatja egy paraméterül kapott **W broaf()** metódusát, mert a **broaf** metódus statikus.
- [B] **R set(m:M)** metódusa kaphat paraméterül **L** objektumot, mert **M** az **L** leszármazottja.
- [B] **L nesze(w:W)** metódusa meghívhatja a paraméterül kapott objektum **qux(r:R)** metódusát, mert mindkét metódus privát.
- [B] **W** bárhol helyettesíthető **L**-l, mert mindketten megvalósítják a **Q** interfészt.
- [E] **R**-nek nincs **foo()** szignatúrájú metódusa, mert nem valósítja meg a **G** interfészt.
- [A] **C hoci(g:G)** metódusa kaphat paraméterül **M** objektumot, mert **M hoci(q:Q)** metódusa is kaphat paraméterül **C**-t.
- [B] **L nesze(w:W)** metódusa nem hívhatja meg a **test()** metódust, mert a **test()** statikus.
- [A] **W qux(r:R)** metódusából bármikor meghívható a paraméter **irgum()** metódusa, mert a két osztály nem függ egymástól.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Az E entitás élettörténetét az alábbi állapottáblával írhatjuk le (**ax** a kezdőállapot). Rajzolja fel az élettörténetet JSD ábrán ! (4 pont)

	letter	nbr	nl
ax	by	--	--
by	by	cz	--
cz	--	ax	ax



3. A szoftver fejlesztés Ward-Mellor modelljében mi áll a (2 pont)

vízszintes tengelyen: **implementációs függőség**

függőleges tengelyen: **absztrakció**

Egy mondatban fogalmazza meg a vízszintes tengelyen szereplő fogalom jelentését ! (2 pont)

A leíráshoz használt fogalom mennyire “szoftveres”

4 Legyen egy C osztálynak egy $int\ y(int\ x)$ metódusa, ahol csak $500 < x < 5000$ megengedett, és akkor a $100 < y < 200$ igaz. C leszármazottjában felüldefiniáljuk y -t. Az alábbi állítások közül jelölje meg azokat, amelyek ha igazak a felüldefiniált y -on, akkor a leszármazott kielégíti a Liskov-elvet ! (4 pont)

Rossz válasz esetén pontot veszít !

- | | |
|--|--|
| <input type="checkbox"/> .. $100 < x < 200$ | <input checked="" type="checkbox"/> .. $150 < y < 200$ |
| <input checked="" type="checkbox"/> .. $100 < x < 10000$ | <input type="checkbox"/> .. $501 < x < 9990$ |
| <input type="checkbox"/> .. $555 < x < 5555$ | <input type="checkbox"/> .. $155 < y < 555$ |
| <input type="checkbox"/> .. $0 < y < 500$ | <input checked="" type="checkbox"/> .. $155 < y < 195$ |
| <input type="checkbox"/> .. $100 < y < 3000$ | <input checked="" type="checkbox"/> .. $0 < x < 5050$ |

Mi az összefüggés a metódus és az operáció között ? (1 pont)

A metódus implementálja az operációt

5. Adja meg, hogy egy szoftver elem (dokumentáció, kód) felülvizsgálata (review, inspection) során minden egyes azonosított probléma vagy hiba esetén a felülvizsgálók mit rendelkeznek (mi van a jegyzőkönyvben ?) ! (3 pont)

kijelölik a felelőst

előírják a szükséges tevékenységet (pl. további vizsgálat, átdolgozás)

meghatározzák a hiba típusát és a probléma súlyát

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

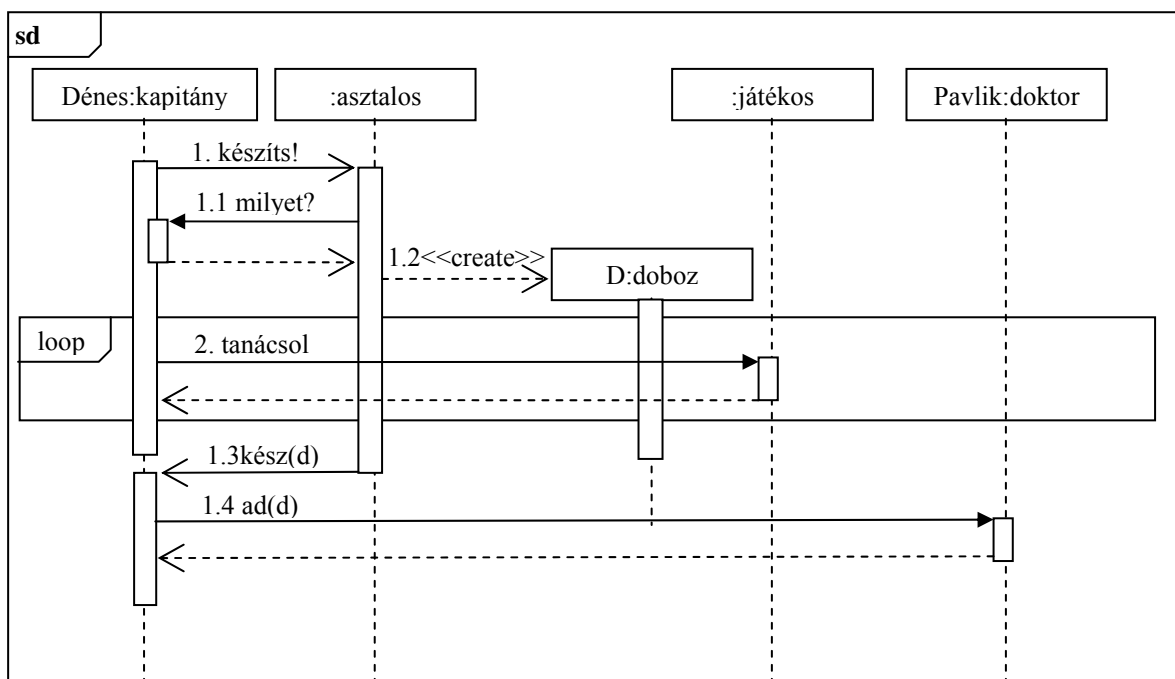
6. Jellemezzünk egy stringet az alábbi műveletekkel ! Adja meg a CUT műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket. (4 pont)

- CRT()** új (üres) stringet hoz létre.
- SET(s,x)** az s string elejére rakja az x karaktert.
- LGTH(s)** az s string karaktereinek számát adja.
- CUT(s, n)** az s string legrégebbi n darab karakterének levágása után maradó stringet adja
Ha n nem kisebb, mint s string hossza, üres stringet kapunk. Tételezze fel, hogy $n > 0$!

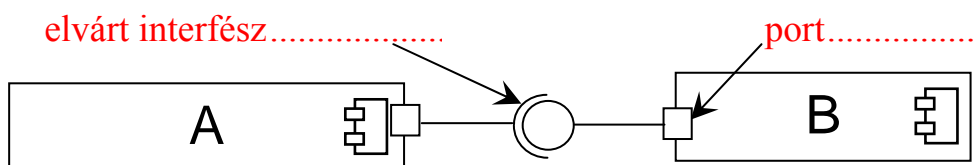
- CUT(CRT()) = CRT() -0.5 pont, ha hiányzik
- CUT(SET(s, x), n) == if (n > LGTH(s)) CRT() 1 pont
- if (n == LGTH(s)) SET(CRT(), x) 1 pont
- if (n < LGTH(s)) SET(CUT(s, n), x) 2 pont

7. Rajzoljon UML 2.0 szekvenciadiagramot (sequence diagram) az alábbi leírás alapján! (8 pont)

Kemény Dénes (a vízilabda-válogatott szövetségi kapitánya), mivel tudja, hogy Pavlik doktornak nincs miben tartania az olimpiai aranyait, ezért egy intarziás fadobozt csináltat kedvenc asztalosával. Az asztalos, mielőtt elkezdené a munkát, megkérdi Dénest, hogy pontosan milyen minta legyen a dobozon, majd elkezd legyártani a remekművet. Közben a kapitány minden egyes játékosát egyenként szakmai tanácsokkal látja el. Mikor a doboz elkészül, az asztalos elküldi Dénesnek, aki fogja, és azon nyomban átadja a doktornak.

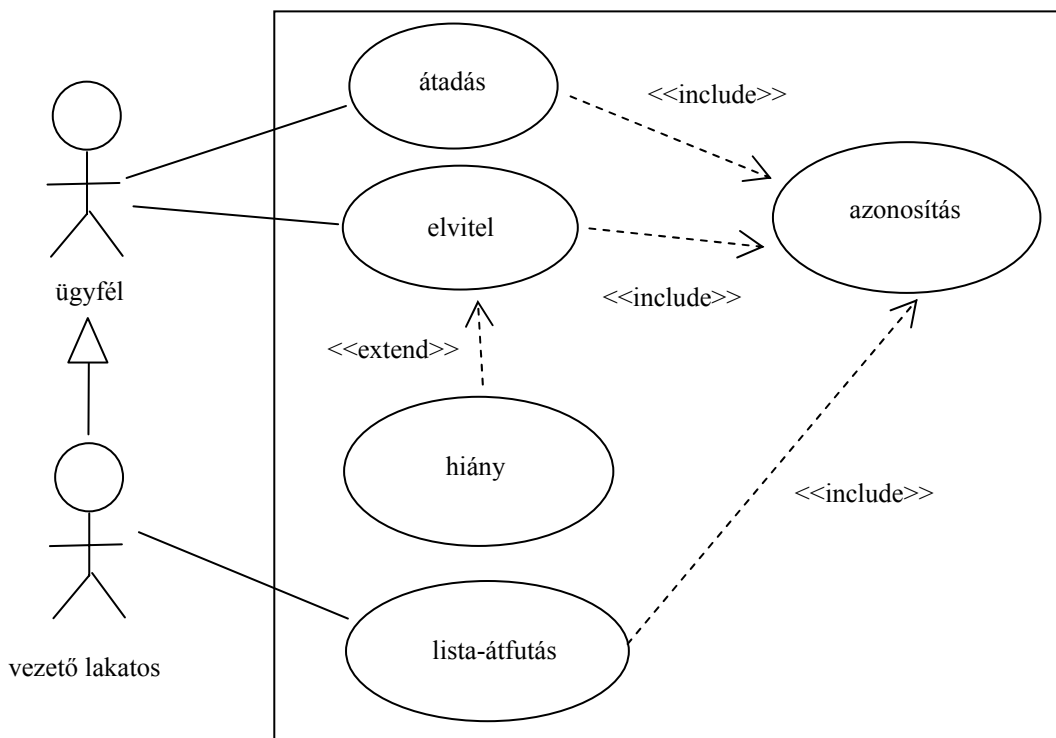


8. Az alábbi ábrán két UML2 modell elemet megjelöltünk. Adja meg elemenként, hogy az melyik UML2 meta-modell elem példánya ! (2 pont)



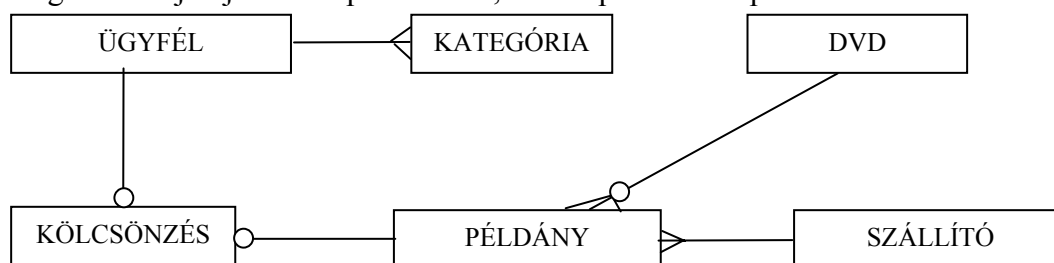
9. Rajzoljon UML 2.0 use-case diagramot az alábbi leírás alapján!

A Rezesbanda Kft. éjjel-nappali autóbontó telepet működtet, amit informatikai rendszerrel kíván megtámogatni. A regisztrált ügyfelek roncs autókat adnak át, és alkalmanként használt autóalkatrészt visznek el. Mindkét esetben jelszóval azonosítják magukat. Ha a kért autóalkatrész nincs raktáron, akkor a rendszer felírja a kérést a kívánságlistára. A telep vezető lakatosja időnként átfutja a kívánságlistát, hogy lássa, mire van szükség, ekkor ő is jelszóval azonosítja magát. A vezető lakatos ügyfélként is viselkedhet. (7 pont)



10. Egy videotékában DVD-ket lehet kölcsönözni. A DVD-kről (DVD) nyilvántartják a címét, a kiadóját, a kiadás évét, a műsor hosszát. Egy DVD-ből több példány (PÉLDÁNY) is lehet, de olyan DVD-k adatait is tárolják, amelyekből még vagy már nincs példány. Példányonként tárolják a vásárlás idejét, az árat, a szállítót és az adott példány kölcsönzéseinek számát. A szállítónak (már szállított) (SZÁLLÍTÓ) ismerik a nevét, címét és bankszámla számát. Az ügyfelet (ÜGYFÉL) azonosító jellemzi, de nyilvántartják a nevet, címet, telefonszámot. Egy kölcsönzés (KÖLCSÖNZÉS) alkalmával egy ügyfél egy DVD példányt megadott határidőig kölcsönbe vesz. A kölcsönzés megszűnik, ha az ügyfél a DVD-t visszahozta. Az ügyfelekről nyilvántartjuk, hogy élete során a különböző kategóriájú (KATEGÓRIA) (pl.: zene, akció, romantikus, vígjáték, krimi, thriller, sex, stb.) DVD-kből hányszor kölcsönzött.

Az alábbi ER diagramba rajzolja be a kapcsolatokat, a multiplicitás és opcionális feltüntetésével !(5 pont)



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

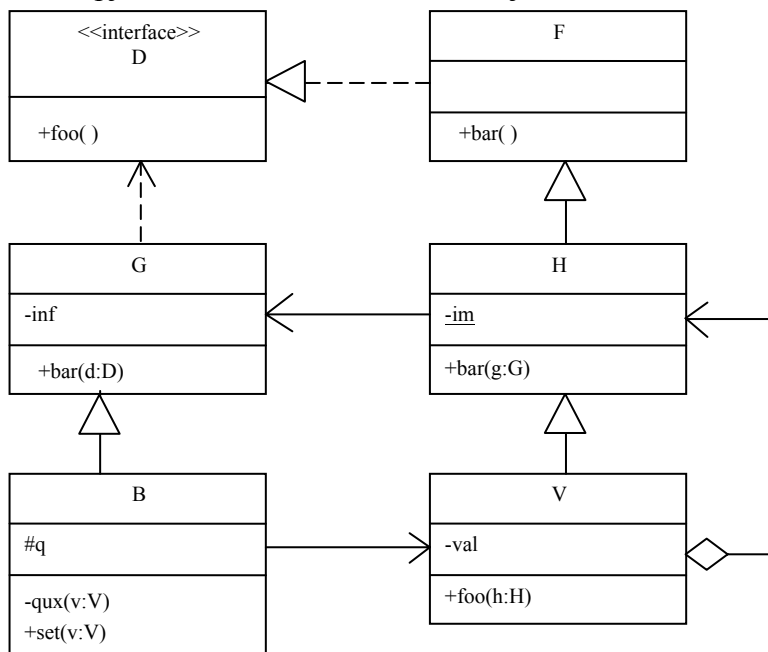
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2009. június 11. 13:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [E] **G** **bar(d:D)** metódusa kaphat paraméterül **B** objektumot, mert **G** a **B** leszármazottja
- [B] **H** **bar(g:G)** metódusa kaphat paraméterül **V** objektumot, mert **V** megvalósítja a **D** interfészt.
- [B] **B** **qux(v:V)** metódusa módosíthatja a paraméter **val** attribútumát, mert mind a metódus, mind az attribútum privát.
- [E] **H** **bar(g:G)** metódusa nem módosíthatja az **im** attribútumot, mert az attribútum konstans.
- [E] **B** objektum nem hívhatja meg egy **V** objektum **foo()** metódusát, mert **V**-nek nincs ilyen szignatúrájú metódusa.
- [E] **G** **bar(d:D)** metódusa meghívhatja egy paraméterül kapott **F** objektum **bar()** metódusát, mert a két metódus azonos szignatúrájú
- [B] **B** **set(v:V)** metódusa nem módosíthatja a **q** attribútumot, mert a láthatóságuk különböző.
- [E] **B**-nek van **foo()** szignatúrájú metódusa, mert **B** megvalósítja a **D** interfészt

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a+,c)|b), a)>
  <!ATTLIST a d CDATA #IMPLIED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <a/><c></c>
  <a d="f">b a d=f</a>
</x>
```

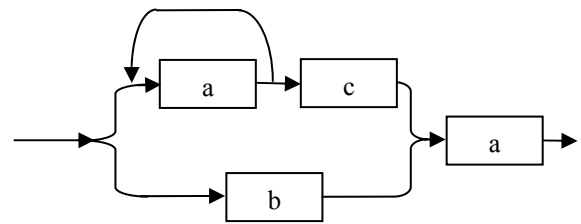
- IGEN
- NEM

Hiba ?:

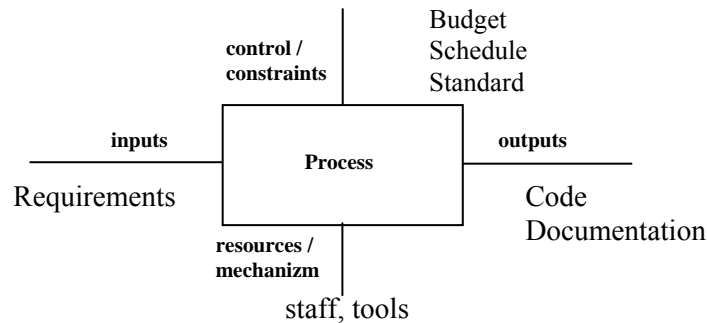
Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van b elem ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <b></b>
  <a></a>
</x>
```

Rajzolja fel az adatszerkezet szintaxisgráfját ! (2 pont)

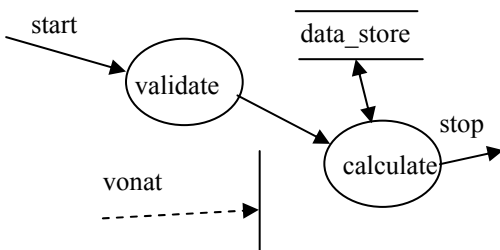


3. Adja meg a szoftver fejlesztési folyamat ICOM modelljét ! (4 pont)



4. Definiálja a vezérlőegység egy lehetséges állapotábráját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) a DFD-vel és az adatspecifikációval (4 pont)

vonat = [gyors | zónázó]



XXX			
YYY			

5. Egy objektum metódusa szekvenciálisan kohézív, ha a metódus ... (2 pont)

- kódjában csak utasítás-szekvenciák vannak (pl. nincs benne iteráció)
- adatszekvenciával (tömb, lista) tér vissza
- paramétere egy vagy több adatszekvencia (tömb, lista)
- több, egymást követő részfunkciót lát el
- nevében benne van a szekvencia (sequence) (pl. read_input_sequence)
- a metódus több funkciót hajt végre, ezek mind a kivételkezeléshez kapcsolódnak

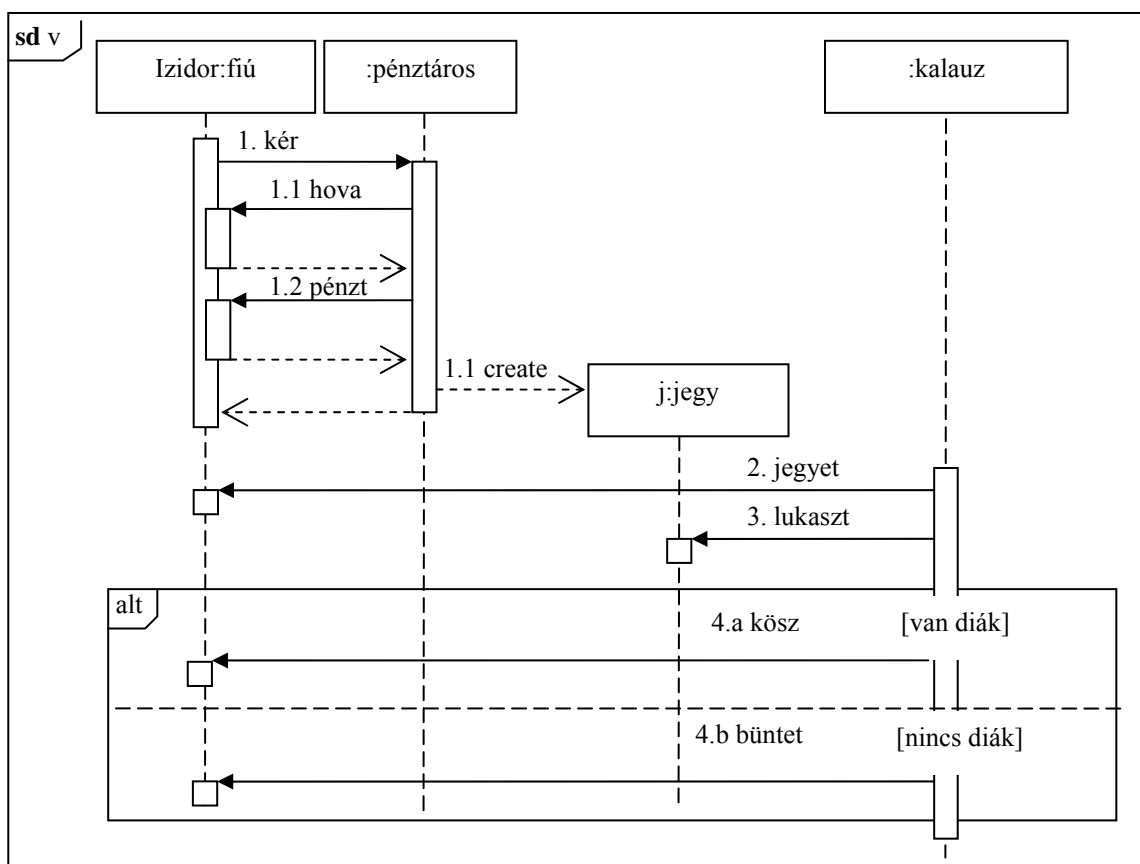
A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Nevezze meg sorrendben a CMM (Capability Maturity Model) szintjeit ! (5 pont)

1. kezdetleges (initial)
2. ismétlődő (repeatable)
3. definiált (defined).....
4. irányított (managed)
5. optimalizált (optimizing).....

7. Rajzoljon UML 2.0 szekvenciadiagramot (sequence diagram) az alábbi leírás alapján! (8 pont)

Izidor vonatjegyet szeretne venni, hogy elutazzon nagymamájához. A jegypénztárnál kér egy retúrjegyet. A pénztáros megkérdezi, hogy hova. A pénztáros elkéri a pénzt, majd kinyomtatja a jegyet, és a visszajáróval együtt Izidornak adja. Később (már a vonaton) a kalauz elkéri a jegyet és kilukasztja. Ha Izidornál nincs nála a diákigazolványa, akkor (a kalauz) megbünteti, ha nála van, akkor (a kalauz) megköszöni.

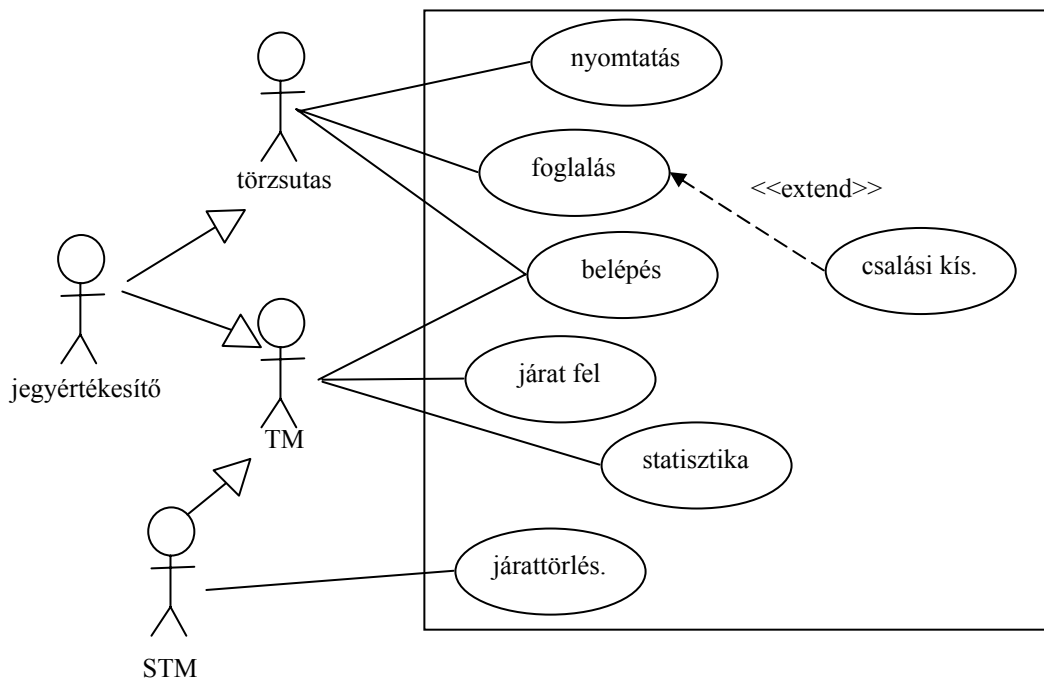


8. Hogyan értelmezzük az UML-ben a szekvenciális konkurenciát ? (2 pont)

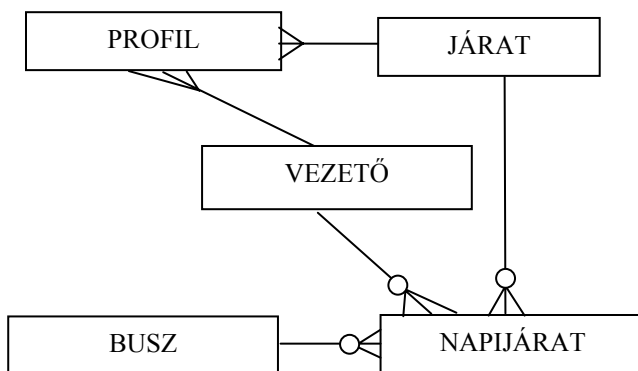
callers must coordinate outside the object so that only one flow is in the object at a time.

9. Rajzoljon UML 2.0 use-case diagramot az alábbi leírás alapján!

A Lushfanta légitársaság jegyfoglaló rendszerét törzsutasok és ticket-managerek (TM) használhatják a repülőjegyfoglalások rögzítésére. A törzsutasok bejelentkezhetnek, jegyet foglalhatnak és beszállókártyát nyomtathatnak. Ha jegyfoglalás közben kiderül, hogy módosultak a session-adatok, akkor a rendszer rögzíti a csalási kísérletet. A TM-ek is belépnek, járatok adatait vihetik fel, illetve jegyvásárlási statisztikákat kérhetnek le. A senior ticket-managerek (STM) az egyszerű TM lehetőségein túl még járatot is törölhetnek. A jegyértékesítők, mivel járatok adatait is kezelniük kell, mind TM-ként, mint törzsutasként használhatják a rendszert. (6 pont)



10. A Grimbusz társaság aktuális és tervezett buszjárait megtaláljuk a menetrendben. A járat jellemzője a járatszám, az induló és célállomás (JÁRAT). Egy adott napon a járatra (NAPIJÁRAT) vonatkozóan nyilvántartjuk a késést. Egy napon nem indul többször ugyanazon számú járat. A konkrét napi járathoz egy buszt (BUSZ) és egy vezetőt (VEZETŐ) rendelnek. Minden vezetőről tudjuk, hogy melyik járaton hány alkalommal vezetett (PROFIL). A napi járatokat 10 nap elteltével töröljük. Az alábbi ER diagramba rajzolja be a kapcsolatokat, a multiplicitás és opcionálitás feltüntetésével! (5 pont)



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

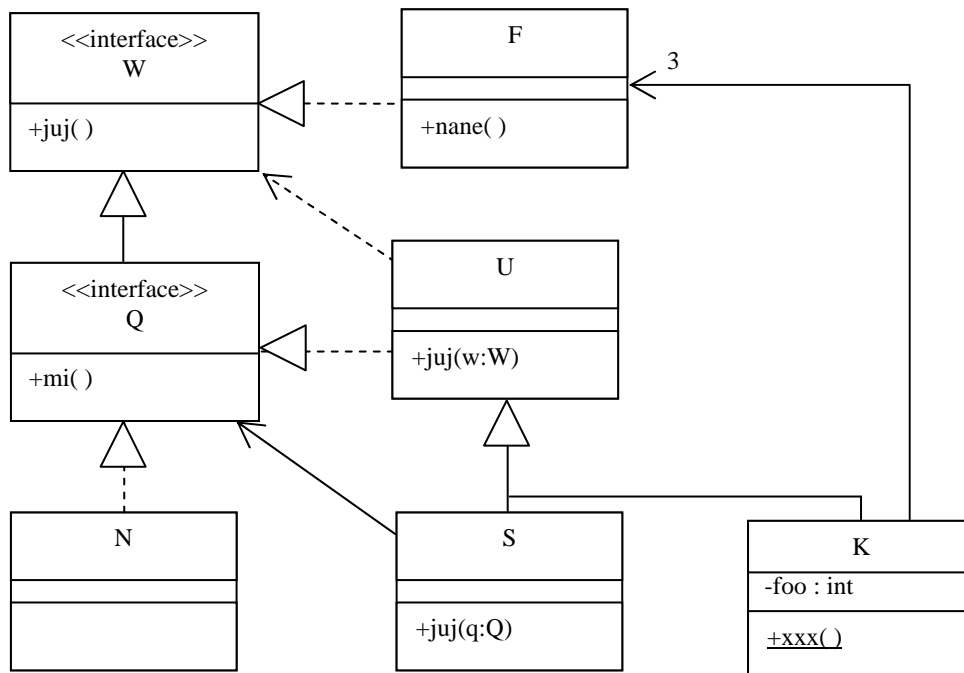
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2009. május 28. 13:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

[B] K helyettesíthető S-sel, mert közös ősük U.

[A] K juj(w:W) metódusa kaphat paraméterül S-t, mert K az F leszármazottja.

[B] K xxx() metódusa módosíthatja bármely K objektum foo attribútumát, mert a metódus statikus.

[B] F nem implementálja a juj() metódust, mert nem U leszármazottja.

[C] S juj(q:Q) metódusában meghívható egy paraméterül kapott N objektum mi() metódusa, mert N megvalósítja a W interfészt.

[E] S-nek nincs juj(w:W) metódusa, mert a juj(q:Q) metódusnak ugyanaz a szignatúrája.

[E] F helyettesíthető U-val, mert K mindkettejük leszármazottja.

[E] U juj(w:W) metódusából meghívhatjuk egy paraméterül kapott F nane() metódusát, mert F megvalósítja a Q interfészt.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a,c+)|b), a>
  <!ATTLIST a d CDATA #REQUIRED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <a d="|b"></a>
  <c>|b</c>
  <c>"|b"</c>
  <a d=|b></a>
</x>
```

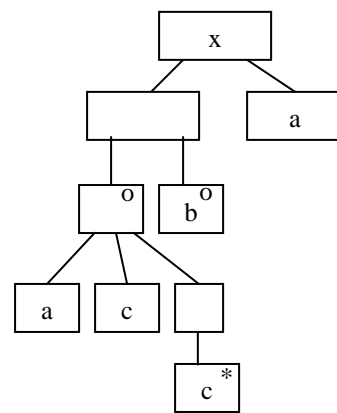
- IGEN
- NEM

Hiba ?: a második "a" attribútum paramétere nincs idézve.....

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van b elem ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <b>dddd</b>
  <a d="xxx"></a>
</x>
```

Rajzolja fel a DTD-nek megfelelő adatszerkezet Jackson-ábráját ! (2 pont)



3. Adja meg, hogy az alább felsorolt célok a JSD melyik lépését jellemzik ! Ha egy célhoz nem rendelhető JSD lépés, akkor azt „NEM”-mel jelölje ! (6 pont) Rossz válasz esetén pontot veszít !

- Az ELH-kat implementáló állapotgépek specifikálása NEM.....
- A modell-processzek hálózatának bővítése funkció processzekkel function
- Az ELH-k alapján processz-hálózat felvétele kezdeti modell.....
- Az entitás-élettörténetek kibővítése funkció-entitásokkal NEM.....
- Az entitások és események (akciók) azonosítása entitás-akció
- Az entitások funkcióinak implementálása NEM.....

4. A verziókezelésben mit jelentenek az alábbi fogalmak? (2 pont)

Check-out: **kezelt elem kiemelése felhasználásra a közös tárból**.....

Check-in: **kezelt elem visszahelyezése a közös táriba**

Milyen elvek szerint kezelik a párhuzamos hozzáférést ? (2 pont)

Reserved check-outs

Modify-update-merge

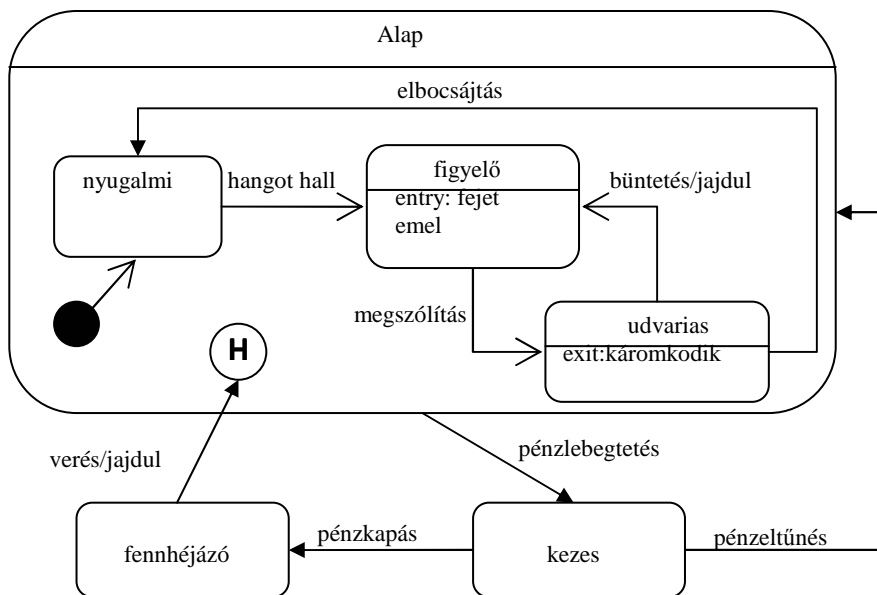
A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Felsoroltunk szofverrel kapcsolatos "hibákat". A hibák mellett jelölje be, hogy az melyik kategóriába tartozik! (4 pont) Rossz válasz esetén pontot veszít !

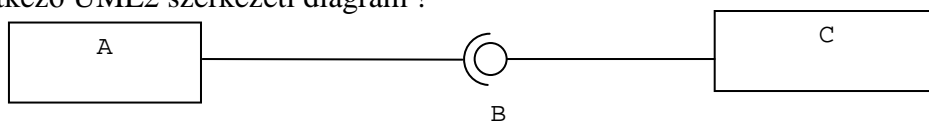
	bug	error	failure	fault
Az "==" helyett "=" áll a kifejezésben	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A weboldal nem jön le	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
A szerver memóriaigényét alulbecsülték	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A program lefagy, ha hibás bemenetet adunk meg	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Elmaradt a javított metódus tesztelése	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A function két paramétere fel van cserélve a kódban	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

6. Az alábbi történet alapján rajzoljon a diagramba UML 2 állapotábrát (state chart)! (6 pont)

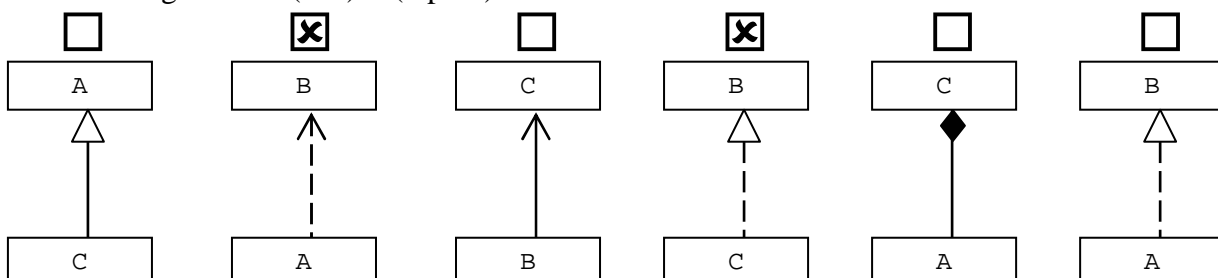
A Titanon létezik egy Lowyir-nek keresztelt parazita életforma. A Lowyir életét nyugalmi állapotban kezdi. Ha hangot hall, figyelő állásba lép (figyelő állás kezdetén mindig felemeli a fejét). Ha ekkor megszólítják, akkor udvarias lesz. Udvariasságából két módon lehet kimozdítani. Elbocsájtással, amire ismét nyugalmi állapotba kerül, vagy büntetéssel, ekkor megint figyelő állásba lép, de előtte feljajdul. Az udvarias állapotból való kikökkentéskor mindig elkáromkodik magát. Bármely fenti állapotban is volt, ha pénzt lebegtetnek meg előtte, akkor kezessé válik, ha a pénz eltűnik, a nyugalmi helyzetét veszi fel. Ha kezés és a pénzből kap, akkor fennhéjázó lesz. Fennhéjázása csak akkor szűnik meg, ha megverik, ekkor feljajdul, de aztán ott folytatja, ahol a pénz meglebegtetése előtt abbahagyta.



7. Legyen a következő UML2 szerkezeti diagram !

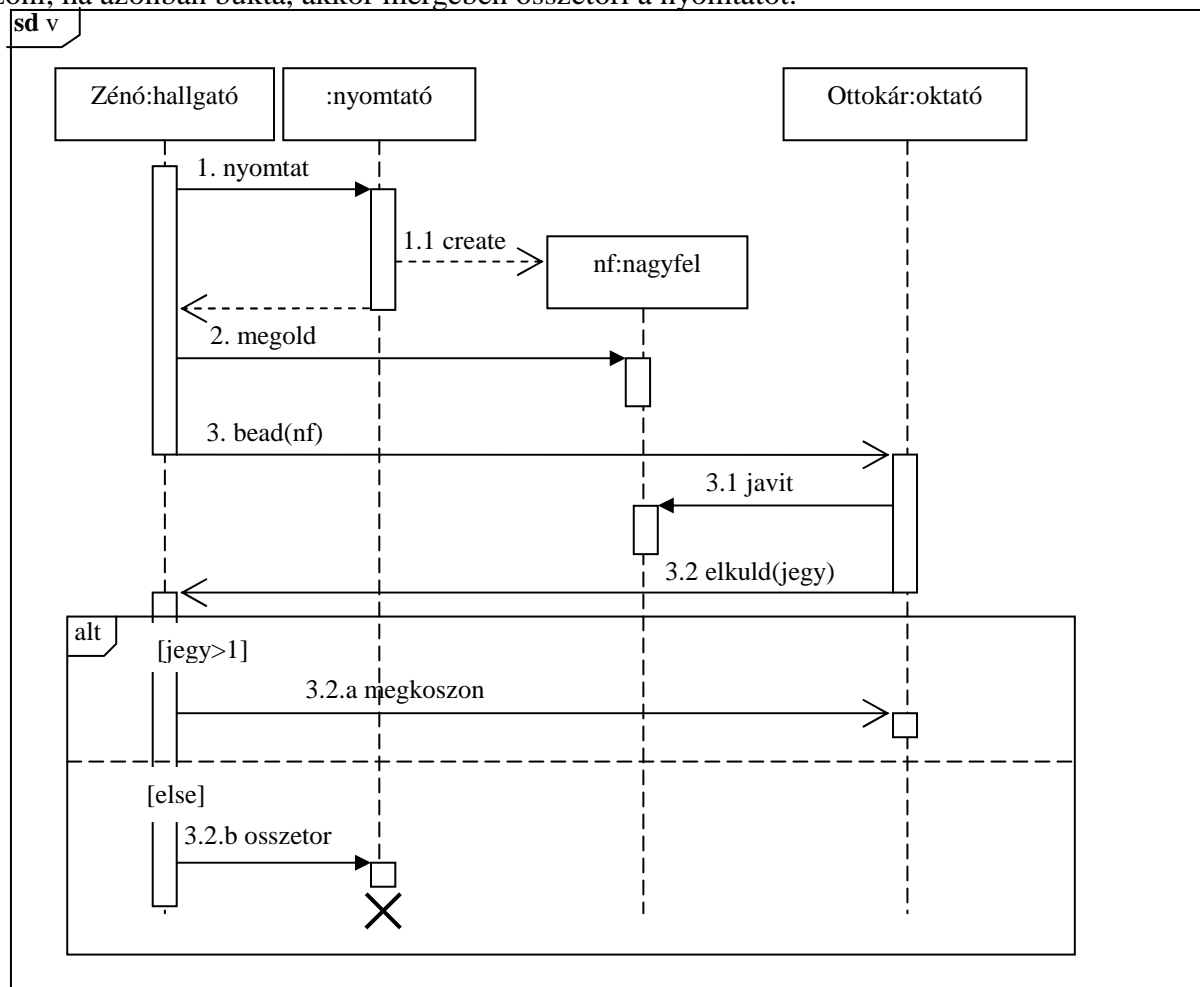


Feltételezve, hogy a fenti szerkezeti diagramon szereplő elemek között egyéb kapcsolat nincs, jelölje meg az alábbiak közül az igaz állítás(oka)t ! (4 pont)



8. A történet alapján rajzoljon UML 2.0 szekvenciadiagramot (sequence diagram). **Az üzeneteket hierarchikus számozással lássa el ! (8 pont)**

Zénó (Izidor bátyja) otthoni printerén kinyomtatja Bitgörbítés nagyfeladatát, a papíron megoldja, és elküldi Gyíkcú Ottokárnak, a tárgy oktatójának javításra. Ottokár egyből nekilát és kijavítja a feladatot, majd ezzel a lendülettel vissza is küldi a jegyet Zénónak. Zénó, ha jobb jegyet kap, mint elégtelen, akkor választ sem várva megköszöni, ha azonban bukta, akkor mérgében összetöri a nyomtatót.



9. Jellemezzünk egy stringet az alábbi műveletekkel ! Adja meg a PALIN műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja az egészek és a karakterek összehasonlítására szolgáló műveleteket. (4 pont)

- CRT()** új (üres) stringet hoz létre
- LGTH(s)** az s string karaktereinek számát adja
- TAIL(s)** az s string első karakterének levágása után maradó stringet adja
- APPEND(s,x)** az s string végére rakja az x karaktert
- HEAD(s)** az s string első karakterét mutatja meg
- PALIN(s)** igaz, ha az s string palindróma

Egy string palindróma, ha az elejéről olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

$$\text{PALIN}(\text{CRT}()) = \text{true}$$

$$\text{PALIN}(\text{APPEND}(s, x)) = (\text{LGTH}(s) == 0) \text{ or } (\text{PALIN}(\text{TAIL}(s)) \text{ and } (\text{HEAD}(s) == x))$$

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

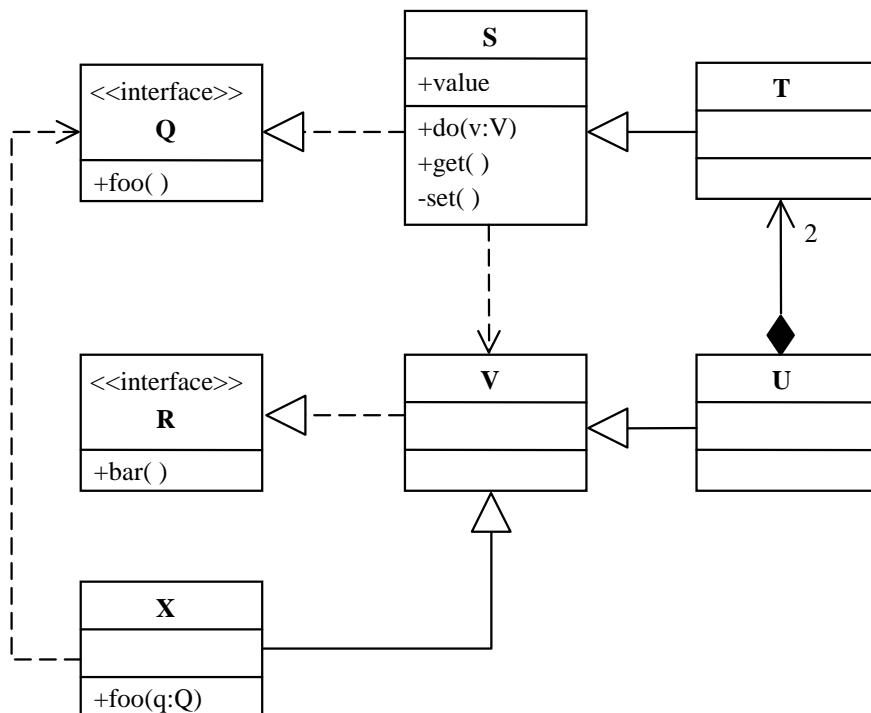
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2009. január 27. 14:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [A] S létrehozhat V osztályú objektumot, mert V függ az S-től.
- [D] X foo(q:Q) metódusa kaphat paraméterül T-t, mert T megvalósítja a Q interfészt.
- [B] X foo(q:Q) metódusa meghívhatja a paraméterül kapott S get() metódusát, mert S megvalósítja a Q interfészt.
- [D] T-ből legalább kétszer annyi példány van, mint U-ból, mert egy T példány nem tartozhat két U-hoz.
- [B] T meghívhatja U bar() metódusát, mert U-nak van bar () metódusa.
- [A] X meghívhatja egy Q interfészes objektum foo() metódusát, mert X implementálja Q-t
- [C] V helyettesíthető U-val, mert mindketten megvalósítják az R interfészt
- [B] S set() metódusa nem módosíthatja a value attribútumot, mert a láthatóságuk különböző

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

gyakorlati problémák megoldására szolgáló **dolgok**

a **társadalom** által szabályozott **tudományos** alapokon álló

gazdaságos
előállításának mikéntje

3. Mi a lényege a JSD-ben a “funkció” lépésnek (function step) ? (2 pont) Rossz válasz esetén pontot veszít !

- Az entitások funkcióinak specifikálása
- A funkció processzek élettörténetének tervezése
- A funkció processzek ütemezése
- Az entitások halmazának kibővítése funkció-entitásokkal
- A funkcionális dekompozíció
- A funkció processzek hozzáadása a modell-processzek hálózatához.

Mi az **előző** lépés? (1 pont)

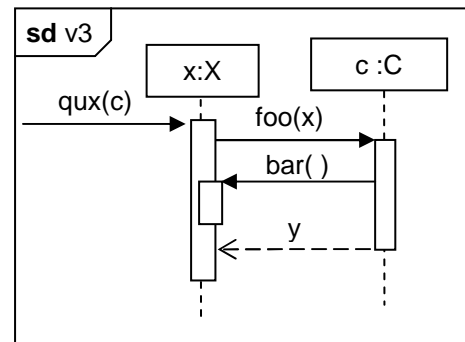
initial model

Mi a **következő** lépés? (1 pont)

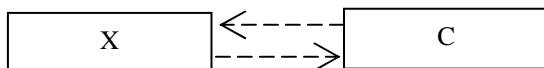
timing.....

4. Tételezze fel, hogy az alábbi (v3 nevű) UML2 szekvenciadiagramon szereplő objektumok osztályai között nincs más egyéb – a diagramból nem kiolvasható – kapcsolat (pl. öröklés) ! Mi a kapcsolat X és C között ? (2 pont)

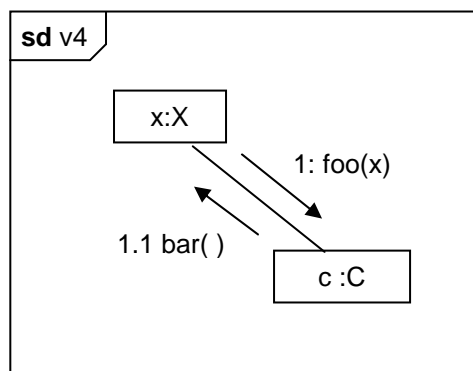
- implementálás (implementation)
- kollaboráció (collaboration)
- függőség (dependency) C függ X-től
- függőség (dependency) X függ C-től
- asszociáció (association)
- példányosítás (instantiation)
- interakció (interaction)



Rajzolja be választását az alábbi osztálydiagramba ! (2 pont)



5. Rajzolja le a fenti (v3 nevű) szekvenciadiagramnak megfelelő UML2 kommunikációs diagramot ! (3 pont)

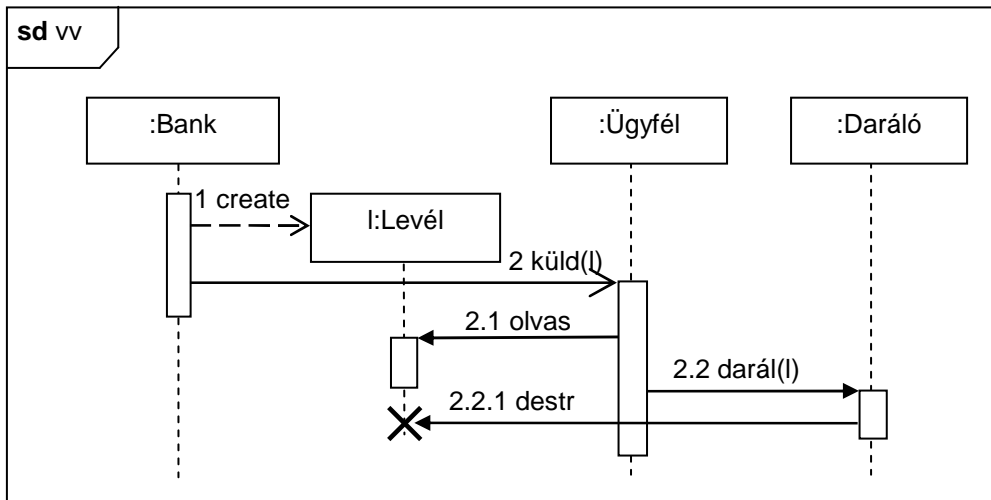


A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

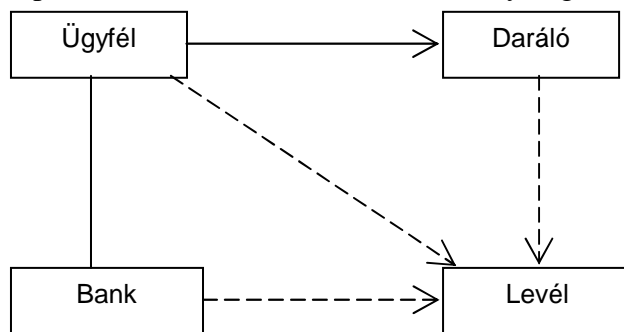
6. A specifikáció célja a követelményeknek eleget tevő rendszer formális leírása. Milyen három fontos nézőpontból készítjük a leírásokat ? (3 pont) !

funkcionalitás szerkezet (struktúra)
 dinamika (viselkedés).....

7. Rajzoljon UML2 **szekvenciadiagramot** ! Az üzeneteket hierarchikus számozással lássa el ! (6 pont)
 Az InterCredit Bank felszólító levelet ír, amelyet elküld egyik ügyfelének, Gerzsonnak, akinek hiteltartozása van. Gerzson a levelet elolvassa, majd a darálón ledarálja.



Tételezze fel, hogy a fenti történetben szereplő objektumok osztályai között nincs más egyéb – a történetből nem kiolvasható – kapcsolat (pl. öröklés) ! Az alábbi UML2 osztálydiagramba rajzolja be az osztályok közötti kapcsolatokat ! (2 pont)



8. Elkészítjük az alábbi Y osztály két példányát, y1-et és y2-t. Ezt követően végrehajtjuk a következő műveleteket:

```

y2.a = 3; y1.a = -3;
y1.b = y2.a + 4;
y2.b = y2.a + y1.b;

```

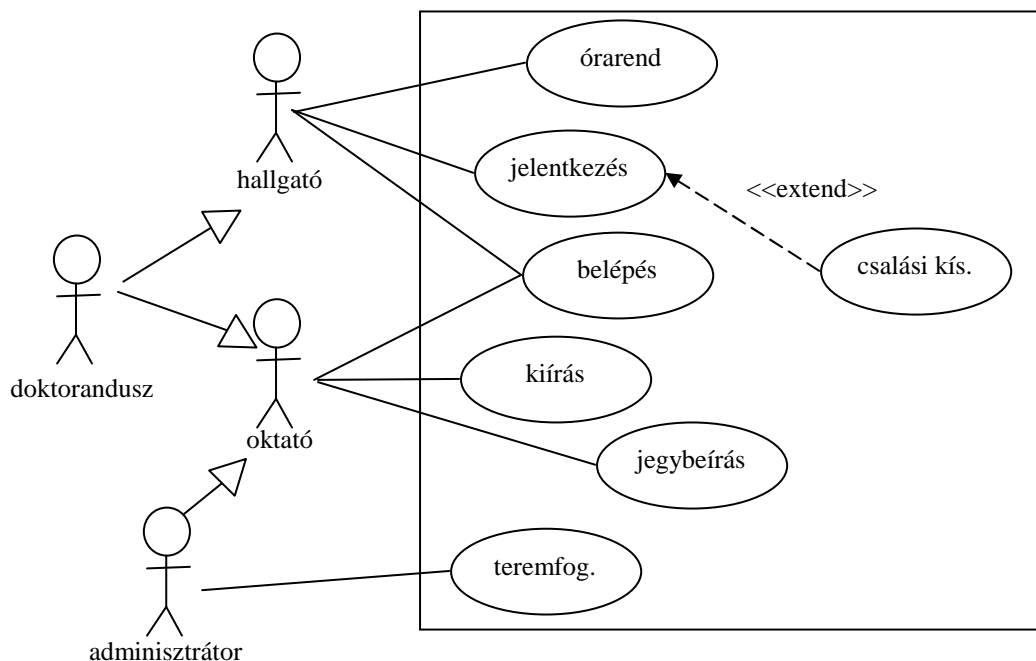
Mennyi lesz a változók értéke ? (4 pont)

y1.b = 1 y2.b = -2.....

Y
<u>int a = 5</u> int b = 2
private aX(): int

9. Rajzoljon UML 2 use-case diagramot az alábbi leírás alapján! (7 pont)

A Poseidon rendszert egyetemi oktatók és hallgatók használhatják a tanulmányi adatok rögzítésére. A hallgatók bejelentkezhetnek, vizsgára jelentkezhetnek és órarendet nyomtathatnak. Ha vizsgára jelentkezés közben kiderül, hogy valamely előfeltétel nem teljesül, akkor a rendszer rögzíti a csalási kísérletet. Az oktatók is belépnek, vizsgát írhatnak ki, illetve vizsgaeredményeket írhatnak be. Az adminisztrátorok az oktatók lehetőségein túl még termet is foglalhatnak. A doktoranduszok, mivel oktatniuk és tanulniuk is kell, mind oktatóként, mind hallgatóként használhatják a rendszert.



10. Az üzleti folyamatok lefutásának két esete az (1) orkesztráció (orchestration) és a (2) koreográfia (choreography). Mi a két megközelítés közötti különbség? (4 pont)

(1) orkesztráció (orchestration) – egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.

(2) koreográfia (choreography) – a folyamat nincs központosítva, minden résztvevő a dolgát elvégezve az általa ismert következő résztvevő(ke)t aktivizálja.

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2004. december 21.

1. Az UML komponensek fajtái: (3 pont)

deployment (a sw indításához kell, pl class)

execution (a végrehajtás közben keletkezik, pl file)..

work product (a fejlesztő munka terméke, pl *.h) .

.....

minden helyes válasz 1 pont

2. Jelölje meg a funkciópont elemzés (Function Point Analysis, FPA) komponenseit ! (5 pont)

External Inputs

Internal Data Flows

Internal Logical Files

External Requirements

External Outputs

External Enquiry

External Data Flows

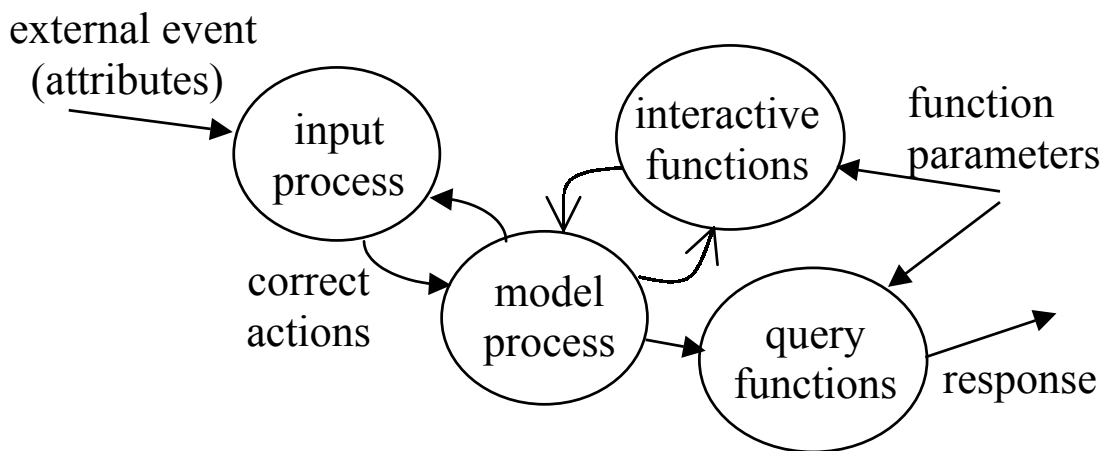
Internal Logical Interfaces

External Interface Files

External Dialog

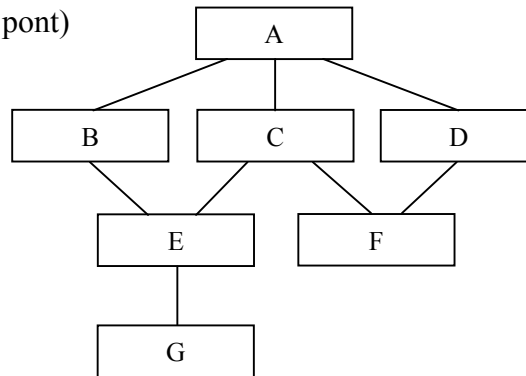
minden helyes válasz 1 pont, minden hiba -1 pont, negatív nem lehet

3. Rajzolja fel DFD ábrán, hogy a JSD-ben a modell és funkció processzek hogyan kapcsolódnak egymáshoz és a külvilághoz. (6 pont)



mindegyik item (gombóc és felirat) 0,5 pont

4. Adott a mellékelt struktúra diagram. Mennyi a (3 pont)



C modul fan-out-ja ? : 2

E modul fan-in-je ? : 2

Minden jó válasz 1,5 pont

5. Készítsen **entitás-relációs modellt** az alábbi problémára! (6 pont)

A kari TDK konferenciára száznál több dolgozatot nyújtottak be a szerzők. A dolgozatot jellemzi a címe, oldalszáma, és a konferencián szerzett díj. Egy dolgozatnak több szerzője lehet, egy szerző több dolgozatban is érdekelt lehet. A szerzőre jellemző a neve, neptun-kódja és az évfolyama. A dolgozatok oktatók irányításával készültek. Egy oktató több dolgozatot is konzultálhatott, de nem mindegyik oktátónak ért el olyan eredményt a hallgatója, amely eredmény alapján dolgozat született. Van olyan dolgozat is, aminek több konzulense is van. Az oktató jellemzője a neve, beosztása és tanszéke. A bírálókat objektivitása érdekében minden dolgozatot pontosan két egyetemen kívüli szakértővel bíráltatnak el. Egy szakértő (név, cég, telefon) csak egyetlen dolgozatot bírál. Minden dolgozatot a témája alapján egy névvel ellátott szekcióba sorolják; egy szekcióba 7-10 dolgozat kerül. A szekciók ülésére különböző termekben, de egyidőben kerül sor. Egy szekció munkáját az elnök irányítja, aki egyetemi oktató. A szekció munkájában szekciónként részt vesz még legalább egy oktató és egy a bírálásban is érintett szakértő. A szekció ülésén tartott előadást is figyelembe véve a dolgozatoknak a szekció díjat adományoz.

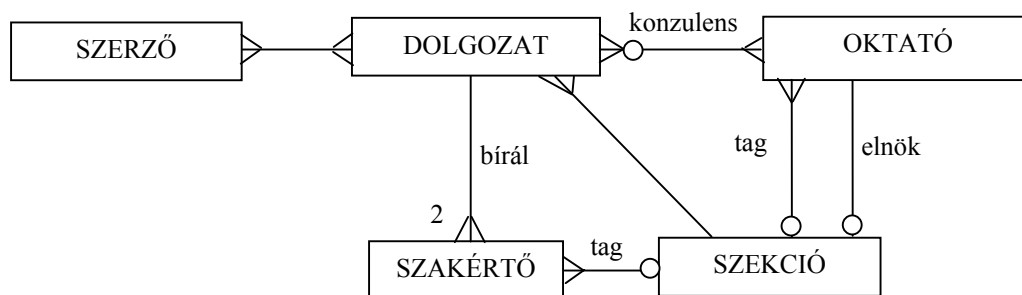
SZERZŐ (név, neptun-kód, évfolyam)

DOLGOZAT (cím, oldalszám, díj)

OKTATÓ (név, beosztás, tanszék)

SZAKÉRTŐ (név, cég, telefon)

SZEKCIÓ (név, terem)



Minden entitás és reláció 0,5 pont

6. Készítsen **JSD** modellt az alábbi problémára ! (11 pont)

A Gyakorlati Oximorisztika c. tárgyat felvevő hallgatók akkor teljesítik a félévközi követelményeket, ha beadják a nagyfeladatukat és a két labormérést elvégzik. A feladatok megoldásához a hallgatók a tanárok által tartott konzultációkon kaphatnak segítséget. A második mérésre csak az mehet, aki az első mérést sikeresen elvégezte. Mindkét sikertelen mérés tetszőlegesen sokszor pótolható. A laborok teljesítésével párhuzamosan zajlik a feladat elkészítése. A konzultáló tanárok a félév során gyakran továbbképzéseken vesznek részt.

Definiálja az alábbi funkciókat ellátó processzeket:

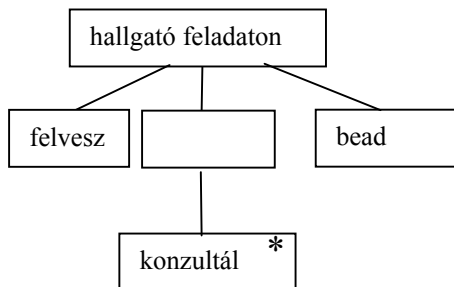
Azonnal jelezzen, amikor egy tanárnál egy továbbképzés után az első hallgató megjelenik konzultációra!
Kérésre listázza ki azokat a hallgatókat, akik az első labort már, de a másodikat még nem teljesítették és eddig legalább háromszor pótmértek !

Kérésre adja meg azokat a hallgatókat, akik beadták a feladatukat és legalább három alkalommal voltak konzultálni és mindkét labort teljesítették és nem volt pótmérésük !

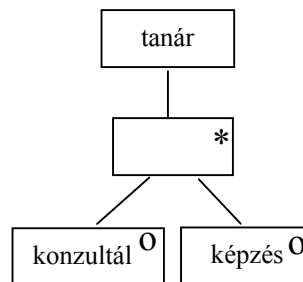
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
kapcsolja a modellhez a **funkció processzeket.**

- felvesz (hallgató feladaton, hallgató laboron) 2 pont
- konzultál (hallgató feladaton, tanár)
- bead (hallgató feladaton)
- sikertelen labor (hallgató laboron)
- labor1 (hallgató laboron)
- labor2 (hallgató laboron)
- képzés (tanár)

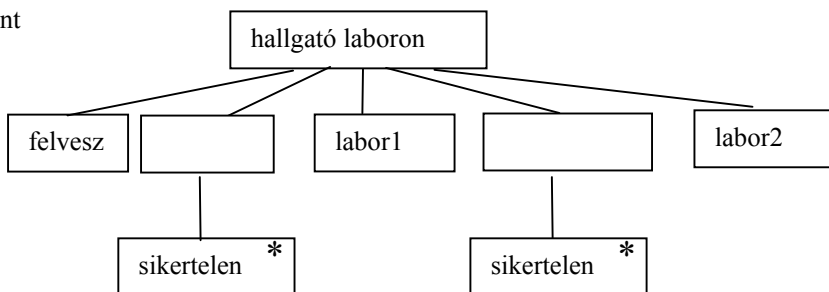
2 pont



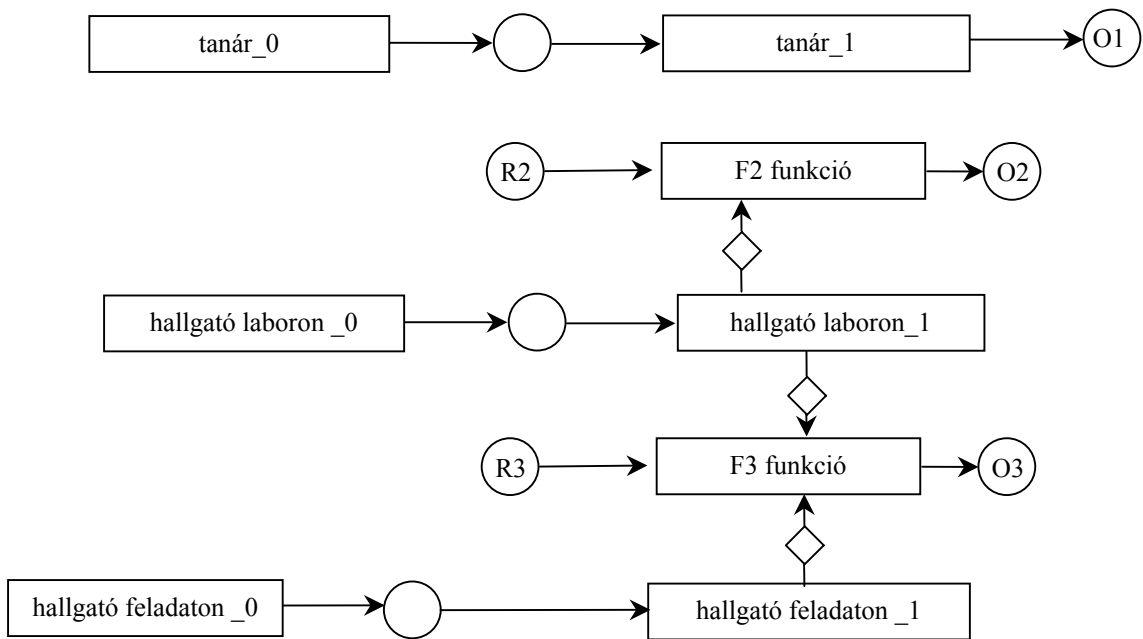
2 pont



3 pont

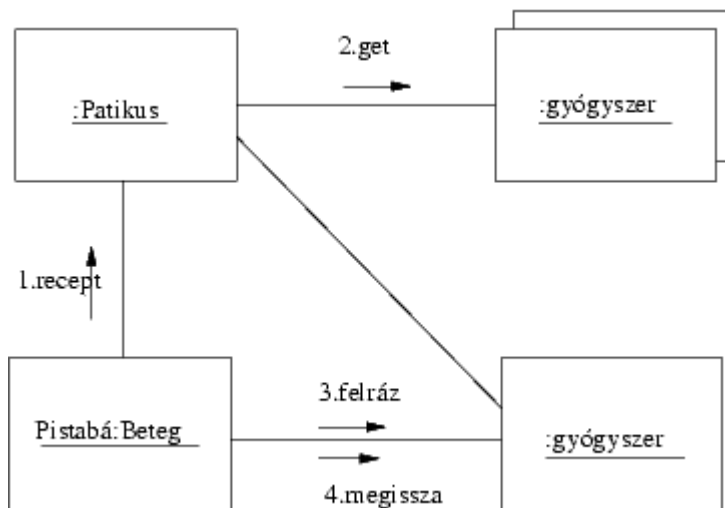


SSD 2 pont



7. Az alábbi történethez készítsen UML kollaborációs diagramot! (8 pont)

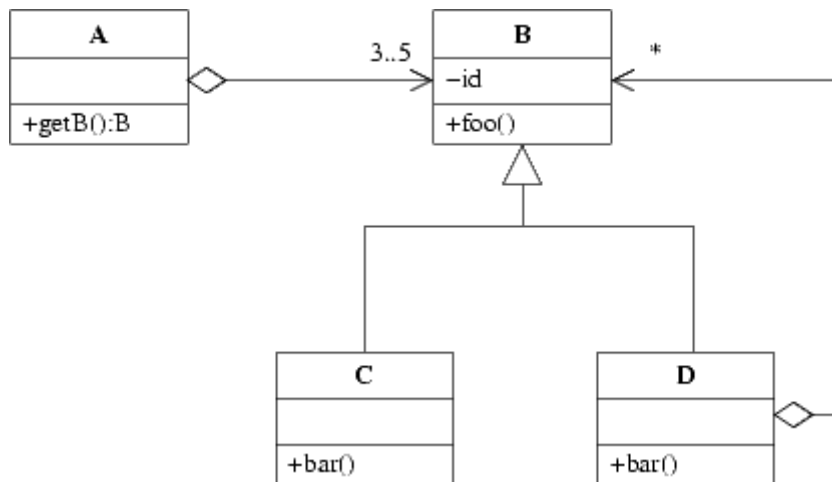
Pista bácsi átadja receptjét a patikusnak. Utóbbi a gyógyszerek közül kiválasztja a megfelelőt, és ezt odaadja a várakozó bácsinak. Ő felrzza a gyógyszert és azután megissza.



multiobject 2pt

minden egyéb helyes objektum és üzenet 1pt

8. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)
 (F - az ki van zárva, mert nem tudom)

- [B] A meghívhatja C *bar()* metódusát, mert B minden leszármazottjában van *bar()*
 [B] D ismeri C *bar()* metódusát, mert közös az ősük.
 [E] A látja B *id* attribútumát, mert a kapcsolat B-ből A-ba irányított.
 [E] C helyettesíthető B-vel, mert B a C leszármazottja.
 [B] A *getb()* metódusa nem adhat vissza C osztályú objektumot, mert C nem ismeri B-t.
 [B] C helyettesíthető D-vel, mert D ismeri C ősét.
 [C] D nem látja A *getB()* metódusát, mert mind A, mind D tartalmazhat B-t.
 [C] B *foo()* metódusa látja B *id* attribútumát, mert *foo()* publikus.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

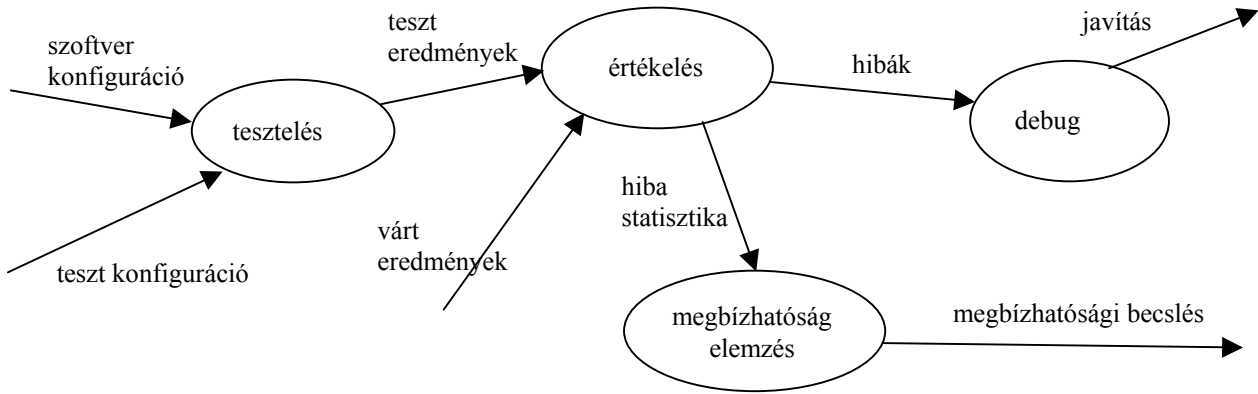
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2005. január 4.

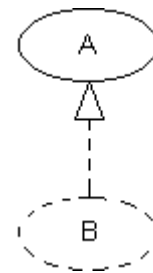
1. Rajzolja fel a tesztelés információs folyamatának adatfolyamábráját ! (6 pont)



mindegyik gombóc és él 0,5 pont.

2. Mi A és B a jobbra látható UML diagramon ? (2 pont)

- | | |
|--|--|
| A | B |
| <input type="checkbox"/> osztály | <input type="checkbox"/> leszármazott osztály |
| <input checked="" type="checkbox"/> use-case | <input type="checkbox"/> leszármazott use-case |
| <input type="checkbox"/> kollaboráció | <input checked="" type="checkbox"/> kollaboráció |
| <input type="checkbox"/> komponens | <input type="checkbox"/> kollaborációs diagram |



minden helyes válasz 1 pont, hiba -1 pont, negatív nem lehet

3. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

gyakorlati problémák megoldására szolgáló **dolgok**

a **társadalom** által szabályozott **tudományos** alapokon álló

gazdaságos

előállításának mikéntje

mindegyik komponens 1 pont.

4. Programom 12 darab C nyelven írt function-ból áll. A 12 function-t a function-ok neve szerint abc sorrendbe rakom, majd három modulra vágom szét. Milyen a modulok kohéziója ? (2 pont)

szekvenciális

logikai

esetleges

temporális

helyes válasz 2 pont, minden rossz -1 pont, de ≥ 0

Sorolja fel azokat a kohéziós típusokat, amelyek a fenti listában nem szerepelnek ! (2 pont)

kommunikációs vagy **funkcionális** vagy **procedurális**.....

minden helyes 1 pont, aki mindhármát megadja annak összesen 3 pont, rossz -1 pont, de total ≥ 0

5. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet ! Az axiómák felírásakor megengedett a két karaktert összehasonlító művelet használata. (6 pont)

CRT() új (üres) stringet hoz létre.

ADD(s, x) az s string végére rakja az x karaktert.

LAST(s) az s string végén álló karaktert adja.

END(s1, s2) igaz, ha az s2 string az s1 string végén áll.

DUPLO(s) igaz, ha az s stringben legalább egyszer dupla karakter fordul elő. Például: aggódik

LAST(CRT()) = undefined 0,5 pont

LAST(ADD(s, x)) = x 0,5 pont

END(s, CRT()) = true 1 pont

END(CRT(), ADD(s, x)) = false 1 pont

END(ADD(s1, x1), ADD(s2, x2)) = (x1 == x2) and END(s1, s2) 1 pont

DUPLO(CRT()) = false 1 pont

DUPLO(ADD(s,x)) = (x == LAST(s)) or DUPLO(s) 1 pont

6. Készítsen **JSD** modellt az alábbi problémára ! (11 pont)

A művház könyvtárába bárki beléphet, aki kitölti a belépési nyilatkozatot. A tagsági díj befizetésével a tag egy éven keresztül kölcsönözhet. Ha az év letelik, akkor újra tagdíjat kell fizetni, ha újabb könyvet akar kölcsönözni. A tagdíj mértéke függ attól, hogy az elmúlt évben a kölcsönzések hány százalékában hozta vissza a tag a könyvet a kölcsönzési határidő lejárta után. A könyv a megvételtől a selejtezésig áll a tagok rendelkezésére. Egy könyv kölcsönzése tetszőlegesen sokszor meghosszabbítható, ha arra az aktuális határidőn belül sor kerül. Ha a határidő lejár, akkor a tagot SMS-ben figyelmeztetik és tovább már nem hosszabbíthatja ezt a kölcsönzést, a könyvet vissza kell hoznia.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha olyan tag akar kölcsönözni, akinél esedékes a tagdíj megfizetése !

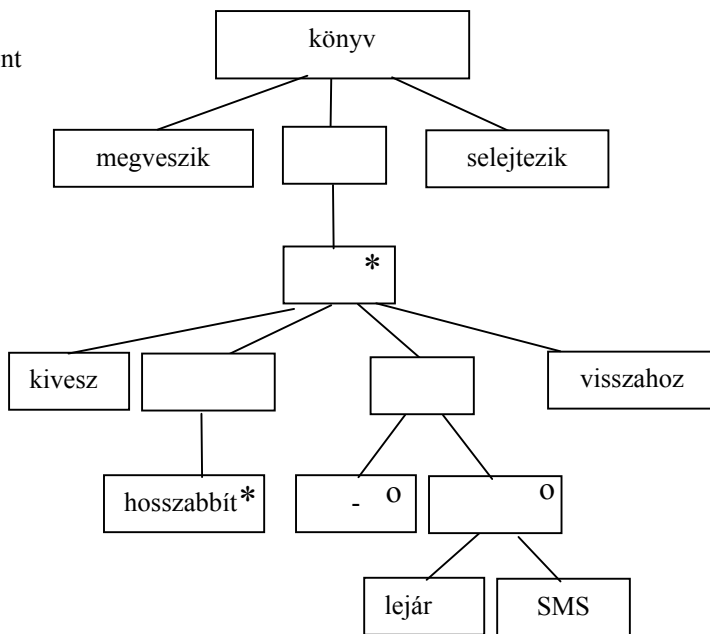
Kérésre listázza ki azokat a tagokat, akik a kölcsönzések több mint 30%-ában késedelmesen hozták vissza a kölcsönzött könyvet !

Kérésre listázza ki az új tagok által a leggyakrabban kért könyveket !

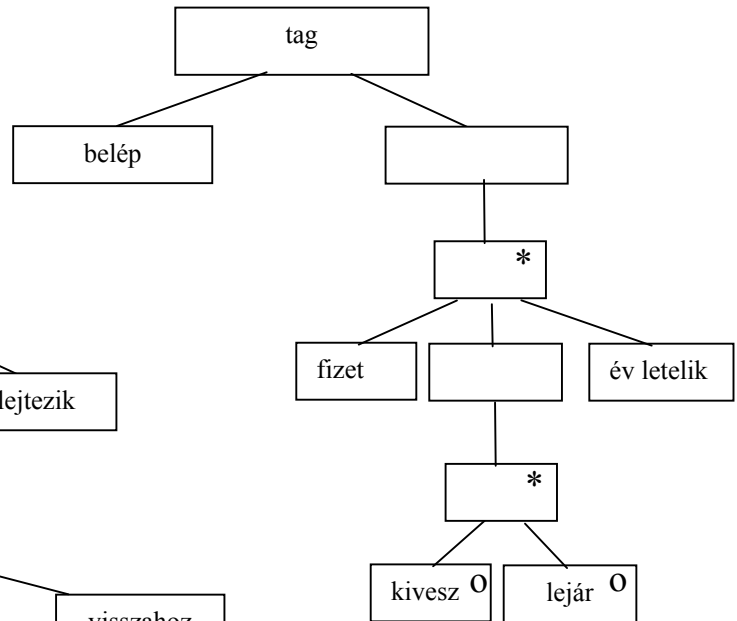
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
 kapcsolja a modellhez a **funkció processzeket.**

- belép (tag) 2 pont
- fizet (tag)
- év letelik (tag)
- megveszik (könyv)
- selejtezik (könyv)
- kivesz (könyv, tag)
- hosszabbít (könyv)
- lejár (könyv, tag)
- SMS (könyv)
- visszahoz (könyv)

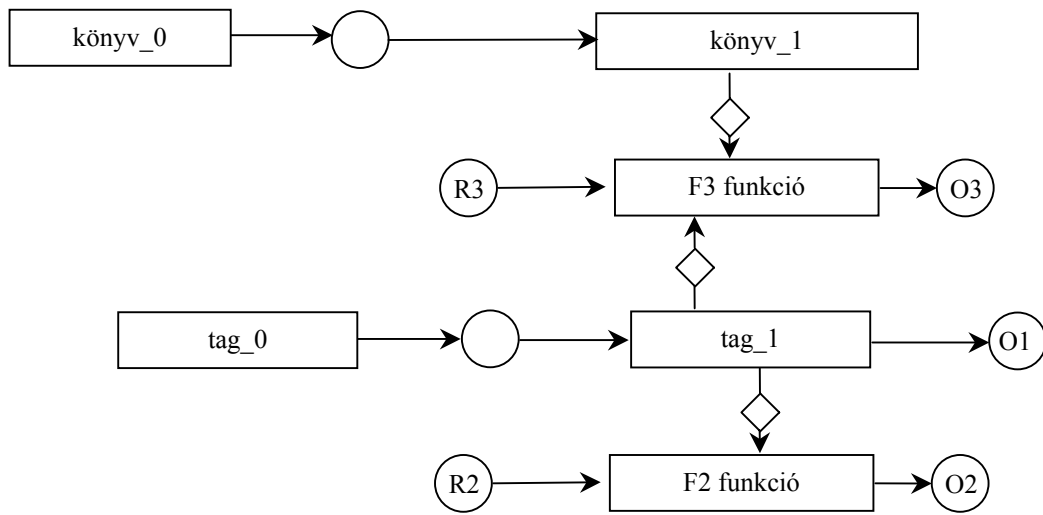
4 pont



3 pont

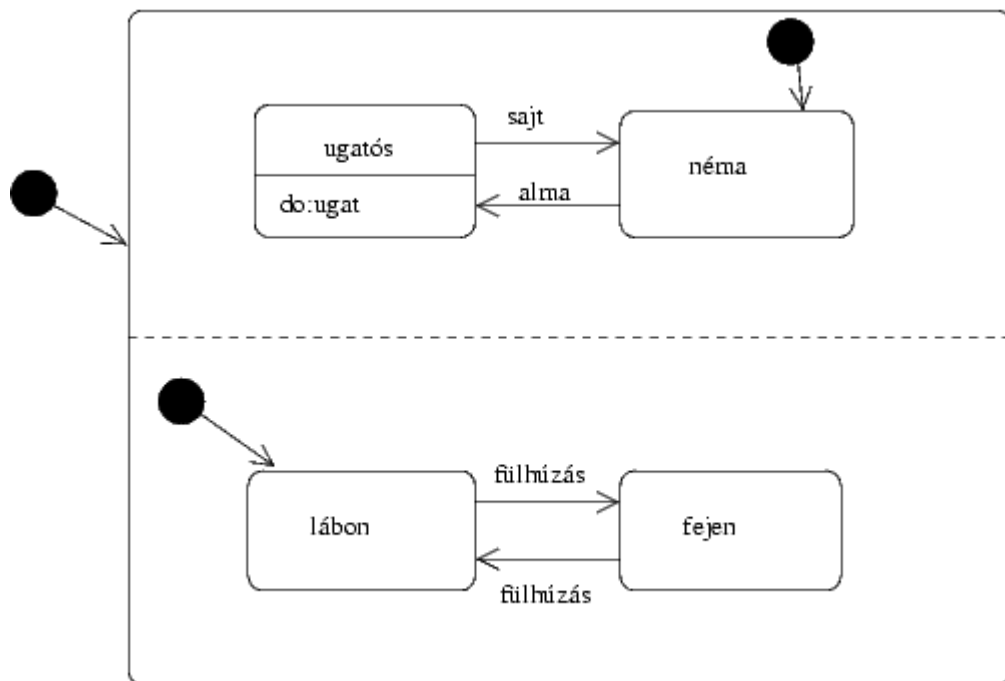


SSD 2 pont



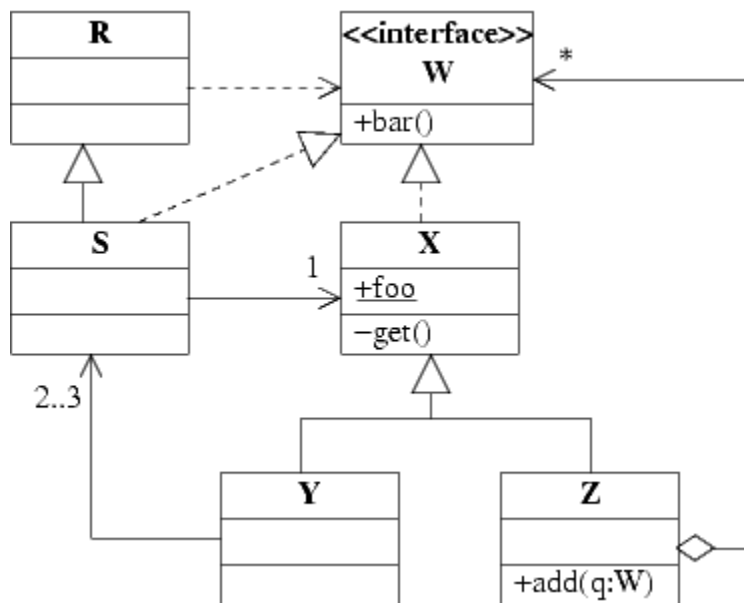
7. Készítsen az alábbi leírás alapján UML állapotábrát (statechart)! (8 pont)

II. Frigyes udvari koboldja alapesetben néma. Ha almát kap, elkezd ugatni, ha ekkor sajtot kap, ismét megnémul. Ha bármikor meghúzzák a fülét, akkor fejre áll, újabb fülhúzásra ismét lábra áll. Mikor megszületett, lábon állt.



Minden helyes állapot és átmenet 0,5 pont, *do* 1 pont, konkurencia 1,5 pont.

8. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)
 (F - az ki van zárva, mert nem tudom)

- [D] R helyettesíthető S-sel, mert az S osztály az R leszármazottja.
 [D] Z aggregálhat S-t, mert S megvalósítja a W interfészt.
 [C] S módosíthatja X foo attribútumát, mert az attribútum statikus.
 [D] Előfordulhat, hogy Z egy W interfészű objektumot sem aggregál, mert a '*' megengedi a nullát is.
 [C] S ismeri a W interfészt, mert kapcsolatban áll egy megvalósításával.
 [A] S nem látja X get metódusát, mert a metódus *protected*.
 [C] S ismeri X-t, mert ugyanazt az interfészt valósítják meg.
 [D] Z add metódusa kaphat paraméterül Y-t, mert Y (közvetetten) megvalósítja a W interfészt.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK

PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2002. december 17.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet ! (6 pont)

CRT()	új (üres) stringet hoz létre.
LGTH(s)	az s string karaktereinek számát adja.
TAIL(s)	az s string első karakterének levágása után maradó stringet adja.
ADD(s,x)	az s string végére rakja az x karaktert.
HEAD(s)	az s string első karakterét mutatja meg.
PALIN(s)	igaz, ha az s string palindróma.

Egy string palindróma, ha az elejéről olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

$LGTH(CRT()) = 0$	- 0.5 pont, ha hiányzik
$LGTH(ADD(s, x)) = LGTH(s) + 1$	- 0.5 pont, ha hiányzik
$HEAD(CRT()) = \text{undefined}$	- 0.5 pont, ha hiányzik
$HEAD(ADD(s, x)) = \text{if } (LGTH(s) == 0) \text{ x else HEAD}(s)$	1 pont
$TAIL(CRT()) = CRT()$	- 0.5 pont, ha hiányzik
$TAIL(ADD(s, x)) = \text{if } (LGTH(s) == 0) \text{ s else ADD}(TAIL(s), x)$	1 pont
$PALIN(CRT()) = \text{true}$	1 pont
$PALIN(ADD(s, x)) = (LGTH(s) == 0) \parallel (PALIN(TAIL(s)) \&\& (HEAD(s) == x))$	3 pont

2. Írja fel egy operáció szignatúrájának általános formáját ! (6 pont)

[oneway] <op_type_spec> <identifier> (param1, ..., paramL) [raises(except1, ..., exceptN)]
[context(name1, ..., nameM)]

3. Egy modul egy másik modul használata során record (vagy struct) paramétert ad át. Milyen a két modul közötti csatolásban (coupling) a "kommunikáció fajtája" (kind of communication) ? (2 pont)

- | | |
|--|---|
| <input type="checkbox"/> Content (tartalom) | <input type="checkbox"/> Common (közös) |
| <input checked="" type="checkbox"/> Stamp (bélyeg) | <input type="checkbox"/> Control (vezérlés) |
| <input type="checkbox"/> Data (adat) | <input type="checkbox"/> Logic (logika) |

4. Legyen egy PtVizsga osztályunk, amelynek operációi a mellékelt táblázatban X-szel jelölt változókat használják.

PtVizsga
-datum
-hely
-csoport
-sorszám
-létszám
+meghirdet ()
+javít ()
+osztályoz ()

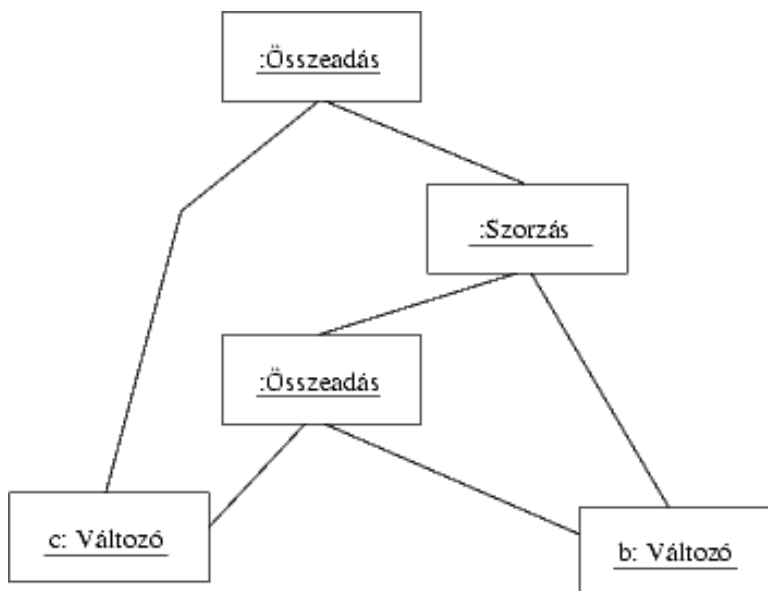
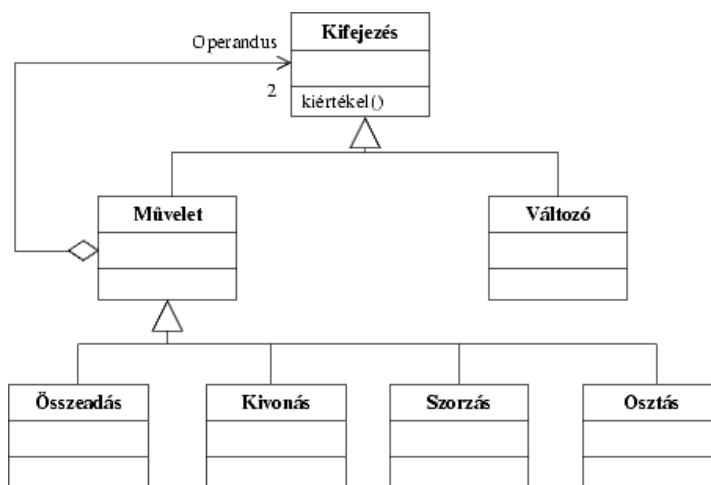
	datum	hely	csoport	sorszám	létszám
meghirdet ()	X	X		X	
javít ()			X		X
osztályoz ()			X		

Mennyi az LCOM (Lack of COhesion Metric) ? (2 pont)



1

5. Az ábrán egy - kifejezéseket leíró - osztálydiagram látható. Egy kifejezés művelet vagy változó lehet. Egy műveletnek két kifejezés típusú operandusa van. Rajzoljon UML objektum diagramot a $(b * (b + c)) + c$ kifejezés leírására ! (8 pont)



Minden objektum 1 pont és minden link 0.5 pont

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A Hírharsona c. lap hirdetések is közöl. A hirdetés feladója megrendeli, hogy a hirdetés hány egymást követő napon jelenjen meg. A megjelenések száma minimum 2. A hirdetések megjelentetését az első megjelenést követően a feladója lemondhatja. A lemondást csak akkor fogadják el, ha a lemondáskor még legalább két hirdetési alkalom van hátra, ugyanis a nyomdai előkészítés átfutása miatt a lemondást követő napon még megjelenik a hirdetés. Amikor a hirdetés lezárult, a lap a hirdetés feladójának küldött levélben köszöni meg, hogy a lapban hirdetett. A lap előfizetőinek kedvezményes hirdetési díjat számítanak fel. A kedvezmény mértéke függ a hirdetések számától is. Természetesen a lap előfizetését is le lehet mondani.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha egy előfizető hirdető ebben a hónapban már másodszor ad fel hirdetést !

Kérésre készítse el a köszönőleveleket az éppen esedékes ügyfeleknek !

Kérésre listázza ki azon hirdetéseket, amelyek lemondás miatt holnap jelennek meg utoljára !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
 kapcsolja a modellhez a **funkció processzeket.**

megrendel (hirdetés, ügyfél) eseménylista 3 pont (0.5 pont darabonként)

megjelenik (hirdetés)

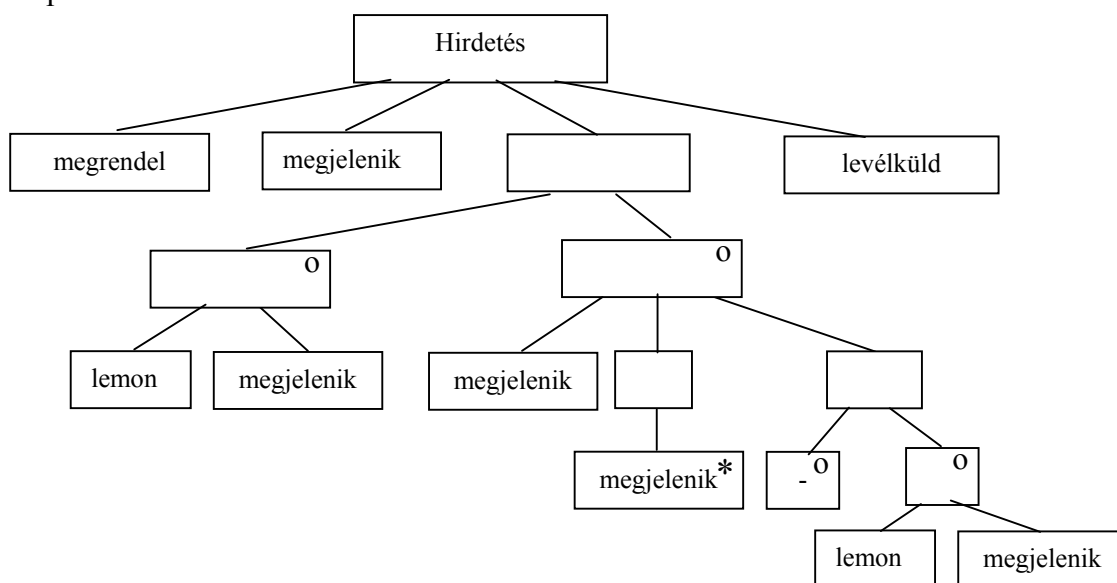
lemond (hirdetés)

levélküld (hirdetés)

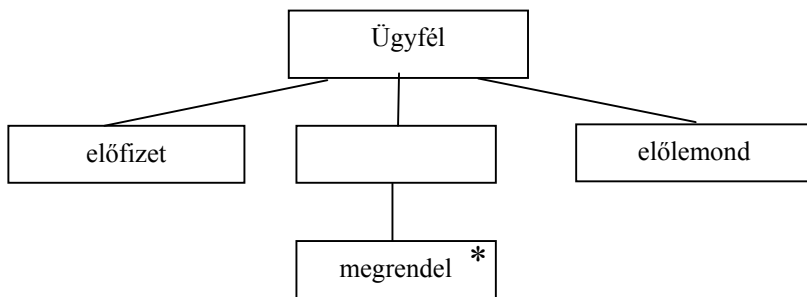
előfizet (ügyfél)

előlemond (ügyfél)

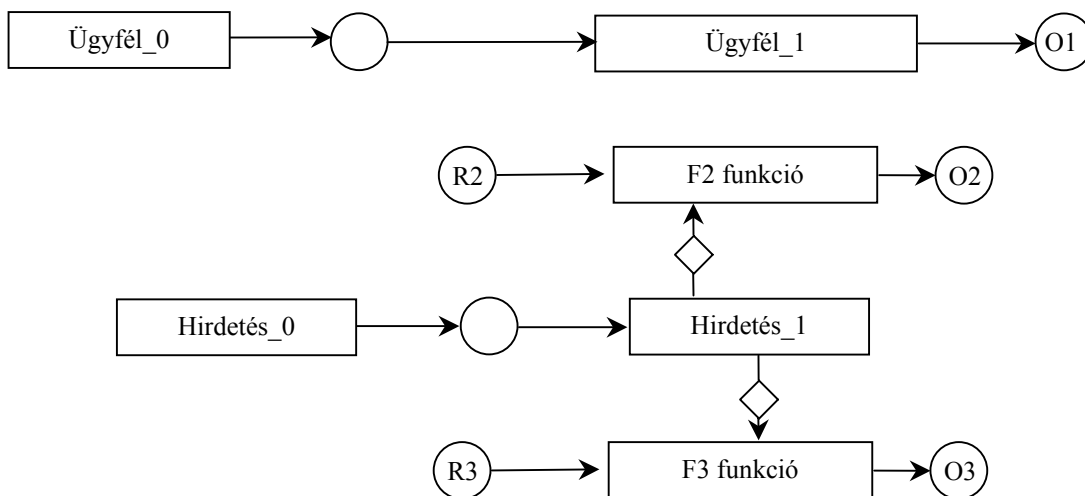
Hirdetés entitás 4 pont



Ügyfél entitás 2 pont



teljes SSD 2 pont

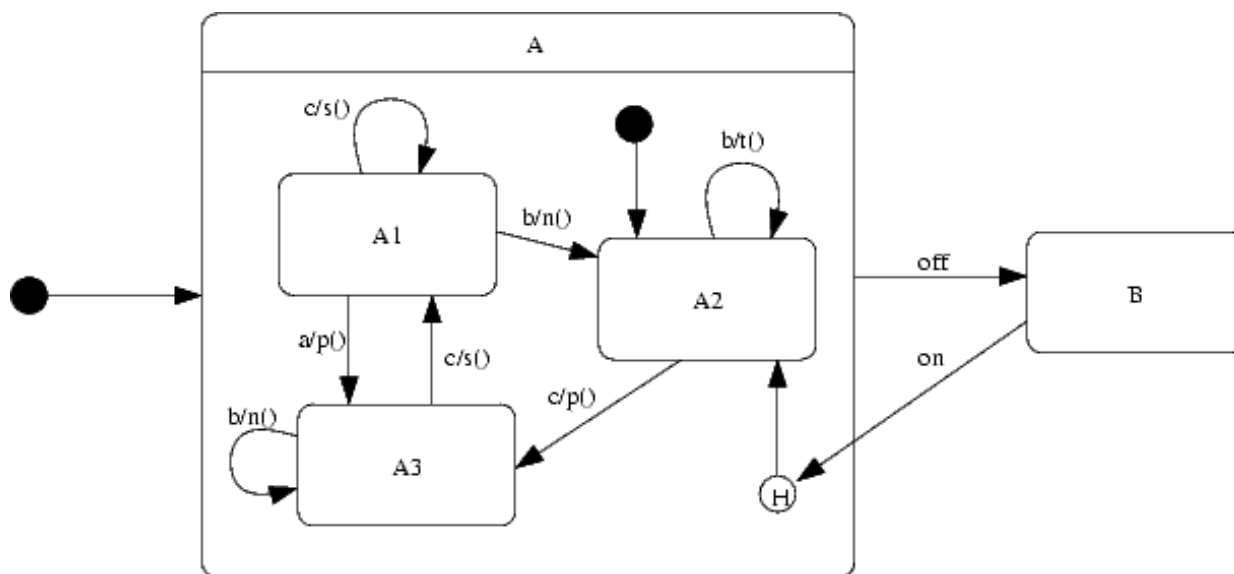


7. Legyen egy objektumunk, amely két fő állapottal (A, B) rendelkezik. Az A állapotban a következő állapotgép működik:

	a	b	c
A1	A3/p()	A2/n()	A1/s()
A2	-	A2/t	A3/p()
A3	-	A3/n()	A1/s()

Az A állapotból B-be az `off` esemény hatására kerül. Visszatérni az `on`-ra fog, és ekkor ott folytatja, ahol a kilépéskor abbahagyta. Kezdetben az A állapot aktivizálódik, az A-n belül pedig az A2. Ugyancsak A2 az `on`-ra vonatkozó predefinit állapot.

Rajzolja meg a objektum **UML state-chart**-ját ! (7 pont)



off – 1 pont

on H-ba – 1 pont

H- A2 – 1 pont

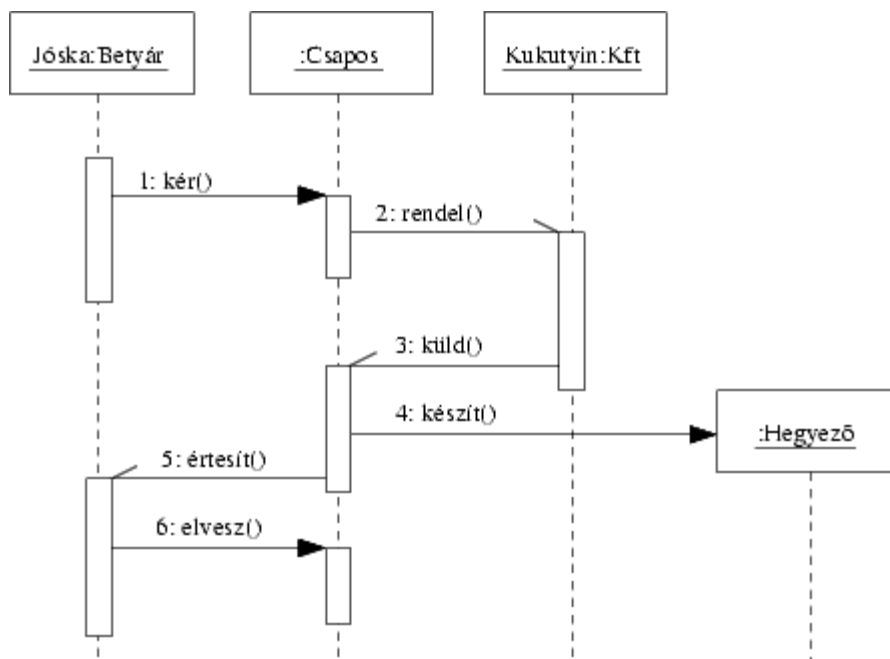
A2-re indul – 1 pont

A-ra indul – 1 pont

A belső részén minden állapotváltás, ahol valóságos átmenet van (4 db) 0.5 pont darabonként

8. Sóbri Jóska a főispán zabját szeretné tréfából kihegyezni. Ezért bemegy a legközelebbi csárdába, és kér egy zabhegyezőt a csapostól. Neki ilyen éppen nincs, ezért a csapos felhívja a Kukutyin Kft-t, és rendel alapanyagot. Jósának azt feleli, hogy majd értesíti, ha elkészült a zabhegyező. Mikor a csaposhoz megérkezik az alapanyag, a csapos készít egy zabhegyezőt, majd értesíti Jóskát. Jóska erre bemegy, és elveszi a hegyezőt, fizetés nélkül távozik.

Rajzoljon **UML szekvencia diagramot** a fenti eseménysorra! (8 pont)



minden objektum 0.5 pont

minden üzenet 1 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2002. január 22.

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (7 pont)

A MONEY iroda kisebb cégek pénzügyeit bonyolítja egy olyan világban, ahol nincs készpénz és minden fizetés bankszámlák közötti átutalással történik. A céget jellemző adat a név, a cím, az adószám és a banki folyószámla (egy cégnek csak egy lehet) száma. Az iroda állítja ki a cég – mint szállító – számláit és küldi meg a vevő cégnek. A kiállított számlán szerepel a szállító és a vevő cége, a számla sorszáma, dátuma, az összeg és az ÁFA. A számla szállítója mindig az iroda ügyfele (hiszen csak ügyfele nevében állíthat ki számlát az iroda), a vevő cég általában nem ügyfél, de lehet az is. Az iroda a vevő cégek adatait is nyilvántartja, még akkor is, ha azok nem az iroda ügyfelei. A számla egy sorában a szállított termék, annak mennyisége, nettó ára és ÁFA-ja áll. Egy számla több sorból állhat. A számlák kiállítását egyszerűsíti a termékjegyzék, amelyben megtalálható minden termék kódja, megnevezése és ÁFA kulcsa. Természetesen az iroda fogadja az ügyfeleinek érkező számlákat és intézkedik a kifizetésről. Az iroda megkapja ügyfeleinek bankjától az egyenlegértékesítést. Ezen szerepelnek: az értesítő sorszáma, a dátuma, az egyenleg, valamint a legutóbbi értesítést követően lezajlott tranzakciók (legalább egy). Egy tranzakció egy pénzmozgást rögzít. Megadja a mozgás összegét, irányát (-kiadás , + bevétel) és hivatkozik a pénzmozgást kiváltó számlára.

CÉG (név, cím, adószám, folyószámla, ügyfél-e)

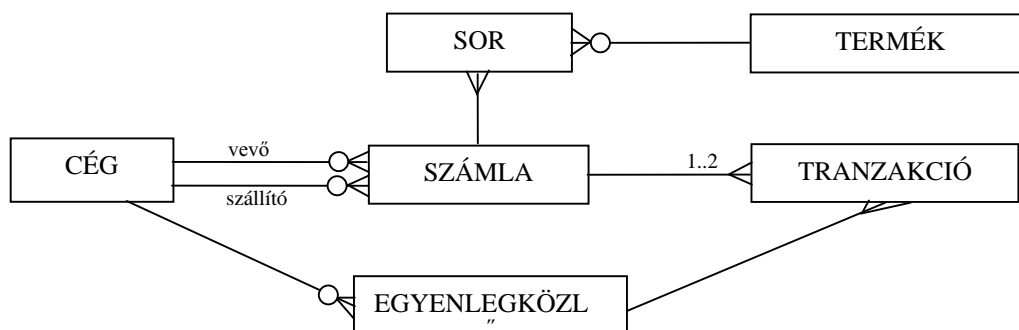
SZÁMLA (cég(szállító), számlaszám, cég(vevő), összeg, ÁFA, dátum)

SOR (számla, sorszám, termék, mennyiség, nettóár, ÁFA)

TERMÉK (termékkód, terméknév, ÁFA-kulcs)

EGYENLEGKÖZLŐ (cég, sorszám, dátum, egyenleg)

TRANZAKCIÓ (egyenlegközlő, sorszám, összeg, számla)



mindegyik entitás és mindegyik reláció 0.5 pontot ér. 3 pont fölött +0.5 pont.

2. Adja meg a szoftver munka kiszámításának képletét, a képletben szereplő értékek jelentését, valamint a képletből levonható legfontosabb következtetést ! (4 pont)

$$E = \frac{V}{l} = \frac{V^2}{V^*} \quad 2 \text{ pont}$$

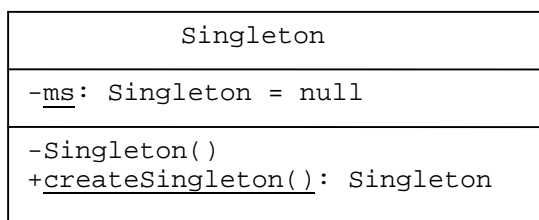
V = tényleges programtérfohat
 V* = elméleti programtérfohat 2 pont
 l = implementációs szint
 a programot részekre kell bontani !

3. Adja meg a felsorolt komponens rendszerekben alkalmazott jellemző adaptációs módszereket. (3 pont)

rendszer	adaptáció
Modul	tipikusan kódolás, de nyelvfüggő.....
Keretrendszer	öröklés, delegálás.....
CORBA és társai	csomagoló osztályok.....
Architektúra rendszerek	konnektorok.....
Aspect oriented programming	szövézés.....

Mindegyik elem 0.5 pont

4. Legyen egy olyan Singleton osztály, amelyből kizárólag egyetlen példány hozható létre. Rajzoljon **UML osztálydiagramot**, a metódusok és attribútumok, valamint ezek összes tulajdonságainak (láthatóság, kezdeti és visszatérési érték, paraméter, stb.) feltüntetésével ! (6 pont)



```
public class Singleton {
    private static Singleton ms = null;
    private Singleton() {super();}
    public static Singleton createSingle() {
        if (ms == null) {ms = new Singleton(); }
        return my; }
}
```

/* mutató az egyetlen példányra
 /* konstruktort más nem használhatja
 /* ezzel kérünk mutatót az egyetlenre
 /* ha még nincs, akkor konstruktor hívás
 /* visszaadjuk a példányt

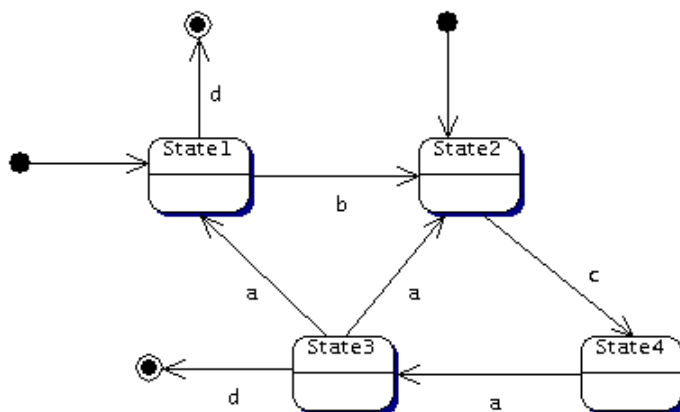
itemenként 2 pont

5. Milyen hibá(ka)t talál az alábbi **UML statechart**-on ? (3 pont)

Két kezdő állapot.....

State3-ban a hatása nem egyértelmű.....

.....



6 Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

Törppapa szokásos évi sítúrájára ment, és Okoskára bízta Aprajafalva felügyeletét. Okoska úgy döntött, ideje megmozgatni az ellustult törpöket, ezért téli törpbogyó kutatást szervez. Minden törpnek, aki részt akar venni a kutatásban Okoskánál kell jelentkeznie. A jelentkezést követően Okoskától lehet bogyót kérni (a bogyót Okoska egyedi azonosítóval látja el, hogy ne lehessen csalni), amelyet ezután a faluban kell elrejtteni, és a rejtkehelyről Okoskát értesíteni kell. Újabb bogyó csak akkor kérhető, ha az előző rejtkehelyét Okoskának jelentették. Mielőtt a verseny elkezdődne, a még el nem rejtett bogyót Okoskának vissza kell szolgáltatni. A kutatás indulásakor a nevezők Hókuszpók egy-egy zokniját kapják, amelybe gyűjthetik a bogyókat. A kutatásra 1 törpórányi idő áll rendelkezésre. Az idő lejártakor a zoknikat Okoskának kell leadni, aki a bogyók száma alapján eredményt hirdet. Aki saját bogyót is begyűjtött azt Okoska kizárja, a többiek pedig Okoska összes mondását tartalmazó papirusztekerccset kapnak ajándékba. A végén a törpök a begyűjtött bogyókat megeszik. A meg nem talált bogyókat Okoska szedi össze, és eszi meg.

A játék lebonyolítását csak Hami zavarja, aki az elrejtett bogyókat versenyen kívül kutatja, és amelyiket megtalálja, azt rögvest meg is eszi.

Definiálja az alábbi funkciókat ellátó processzeket:

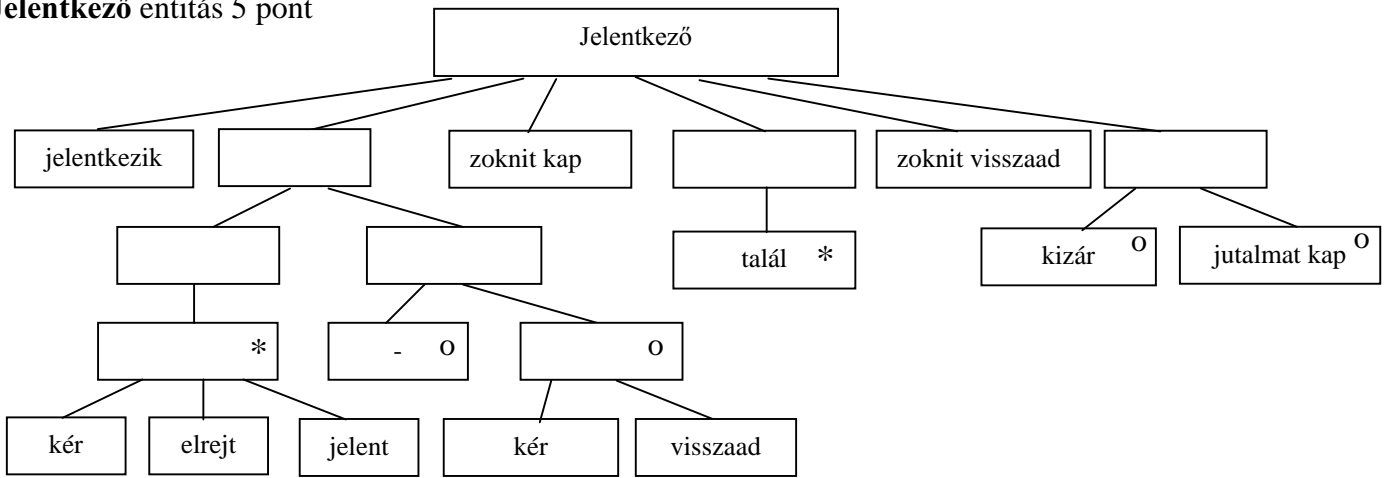
Jelezzzen, ha olyan törp kap zoknit, aki nem rejtett el bogyót !

Kérésre listázza ki azokat a Hami által elfogyasztott bogyókat, amelyeket kizárt törp rejtett el !

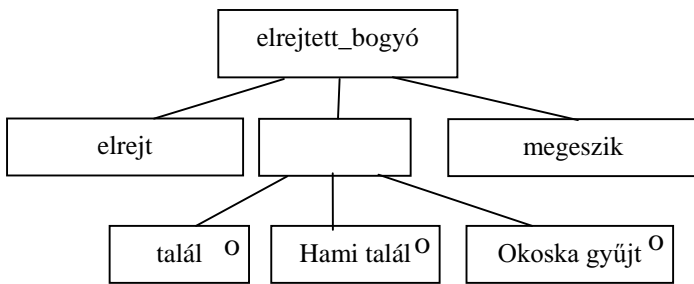
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

- jelentkezik (törp) eseménylista 2 pont
- kér (törp, bogyó)
- elrejt (törp, bogyó)
- jelent (törp, bogyó, hely)
- visszaad (törp, bogyó)
- zoknit kap (törp)
- talál (törp, bogyó)
- zoknit visszaad (törp)
- kizár (törp)
- jutalmat kap (törp)
- Hami talál (bogyó)
- Okoska gyűjt (bogyó)
- megeszik (bogyó)

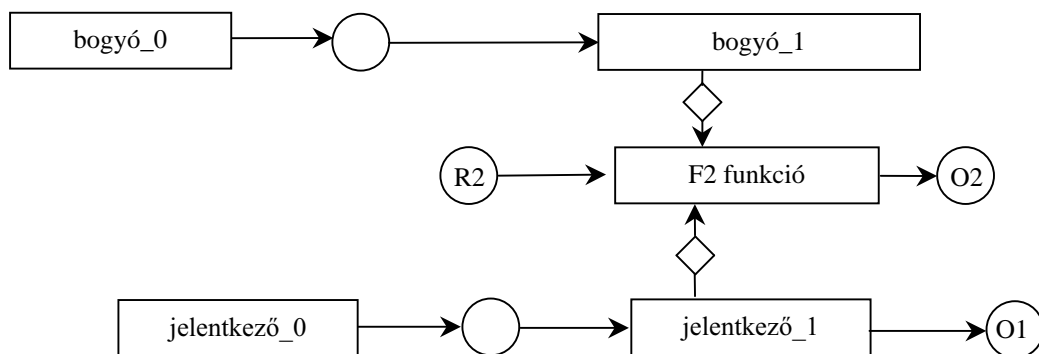
Jelentkező entitás 5 pont



elrejtett bogyó entitás 2 pont



teljes SSD 2 pont



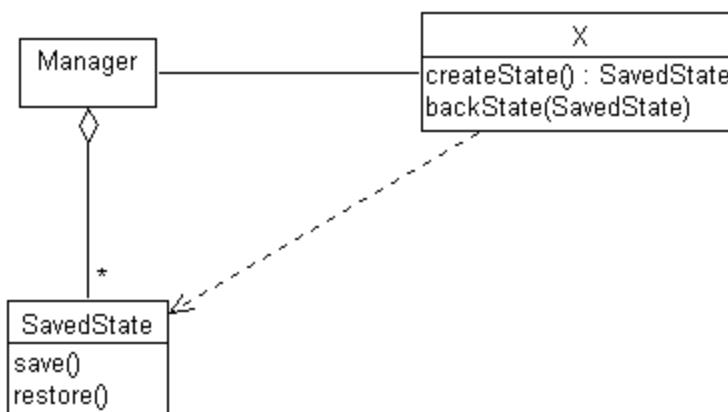
7. Legyen egy X objektumunk, amelynek állapotát időnként menteni kell egy (SavedState objektumba, annak a save() metódusát meghívva), hogy az később visszaállítható legyen (a SavedState restore() metódusával). A SavedState objektumokat egy Manager objektum kezeli, amely utasítja X-et egy mentésre (createState()), vagy az átadott SavedState szerinti állapot visszaállítására (backState(SavedState)).

Rajzoljon UML osztálydiagramot, az operációk feltüntetésével ! (3 pont)

Hogyan tudja megakadályozni, hogy a Manager az elmentett állapotot megváltoztassa ? (Minden jó megoldás 3 pont)

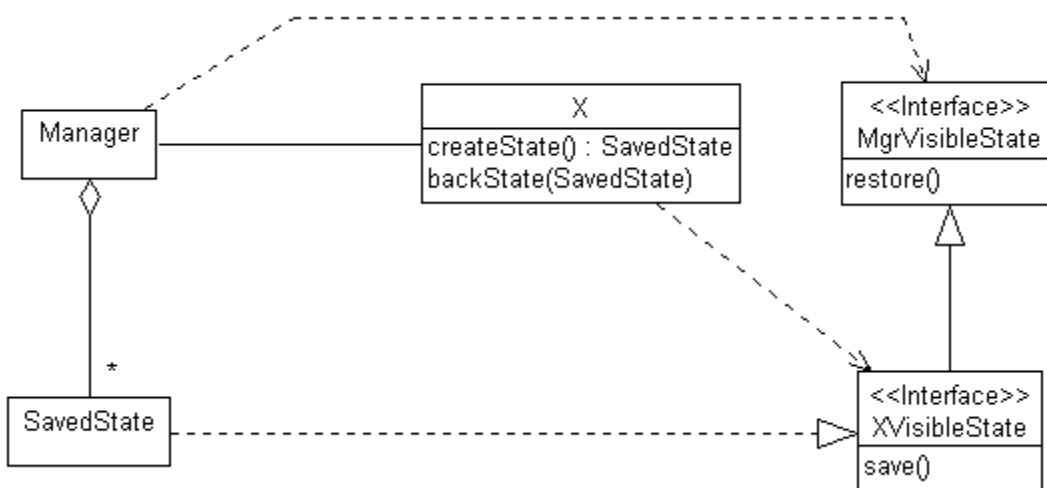
Classdiagram (3 pont)

minden class és assoc 0.5 pont



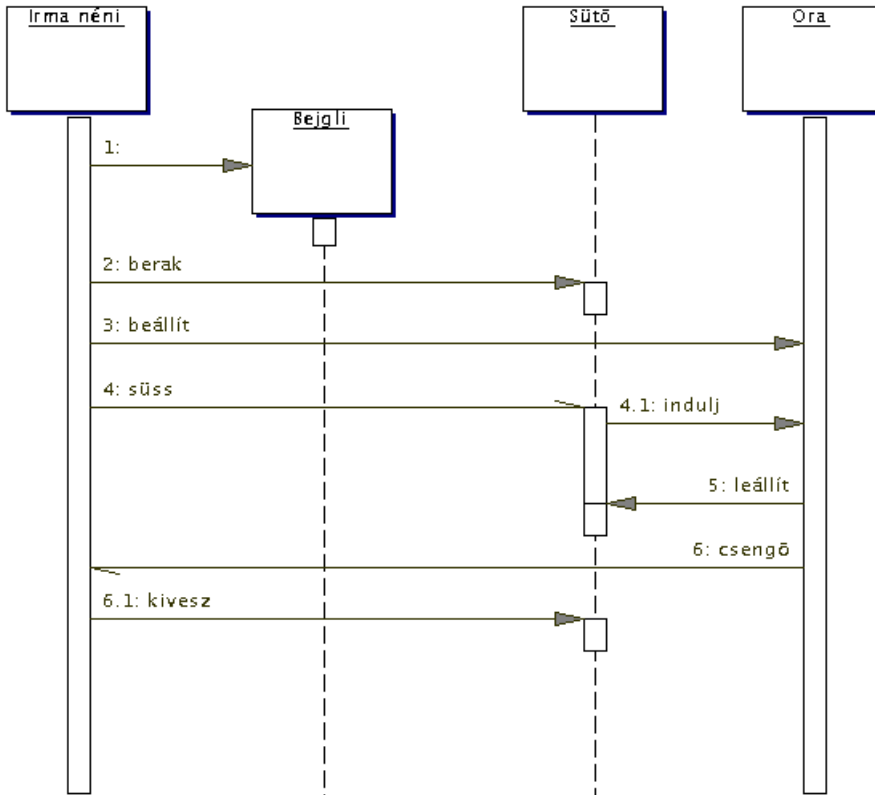
A) SavedState-ben van egy boolean változó, amelyet a konstruktor false-ra állít. A save() belépéskor megnézi, hogy ez a változó false-e. Ha igen, engedi a továbbiak végrehajtását és a változót true-ba viszi. Gond lehet: senki sem tud változtatni (3 pont)

B) Legyen MgrVisibleState interfész, amelyből származtatjuk az XVisibleState interfészt úgy, hogy ez utóbbi interfésznek van save() operációja, amit Manager nem lát. (3 pont)



8. Rajzoljon az eseményekről **UML szekvencia-diagramot** ! (7 pont)

Irma néni bejglit süt. Elkészíti a tésztát, majd azt berakja a sütőbe. A sütő – egyébként állandóan járó – óráját beállítja, hogy az pontosan a sütés megkezdése után 93 perccel a sütőt állítsa le. Ezek után elindítja a sütőt, amely ekkor elindítja az órát. 93 perc múltán az óra leállítja a sütőt, és csenget. Irma néni ekkor kiveszi a bejglit.



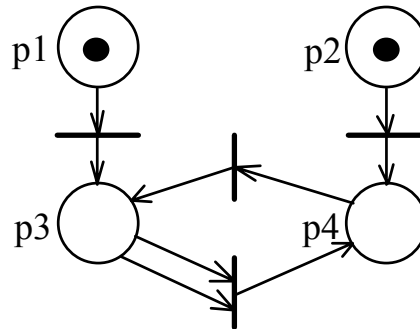
minden objektum és üzenet 0.5 pont, két aszinkron +1 pont

Eredmények értékelése:

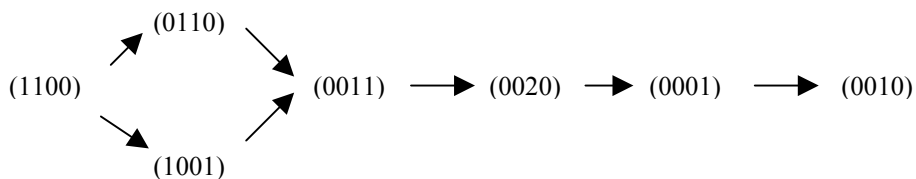
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
 PROGRAMOZÁS TECHNOLÓGIÁJA
 c. tárgyból
 2002. június 15.

1. Az ábrán szereplő markerezésből kiindulva adja meg a Petri-háló elérhetőségi gráfját! (3 pont)



Megoldás: minden él a korrekt értékkel 0,5 pont



2. Mit nevezünk „temporális kohézió”-nak és mikor használjuk ? (4 pont)

A kohézió temporális, ha a vizsgált egységben (modulban, objektumban) szereplő elemek közötti kapcsolat lényege nem a funkcionalitás, hanem az egyidejűség. Olyankor használjuk, amikor a különböző tevékenységek időbelisége nagyon kötött, így inicializálásnál, kivételes helyzetek kezelésénél, a programok (taskok) futásának lezárásánál, felfüggesztésénél.

3. Milyen mennyiségek között definiál kapcsolatokat a Constructive Cost Model ? (3 pont)

MM – erőfeszítés emberhónap

KS – kilosource

T – fejlesztési idő

4. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (5 pont)

Egy vasúttársaság több mozdonytípust is használ. A típusokra jellemző a fajta (gőz, diesel, villamos), a tengelynyomás és a vonóerő. A típus kódja egy háromjegyű szám vagy egy betű és egy kétjegyű szám. Döntéshozzáértési és összehasonlítási célból olyan típusok adatait is tárolják, amelyen mozdonyral nem rendelkeznek. A mozdonyokat a típus és egy három vagy négyjegyű sorszám azonosítja, jellemzője még a gyártás éve. A mozdonyok vonatokat továbbítanak, minden vonat pontosan egy mozdony. A vonat jellemzője az össztömege, a dátum, a késés és a vonatszám. Ugyanazzal a vonatszámmal nem indulhat két vonat ugyanazon a napon. A vonatszámtól függ a vonat fajtája (személy, teher), az induló- és célállomás. Az állomások jellemzője a név, a vágányok száma és a szolgálat létszáma.

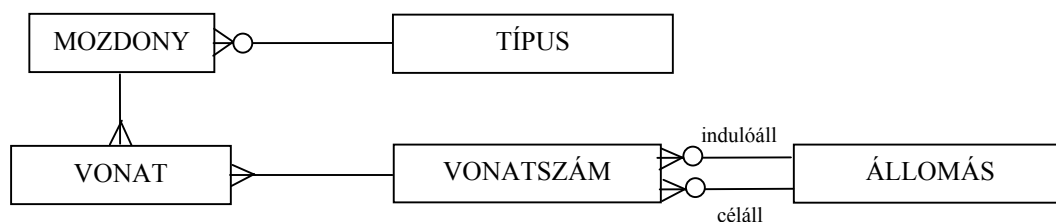
TÍPUS (kód, fajta, tengelynyomás, vonóerő,)

MOZDONY (típus, sorszám, gyártási év)

VONAT (vonatszám, össztömeg, dátum, késés)

VONATSZÁM (szám, fajta, indulóállomás, célállomás)

ÁLLOMÁS (név, vágányok száma, létszám)



Minden entitás és reláció 0,5 pont

5. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A Super Business Scholl (SBS) minden félévben kurzusokat hirdet meg hallgatói számára. Egy hallgató több kurzusra is beiratkozhat. Csak az a kurzus indul, amelyre legalább tizen beiratkoztak. A hallgatóknak a félév közben alkalmanként kiadott házi feladatot kell határidőre elkészíteniük, és a megoldást e-mail-ben a kurzusvezetőnek elküldeniük. A feladatra kapott pontszámáról - ugyancsak e-mailben – a beküldést követően csaknem azonnal értesítést kapnak. A félév végén vizsgát kell tenni. Csak az vizsgázhat, aki a feladatok legalább 80 %-át beküldte és legalább 10 pontos átlagot ért el. Minden kurzuson legalább 3 vizsgát tartanak, de egy tárgyból egy hallgató legfeljebb kétszer vizsgázhat. A második vizsgára csak akkor kerülhet sor, ha az első sikertelen volt.

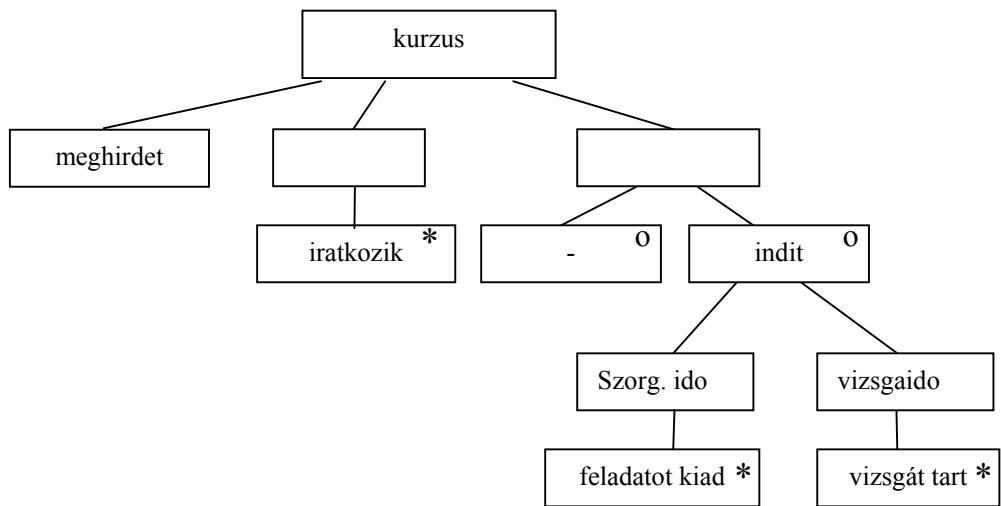
Definiálja az alábbi funkciókat ellátó processzeket:

- Kérésre adja meg azok listáját, akik valamely kurzuson nem vizsgázhatnak
- Kérésre adja meg azokat a kurzusokat, ahol már legalább egy vizsgát tartottak
- Azonnal jelezen, ha valaki másodszor is megbukott

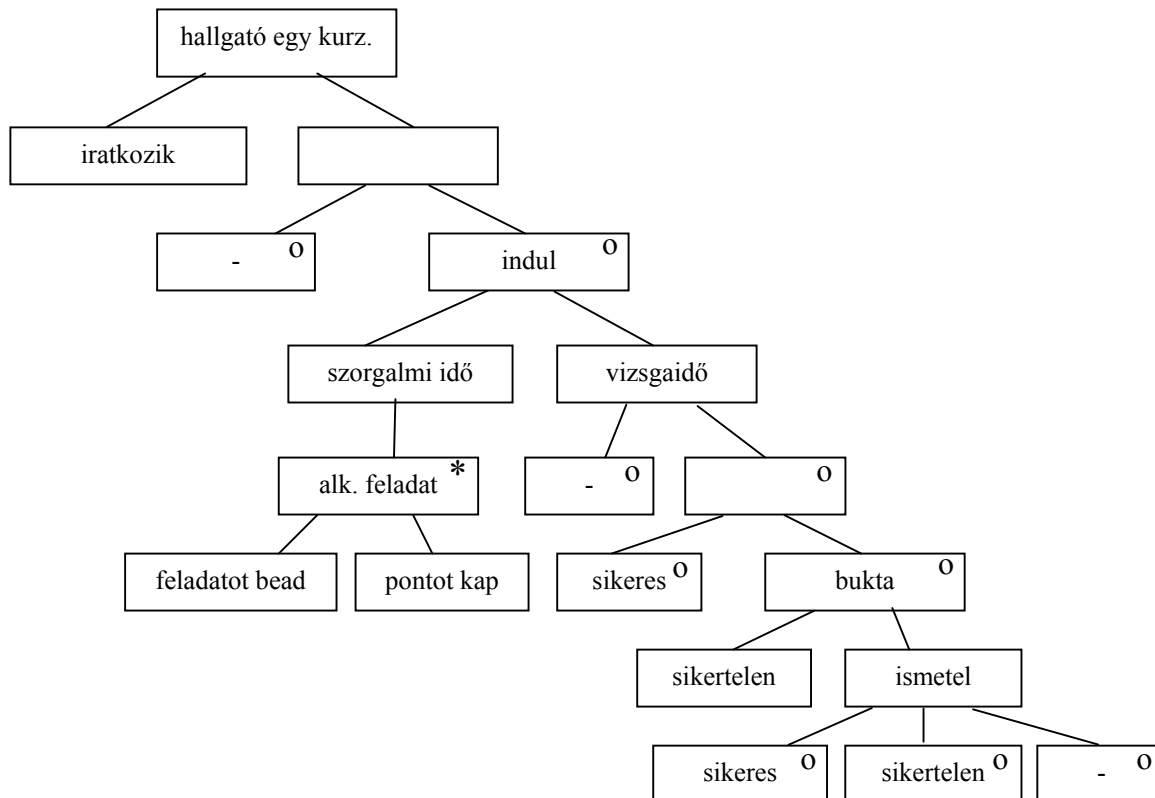
Készítse el az **entitás/akció leírást, az entítások attribútumaival, az entitás struktúra diagramot, az initial modelt, funkció processzeket.**
kapcsolja a modellhez a

- meghirdet (kurzus) eseménylista 2 pont
- iratkozik (hallgató, kurzus)
- feladatot kiad (kurzus)
- feladatot bead (hallgató)
- pontot kap (hallgató)
- vizsgát tart (kurzus)
- sikeres vizsga (hallgató)
- sikertelen vizsga (hallgató)

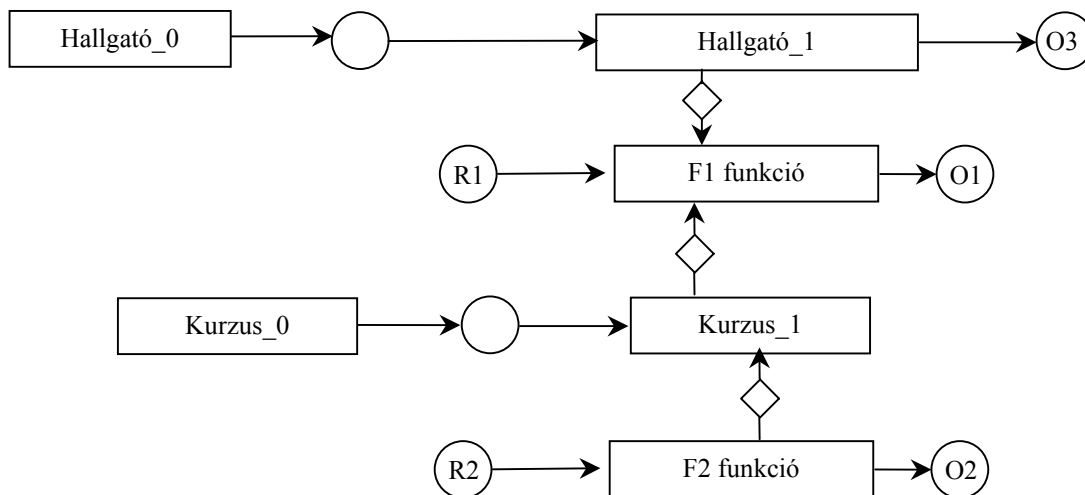
Kurzus entitás 3 pont



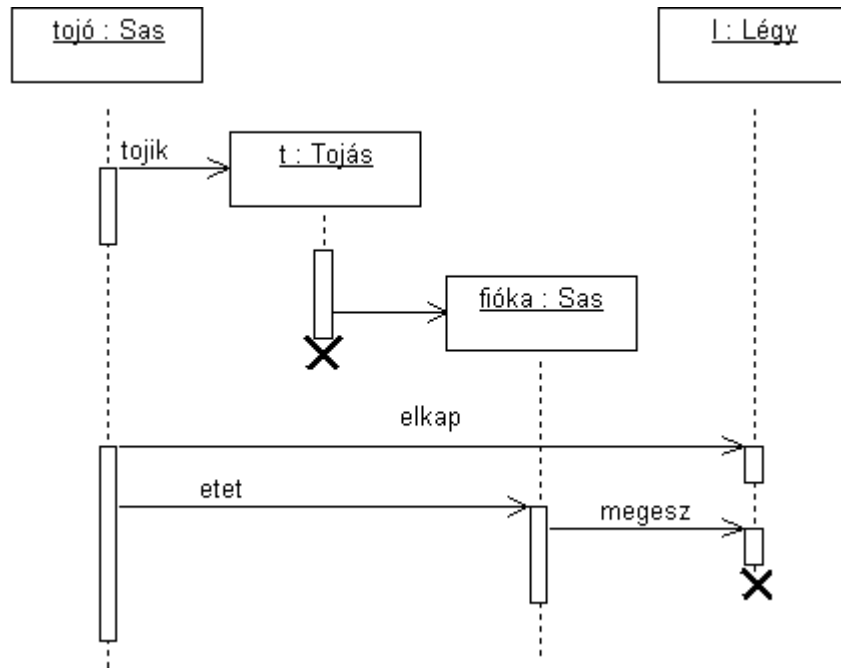
Hallgató egy kurzuson entitás 4 pont



teljes SSD 2 pont

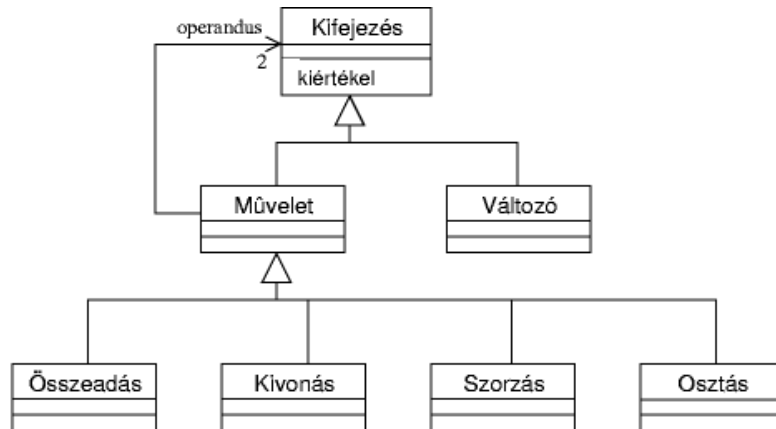


6. A légyfogó sas (*Aquila muscicaptans*) egy tojást tojik a fészkébe. A tojásból idővel kikelő sasfiókat az anyja egy elkapott léggel megeteti. Rajzoljon **UML szekvencia diagramot** ! (9 pont)

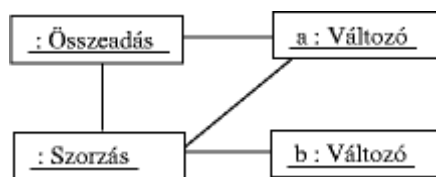


minden objektum 0,5 pont, minden üzenet és pusztulás 1 pont

7. Az ábrán egy – kifejezéseket leíró – osztálydiagram látható. Egy kifejezés művelet vagy változó lehet. Egy műveletnek két kifejezés típusú operandusa van.

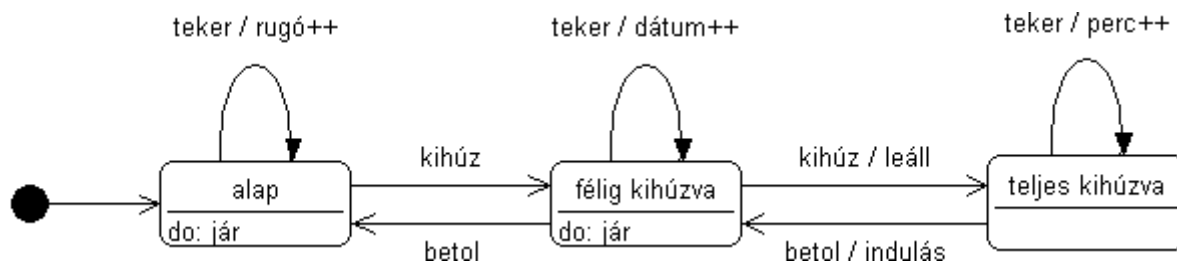


Rajzoljon **UML objektum diagramot** az $a+b*a$ kifejezés leírására. (8 pont)



Minden objektum és minden link 1 pont

8. A Schmalling óragyár gyártja a világhírű NeverStops karórát. Ennek jobb oldalán található az ún. korona. Ezt kihúzni, betolni, illetve tekerni lehet. Kezdetben a korona teljesen be van tolvá. Ha ilyenkor megtekerjük a koronát, akkor az órát felhúzzuk. Ha a koronát egy kicsit kihúzzuk, akkor a tekerésre a dátumot jelző lapka egyet-egyét ugrik. Ha a koronát még jobban kihúzzuk (ennél jobban nem is lehet), a mutatókat tudjuk a tekeréssel állítani. Az óra folyamatosan jár, kivéve azt az esetet, amikor a koronát teljesen kihúztuk; ilyenkor addig áll, míg a koronát beljebb nem toljuk. Rajzoljon **UML state-chart**-ot ! (7 pont)



minden állapot és a kezdet 1 pont, minden átmenet 0.5 pont, max 7 pont

Eredmények értékelése:

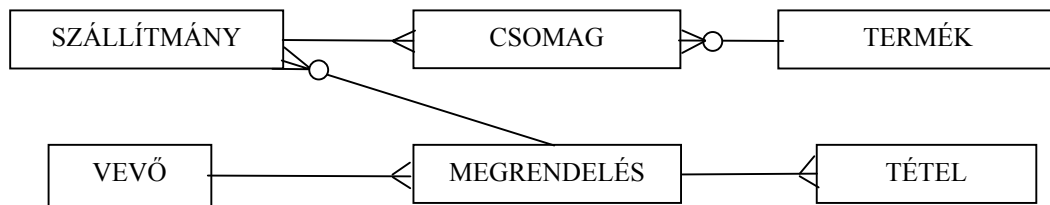
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2002. május 23.

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A nyúlzavarógépgyár az általa gyártott termékekről nyilvántartja azok megnevezését, a raktáron levő mennyiséget, az egységárat és az AFA kulcsot. A gyárhoz folyamatosan érkeznek a megrendelések (szám, vevő, a szállítás várható ideje, helye). Egy megrendelés legkevesebb egy tételből áll. A tétel megmondja, hogy egy termékből hány darabot akar venni a vásárló. A megrendelés vonatkozhat olyasmire is, amit nem gyártanak. Ilyet persze szállítani sem tud a gyár. Az egy megrendelésen szereplő termékeket a gyár egy vagy több – a mennyiségtől függően – szállítmányban (száma, szállítás dátuma, szállítást bonyolító személy/cég) küldi el a megrendelőnek. Egy szállítmány mindig egy megrendeléshez tartozik. A szállítmány csomagokból áll. Egy csomagban (csomag száma, méret, súly) azonos termékek egy vagy több darabja van. Nyilvántartják még a vevők alapadatait (név, cím, adószám, kapcsolattartó személy).



mindegyik entitás és mindegyik reláció 0.5 pontot ér. 3 pont fölött +0.5 pont.

2. Jellemezzünk egy stringet az alábbi műveletekkel ! A string karakterei előről 1-gyel kezdődően számozottak. Adja meg a PAR műveletre vonatkozó algebrai axiómákat ! Az axiómák felírásakor használhatja a két egész összehasonlítására és két karakter összehasonlítására szolgáló műveleteket. (3 pont)

- CRT()** új (üres) stringet hoz létre.
- SET(s,x)** az **s** string elejére – az 1. számú helyre – rakja az **x** karaktert.
- PAR(s)** igaz, ha bárhol a stringben egymás mellett legalább két egyforma karakter áll.
- IN(s,i)** eredményül adja az **s** string **i**-ik karakterét. Ha **i** nagyobb mint a string hossza, akkor az eredmény értelmetlen (nem definiált).
- LGTH(s)** az **s** string karaktereinek számát adja.

PAR(CRT()) = false	-0.5 pont, ha hiányzik
PAR(SET(s, x)) == (LGTH(s) != 0) &&	1 pont
(x == IN(s, 1)	1 pont
PAR(s))	1 pont

ez utóbbival egyenértékű:

if (LGTH(s) == 0) false else if (x == IN(s,1) || PAR(s)) true

3. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

- gyakorlati problémák
 - megoldására szolgáló dolgok
 - a társadalom által korlátozott,
 - tudományos alapokon álló,
 - gazdaságos
- előállításának mikéntje

mindegyik komponens 1 pont.

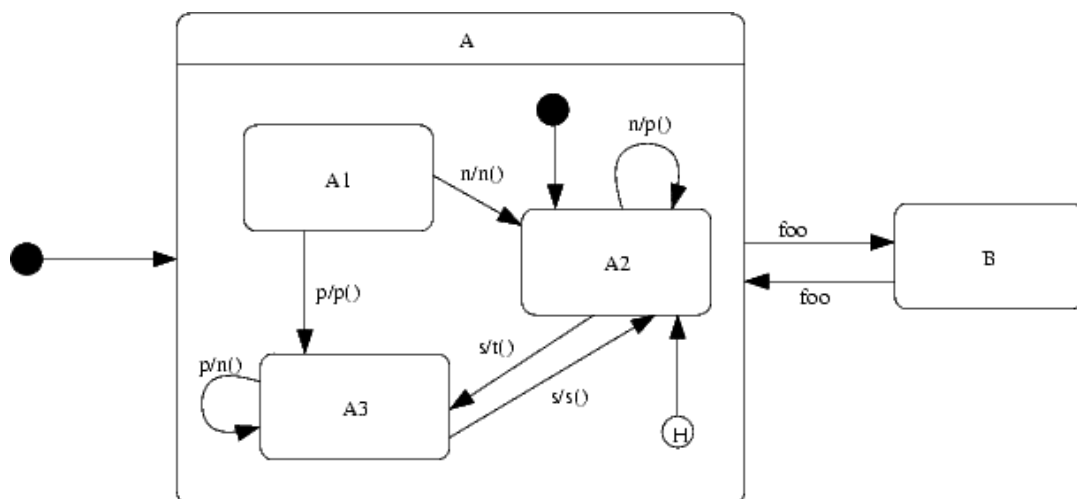
4. Adva van egy objektumunk, amely két fő állapottal (A, B) rendelkezik. Az A állapotban a következő állapotgép működik:

	p	n	s
A1	A3/p()	A2/n()	-
A2	-	A2/p()	A3/t()
A3	A3/n()	-	A2/s()

Az A állapotból B-be a foo esemény hatására kerül. Visszatérni ismét a foo-ra fog, és ekkor ott folytatja, ahol a kilépéskor abbahagyta. A predefinit kezdő állapot az A, azon belül A2.

Rajzolja meg a objektum **UML state-chart**-ját (7 pont)!

- A és B állapot 1 pont
- A belső állapotai 1 pont
- Induló állapotok 1 pont
- Össz átmenet (8) 2 pont
- History indikátor 1 pont
- HI inicializálása 1 pont



5 Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A Rendszerinformisztikai Tanszéken minden évben január végén kezdődik a diplomatervezés azzal, hogy a diplomázók beadják a - minden lényeges információt tartalmazó – adatlapot az adminisztrációban. Ezt követően a hallgató dolgozik, jár konzultálni, amiről az oktató rendszeresen jelentést készít az adminisztrációnak. A diplomaterv beadását (május közepe) követően a diplomatervet kiküldik bíráltni. Legalább két bírálat készül, de inkább három. Sajnos a bírálókkal előfordul, hogy a megadott határidőre nem készítik el a bírálatot, ezért ilyenkor más bírálókat kérnek fel. Az államvizsgán a hallgatónak be kell mutatnia munkáját és a bírálatokban szereplő megjegyzésekre reagálni kell. A bírálók a bírálat elkészítéséért utólag díjazásban részesülnek.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha olyan hallgató akarja beadni a diplomatervét, aki nem járt rendszeresen konzultálni !

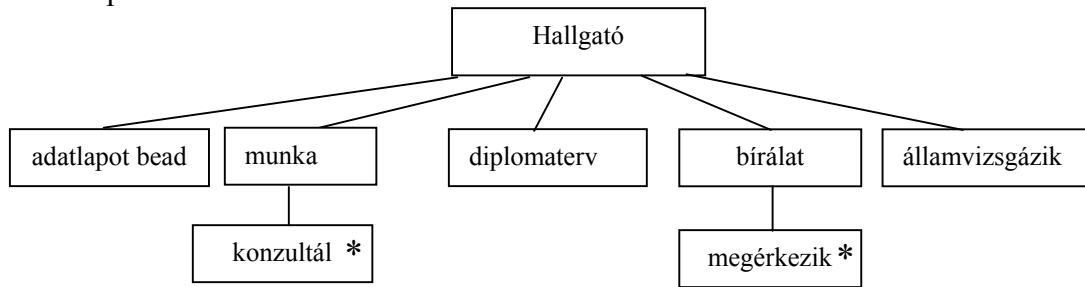
Kérésre listázza ki azokat a hallgatókat, akiknek már legalább két bírálatuk bérkezett !

Kérésre adja meg azokat a bírálókat, akik elkészítették a bírálatot, de még nem kapták meg a díjat !

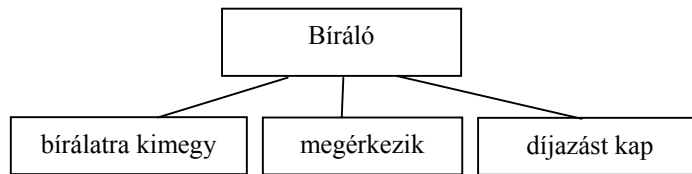
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
 kapcsolja a modellhez a **funkció processzeket.**

- adatlapot bead (hallgató) eseménylista 3 pont
- konzultál (hallgató)
- diplomatervet bead (hallgató)
- bírálatra megy (bíráló)
- bírálat megérkezik (bíráló, hallgató)
- államvizgázik (hallgató)
- díjazást kap (bíráló)

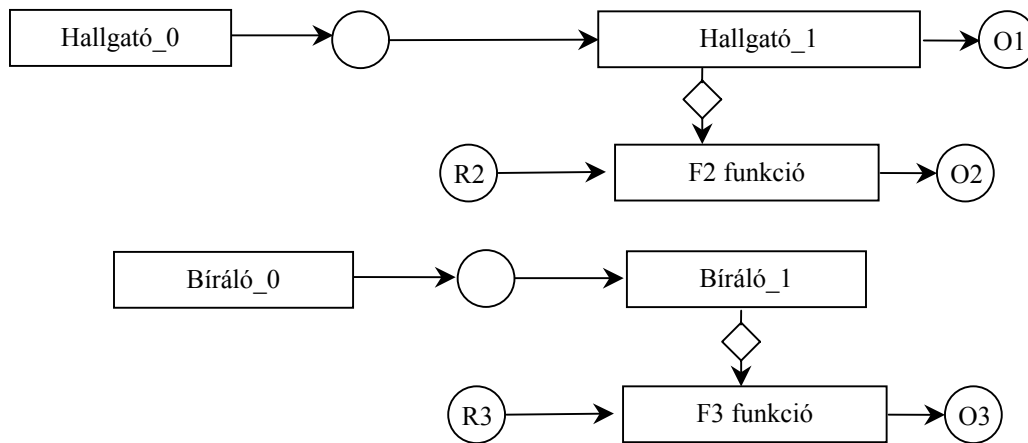
Hallgató entitás 4 pont



Bíráló entitás 2 pont



teljes SSD 2 pont



6. Java közbülső kódban áll rendelkezésre az alábbi két osztály:

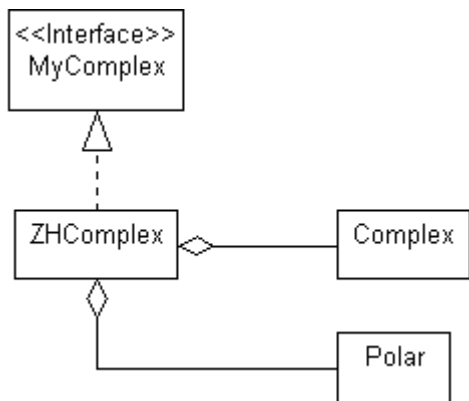
Complex
Complex(re : double, im : double)
add(c : Complex) : Complex
sub(c : Complex) : Complex
getRe() : double
getIm() : double

Polar
Polar(r : double, fi : double)
getR() : double
getFi() : double
mult(p : Polar) : Polar
div(p : Polar) : Polar

Szükségünk lenne egy olyan osztályra (ZHComplex), amelyik megvalósítja az alábbi interfészt:

<<Interface>> MyComplex
MyComplex(re : double) : double
add(c : MyComplex) : MyComplex
sub(c : MyComplex) : MyComplex
getRe() : double
getIm() : double
mult(c : MyComplex) : MyComplex
div(c : MyComplex) : MyComplex

A megoldáshoz felhasználjuk mindkét megadott osztályt. Rajzolja fel a megoldás **UML osztálydiagramját** ! (6 pont)

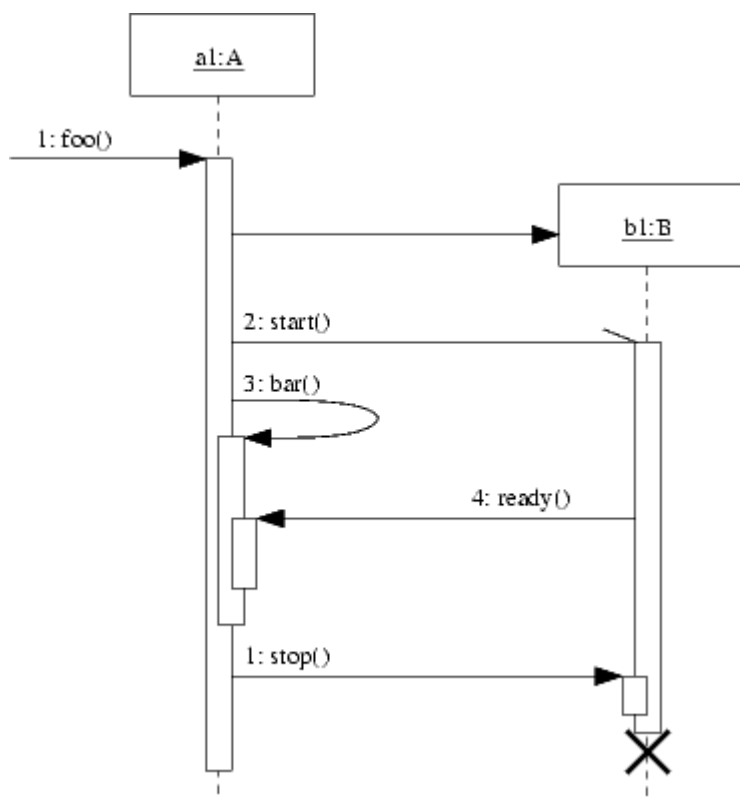


minden korrekt kapcsolat 2 pont

7. Rajzoljon **UML szekvencia-diagramot** az alábbi esemény sor leírására ! (9 pont)

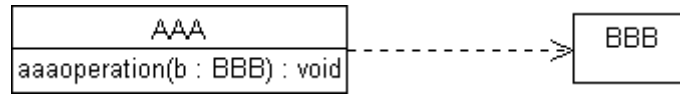
Az A osztály egy a1 példánya a foo() metódus hatására létrehoz egy B osztályba tartozó b1 objektumot, majd aszinkron módon meghívja annak start() metódusát. Ezután a1 meghívja saját bar() metódusát.

A b1 objektum start()-tal elindított metódusa – bar() futása alatt – szinkron ready() üzenetet küld a1-nek. A bar() befejezését követően a1 meghívja b1 stop() metódusát, mire b1 start()-tal indított metódusa befejeződik és b1 megsemmisül.



egy Objektum 1-1 pont, minden üzenet a korrekt fogadó oldallal 1 pont és elpusztul 1 pont

8. Az AAA osztály aaaoperation() művelete paraméterként kap egy BBB osztályú objektumot. Jelölje be az AAA és BBB közötti kapcsolatot az UML jelölésrendszerével az alábbi ábrarészleten ! (3 pont)



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2002. május 30.

1. A Petri-hálót egy 4 elemű algebrai struktúra (P, T, A, M) írja le. Definiálja (algebrailag és szövegesen) az egyes elemeket ! (4pont)

$P = \{P_1, \dots, P_n\}$ – place-ek halmaza 1 pont

$T = \{T_1, \dots, T_m\}$ – transition-ok halmaza 1 pont

$A \subseteq \{(P \times T) \cup (T \times P)\}$ – élek halmaza 1 pont

$M = P \rightarrow N$ – markerezés 1 pont

2. Milyen típusú kohéziójú a
Pacal nyelv “print (... .)” eljárása ?
(1 pont)

- általunk definiált
“getValidRecord()” C function ?
(1 pont)

procedurális

szekvenciális

3. A három rétegű szoftver architektúrában milyen logikai rétegeket definiáltunk ? (3 pont)

..... GUI – felhasználói felület

..... BOM – alkalmazás modell

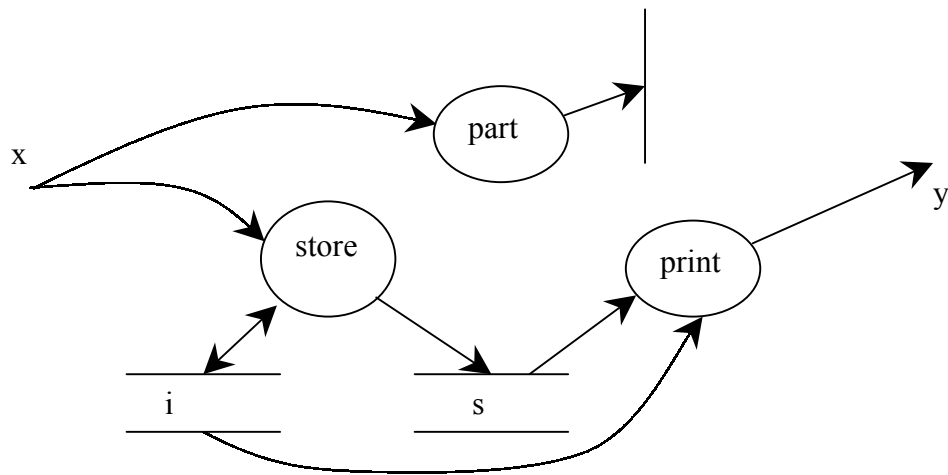
..... DB – adatbázis

4. Egy program (amelynek bemenete x , kimenete y) működését az alábbi állapottábla írja le:

	e1	e2	e3
A1	A1/-	A1/-	A2/s[i++]=x
A2	A3/-	A3/-	A2/s[i++]=x
A3	A3/s[i++]=x	A3/-	A1/y=print(s, i)

Ahol e1, e2, e3 események, amelyek a `part(x)` függvény lehetséges értékei, s egy 100 elemű x típusú értékek tárolására szolgáló tömb, i integer index.

Rajzolja fel a program **adatfolyamábráját** ! (7 pont)



Minden adat, process, store és a control 0,5 pontot ér

5. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

Az Alsóvárosi filmklub új tagokat kíván toborozni a következő módon. A toborzásban valamennyi klubtag részt vehet, a legtöbb új tagot toborzót megjutalmazzák. A toborzásban való részvételt előzetesen be lehet jelenteni, de nem muszály. A tag írásban ajánlja a jelöltet a vezetőségnek. A jelöltet csak akkor veszik fel tagnak, ha a filmklub Jancsó és Pasolini sorozataiból minél több filmet megnéz, majd a végén egy rövid összehasonlító, értékelő dolgozatot ad be a vezetőségnek és a vezetőség ezt a dolgozatot elfogadja. Akinek nem fogadják el a dolgozatát, az fájdalomdíjként kap egy ingyenes bérletet a következő évadra.

Definiálja az alábbi funkciókat ellátó processzeket:

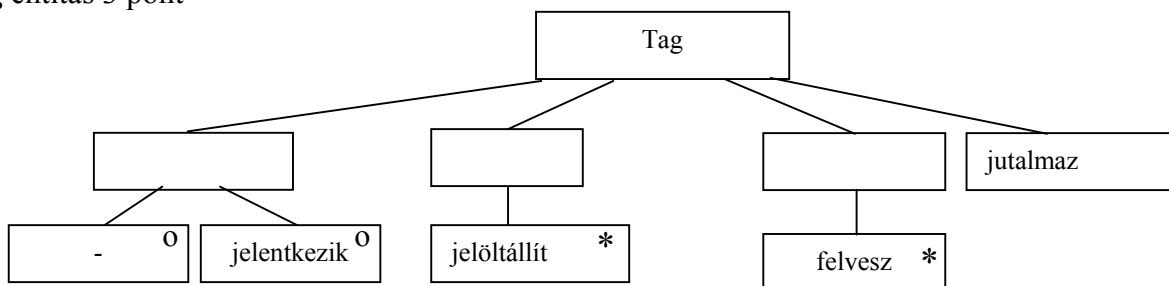
Azonnal jelezzen, ha a tag az ötödik jelöltet ajánlja és előzőleg nem jelentette be a részvételét a toborzásban !
Kérésre listázza ki azokat a jelölteket, akik legalább 8 filmet láttak, de a dolgozatukat a vezetőség nem fogadta el, és a bérletet még nem kapták meg !

Kérésre tagonként adja meg azokat a jelölteket, akik több Jancsó filmet láttak, mint Pasolinit, de még nem adták be az értékelő dolgozatot !

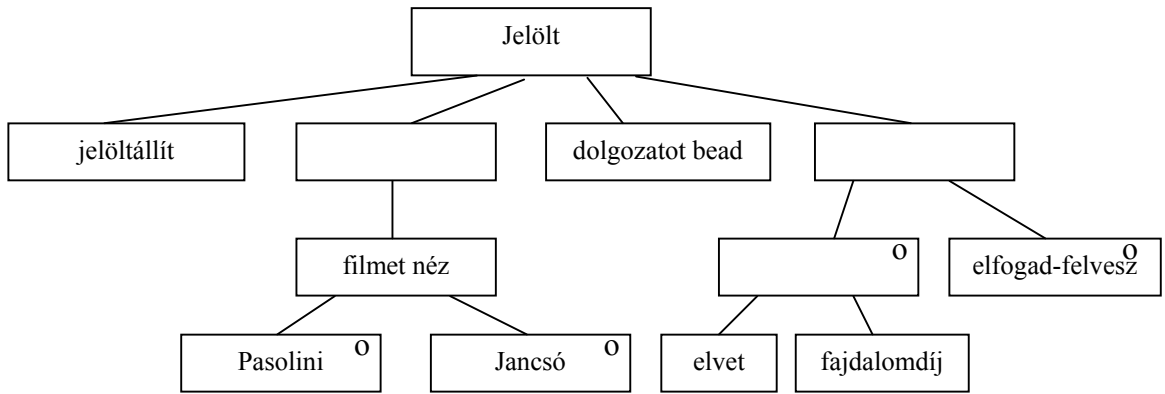
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
kapcsolja a modellhez a **funkció processzeket.**

- jelentkezik (tag) eseménylista 3 pont
- jelöltállít (tag, jelölt)
- filmet néz (jelölt, rendező) vagy Jancsót néz és Pasolinit néz
- dolgozatot bead (jelölt)
- elfogad (jelölt)
- felvesz (jelölt, tag) ez a két esemény lehet egyetlen is.
- elvet (jelölt)
- fájdalomdíjat kap (jelölt)
- jutalmaz (tag)

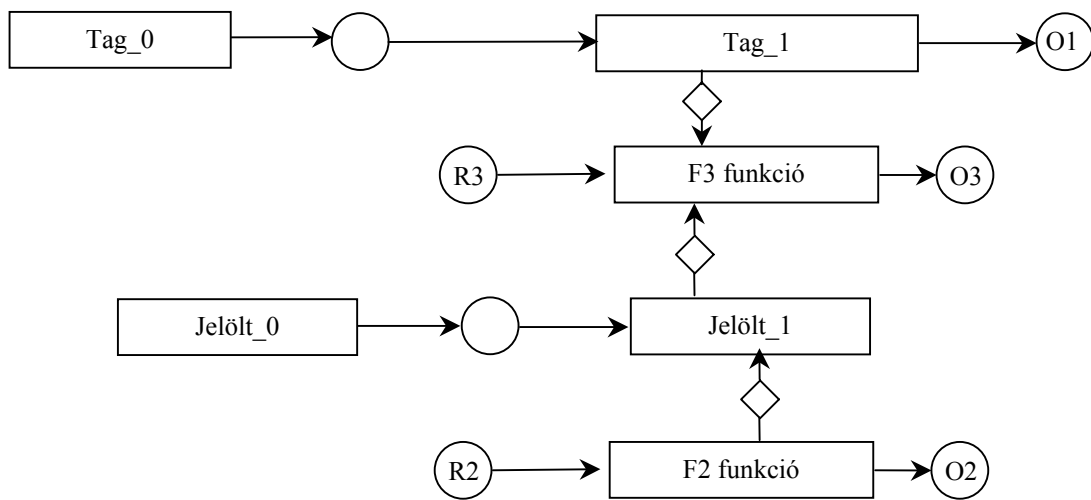
Tag entitás 3 pont



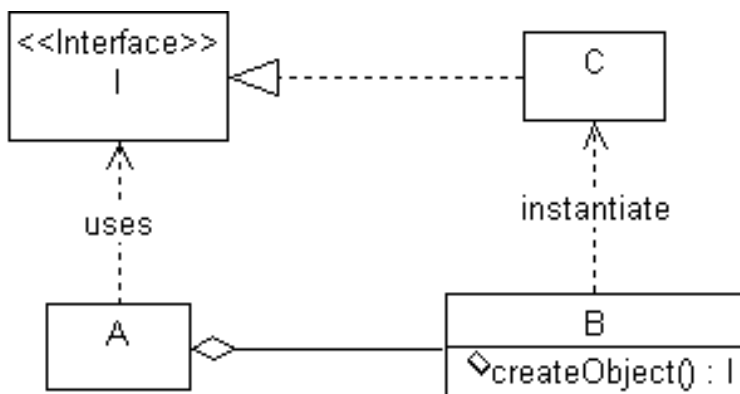
Jelölt entitás 3 pont



teljes SSD 2 pont

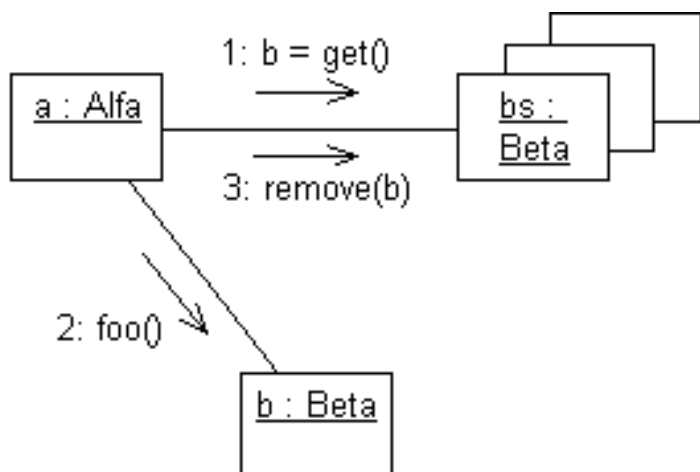


6. Az A osztályú objektum használni akar egy I interfészt megvalósító objektumot, de A nem ismer olyan osztályt, amelyik implementálná I-t. Viszont A-nak van egy B osztályú komponense, amely ismeri az I-t megvalósító C osztályt. A meghívja B createObject() metódusát, amely konstruál egy az A elvárásainak megfelelő objektumot. Rajzoljon **UML struktúra diagramot** ! (8 pont)



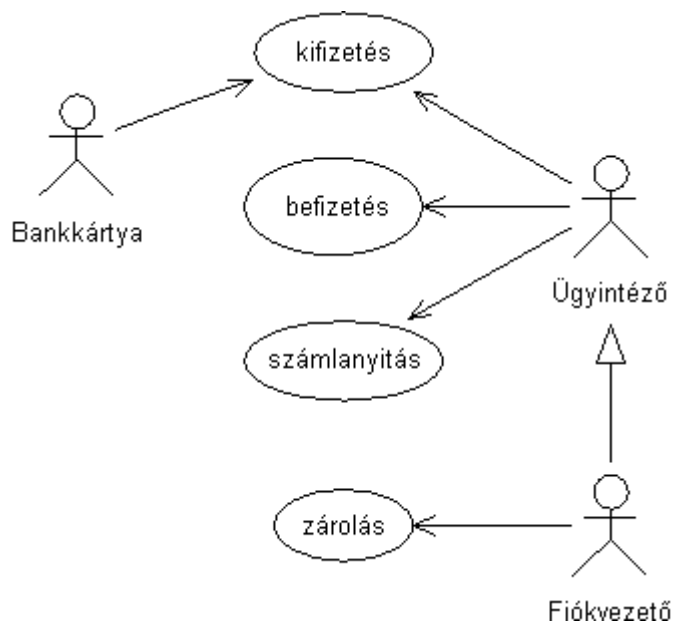
Minden kapcsolat 2 pontot ér

7. Egy Alfa osztálybeli objektumnak van egy Beta osztályú elemekből álló kollekcója. Ebből a `get()` metódussal kiveszi az egyik elemet, amin végrehajtja a `foo()` műveletet. Végül a `remove()` metódussal törli az elemet a gyűjteményből. Rajzoljon **UML kollaborációs diagramot** ! (7 pont)



minden entitás és üzenet (jó sorrendben, jónak küldve) 1 pont kivéve `bs` multiobject, mert az 2.

8. Egy banki rendszerben bankkártyával pénzt tudunk felvenni egy bankszámláról, banki ügyintéző segítségével ezen kívül pénzt lehet befizetni és átutalni. A számlát zárolni csak a fiókvezető tudja, aki természetesen rendelkezik az ügyintézési jogosultságokkal is. Rajzoljon **UML use-case diagramot**. (8 pont)



minden actor és minden use-case 1 pont (jó helyre kapcsolva), az öröklés is 1 pont

Eredmények értékelése:

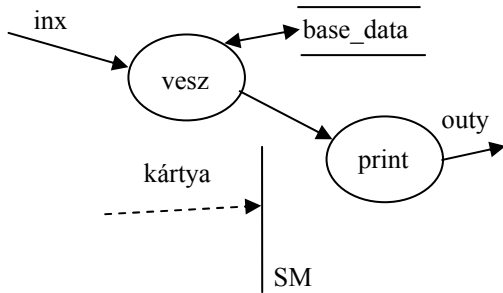
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2003. december 23. 8:00

1. Definiálja az adatfolyam-ábrán szereplő SM vezérlőegység egy lehetséges állapotábráját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) az adatfolyam és adatspecifikációval (5 pont)



a				
b				

kártya = [treff | káró | makk]

az oszlopok paraméterezése (sorrend lényegtelen) csak a kártya konstansai lehetnek 1 pont
 a következő állapotok csak a bejelölt két állapot (a, b) vagy don't care lehet, kell benne lenni állapotváltásnak is. 2 pont
 az akció ha nem üres, csak gombóc (vesz, print) lehet és mindkettőnek legalább egyszer szerepelni kell. 2 pont

2. Mi(k)hez kapcsolódik a CORBA rendszerben az "IDL csontváz" (angolul: "IDL Skeleton") ? (4 pont)

- | | |
|--|---|
| <input type="checkbox"/> kliens | <input type="checkbox"/> entitás bean |
| <input type="checkbox"/> ORB mag | <input checked="" type="checkbox"/> szervant |
| <input checked="" type="checkbox"/> objektum adapter | <input type="checkbox"/> üzenet-vezérelt bean |
| <input type="checkbox"/> objektum attribútum | <input type="checkbox"/> interfész háttértár (repository) |

mindegyik jó 2 pont, mindegyik hiba -0,5, de total >= 0

3. Ha a hagyományos MODULÁRIS programokból építünk komponens rendszert, akkor (4 pont)

mi a kommunikáció módja: **eljárás hívás**.....

mik az interfészek: **paraméteres eljárások, globális változók**.....

létezik-e genericitás: **esetlegesen (Ada)**.....

hogyan adaptálunk: **a kód átírása**.....

mindegyik jó 1 pont

4. Milyen kockázatkezelési stratégiákat alkalmazunk a tervezésben ? (4 pont)

A kockázat valószínűségének csökkentés

A bekövetkező káros hatások minimalizálása

A nem kívánatos eseményt követően felmerülő tevékenységek tervezése

mindegyik jó 2 pont, 2 bonuszpont van benne

5. A pozitív egészek maximum 10 elemű halmazát (minden elem különböző) az alábbi utasításokkal definiáljuk. Egészítse ki az axiómákat ! Tételezze fel, hogy *i* csak pozitív egész lehet. Használhatja az egészeken értelmezett összehasonlító műveleteket. (7 pont)

CRT() új (üres) halmazt hoz létre.
INS(s, i) az *s* halmazhoz adja az *i* pozitív egészet, ha a halmaz elemeinek száma nem nő 10 fölé. Ellenkező esetben a művelet hatástalan.
MBR(s, i) igaz, ha az *i* pozitív egész az *s* halmaz eleme.
BIG(s) az *s* halmazból a legkisebb elem eltávolításával kapott halmazt adja meg.
MIN(s) megadja az *s* halmaz legkisebb elemének értékét, vagy 0-t, ha a halmaz üres.
SIZE(s) megadja az *s* halmaz elemeinek számát

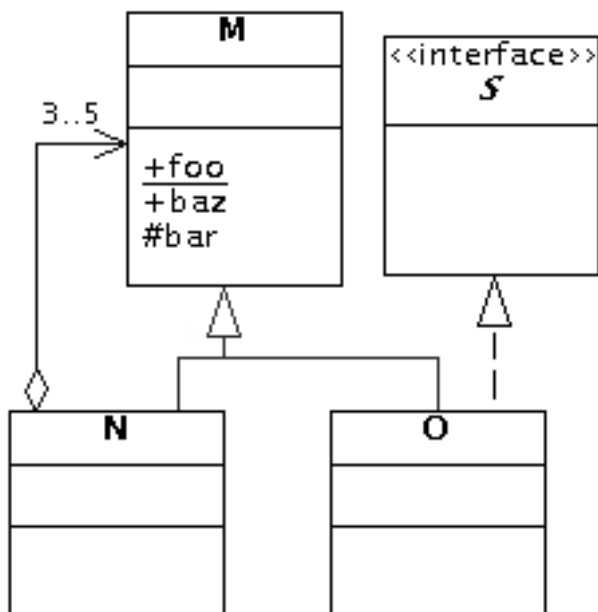
SIZE(CRT()) = 0
 MIN(CRT()) = 0
 BIG(CRT()) = CRT()
 MBR(CRT(), i) = false
 SIZE(INS(s, i)) = if (SIZE(s) < 10 && !MBR(s, i)) SIZE(s)+1 else SIZE(s)
 MIN(INS(s, i)) = if (SIZE(s) == 0 || SIZE(s) < 10 && i < MIN(s)) i else MIN(s)
 MBR(INS(s, i), j) = if (SIZE(s) < 10 && !MBR(s, i)) i==j || MBR(s, j) else MBR(s, j)
 BIG(INS(s, i)) = if (SIZE(s) == 0) s
 else if (SIZE(s) >= 10) BIG(s)
 else if (MBR(s, i)) BIG(s)
 else if (i < MIN(s)) s else INS(BIG(s), i)

mindegyik jó 1 pont

6. Az *M* osztálynak három metódusa van: *foo()*, *bar()* és *baz()*. *foo()* és *baz()* publikus, *bar()* protected. *foo()* osztályszintű.

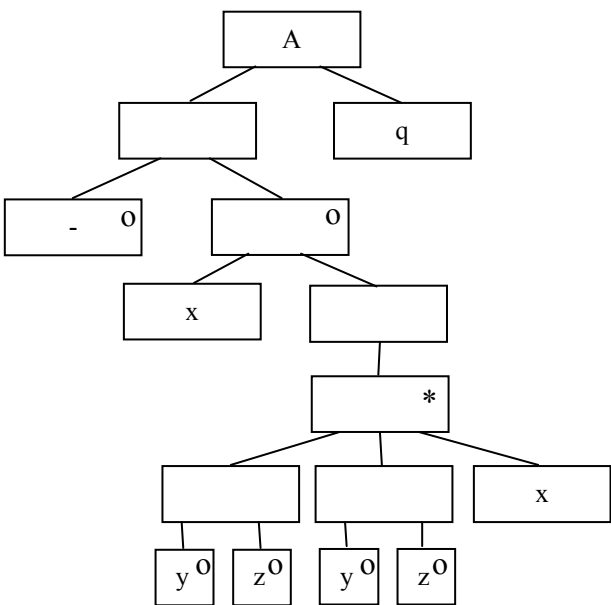
Az *M* osztálynak két leszármazottja van: *N* és *O*. *N* tartalmaz legalább 3, de legfeljebb 5 *M* osztályú objektumot. *O* megvalósítja az *S* interfészt.

Készítsen a fenti leírás alapján **UML osztálydiagramot** ! (7 pont)



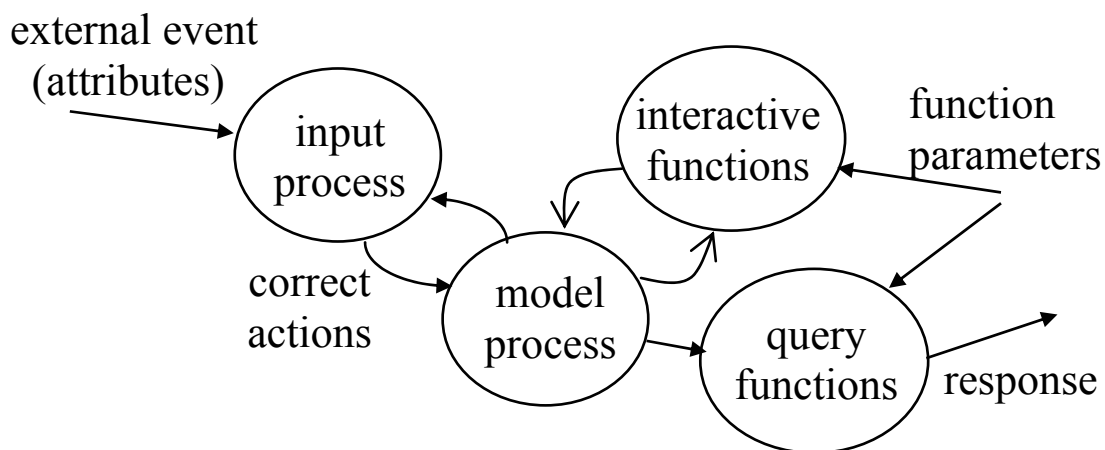
foo, baz, bar, aggregálás, interfész, implementálás, öröklés 1-1 pont

7. Legyen egy A entitás, amelyen az x, y, z és q események fordulhatnak elő. Rajzolja fel az A entitás élettörténetét a JSD szerint, ha az események sorrendjét az alábbi BNF leírással definiáltuk ! (6 pont)



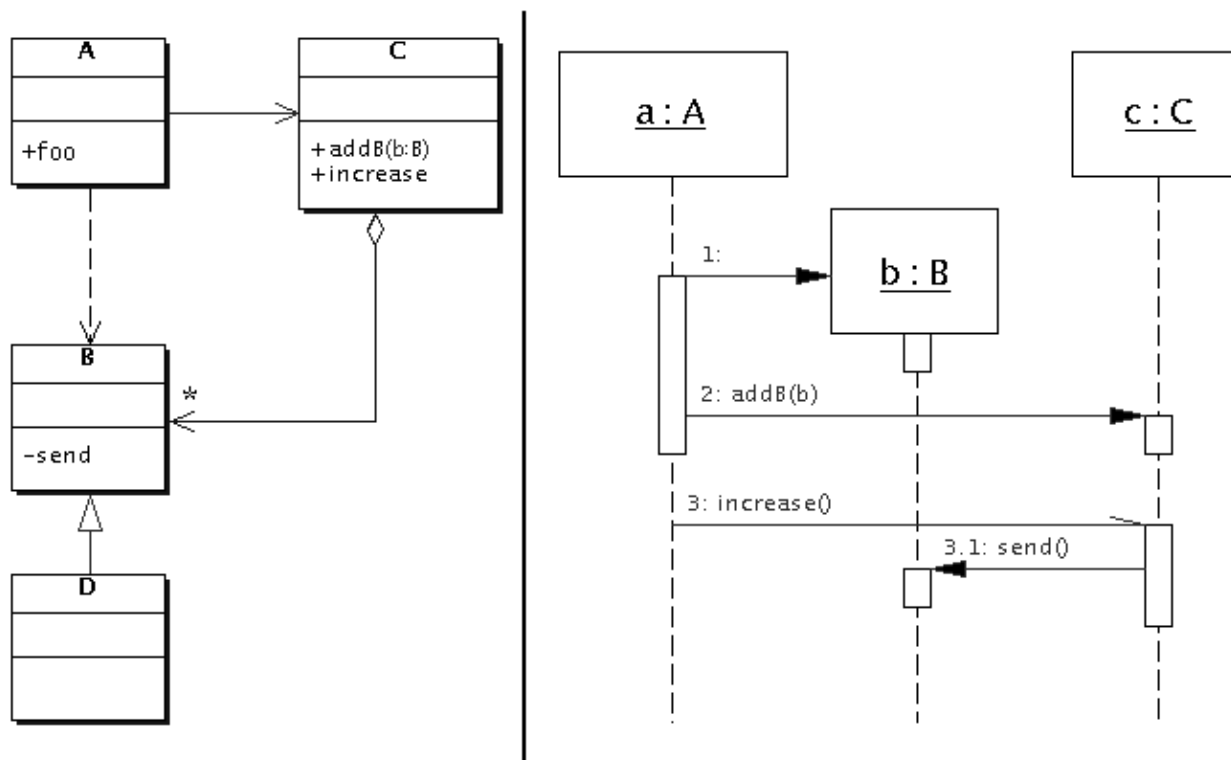
-, x, *, yz, x q 1 pont

8. Rajzolja fel DFD ábrán, hogy a JSD-ben a modell és funkció processzek hogyan kapcsolódnak egymáshoz és a külvilághoz. (6 pont)



mindegyik item (gombóc és felirat) 0,5 pont

9. Az alábbi (összefüggő) UML diagramok alapján értékeljen minden állítást az alábbi kulcs segítségével ! (7 pont)



- B** - csak a második tagmondat igaz
- C** - mindkét tagmondat igaz, de a következtetés hamis
- D** - mindkét tagmondat igaz és a következtetés is helyes
- E** - egyik tagmondat sem igaz
- F** - az ki van zárva, mert nem tudom)

- [D]** A es B osztály között dependencia van, mert A osztályú objektum hoz létre B osztályút.
- [C]** c objektumnak küldött `increase()` üzenet aszinkron, mert c nem ismeri a-t.
- [A]** c nem küldhetne `send()` üzenetet b-nek, mert a `B::send()` `protected`.
- [E]** b küldhetne `foo()` üzenetet a-nak, mert a B osztály A leszármazottja.
- [B]** D helyettesíthető B-vel, mert D osztály B leszármazottja
- [C]** a c objektumnak `addB()` üzenet többször is küldhető, mert a c objektum több B osztályú objektumot is aggregál.
- [B]** a `send()` üzenetnek aszinkronnak kellene lennie, mert az `increase()` üzenet aszinkron volt.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total ≥ 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2003. december 23. 9:30

1. Egy adatfolyam-ábrán a P1, P2 és P3 processzek, valamint a DB1 adattár(store) szerepelnek. A vezérlőegység bemenetét a következő specifikáció írja le: [szende|szundi|tudor]. Töltse ki a vezérlőegység állapotábráját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) az adatfolyam és adatspecifikációval ! (5 pont)

bb				
S13				

az oszlopok paraméterezése (sorrend lényegtelen) csak a bemenet konstansai lehetnek 1 pont
a következő állapotok csak a bejelölt két állapot (bb, S13) vagy don't care lehet, kell benne lenni állapotváltásnak is. 2 pont
az akció ha nem üres, csak processz (P1, P2, P3) lehet és mindegyiknek legalább egyszer szerepelni kell. 2 pont

2. Mi(k)hez kapcsolódik a CORBA rendszerben az "IDL betét" (angolul: "IDL Stub") ? (4 pont)

- | | |
|--|---|
| <input type="checkbox"/> implementációs háttértár (repository) | <input type="checkbox"/> entitás bean |
| <input checked="" type="checkbox"/> ORB mag (core) | <input type="checkbox"/> szervant |
| <input type="checkbox"/> objektum adapter | <input checked="" type="checkbox"/> kliens |
| <input type="checkbox"/> objektum attribútum | <input type="checkbox"/> interfész háttértár (repository) |

mindegyik jó 2 pont, mindegyik hiba -0,5, de total >= 0

3. Ha hagyományos objektum orientált programokból építünk komponens rendszert, akkor (4 pont)

mi a kommunikáció módja: **polimorf eljárás hívás**.....

mik az interfészek: **polimorf metódusok, osztály és objektum változók**.....

létezik-e genericitás: **gyakran előfordul (pl. C++)**.....

hogyan adaptálunk: **öröklés, delegálás**.....

mindegyik jó 1 pont

4. Melyek a kockázatkezelés folyamatai (risk management process) ? (4 pont)

Kockázatok azonosítása (identification)

Kockázatok elemzése (analysis)

Kockázatok tervezése (planning)

Kockázat felügyelet (monitoring)

mindegyik jó 1,5 pont, 2 bonuszpont van benne

5. A pozitív egészek maximum 10 elemű csomagját (azonos elemek többször is előfordulhatnak) az alábbi utasításokkal definiáljuk. Egészítse ki az axiómákat ! Tételezze fel, hogy *i* csak pozitív egész lehet. Használhatja az egészezen értelmezett összehasonlító műveleteket. (7 pont)

CRT() új üres csomagot hoz létre.
INS(b, i) a **b** csomaghoz adja az **i** pozitív egészet, ha a csomag elemeinek száma nem nő 10 fölé. Ellenkező esetben a művelet hatástalan.
NBR(b, i) megadja, hogy a **b** csomagban az **i** pozitív egész hány példányban van meg.
RMV(b, i) a **b** csomagból az összes **i** pozitív egész eltávolításával kapott csomagot adja meg.
DEL(b, i) a **b** csomagból egyetlen **i** pozitív egész eltávolításával kapott csomagot adja meg.
SIZE(b) megadja a **b** csomag elemeinek számát

SIZE(CRT()) = 0

NBR(CRT(), i) = 0

RMV(CRT(), i) = CRT()

DEL(CRT(), i) = CRT()

SIZE(INS(b, i)) = if (SIZE(b) < 10) SIZE(b) + 1 else SIZE(b)

NBR(INS(b, i), j) = if (SIZE(b) < 10 && i == j) NBR(b, j) + 1 else NBR(b, j)

RMV(INS(b, i), j) = if (SIZE(b) < 10 && i != j) INS(RMV(b, j), i)
 else RMV(b, j)

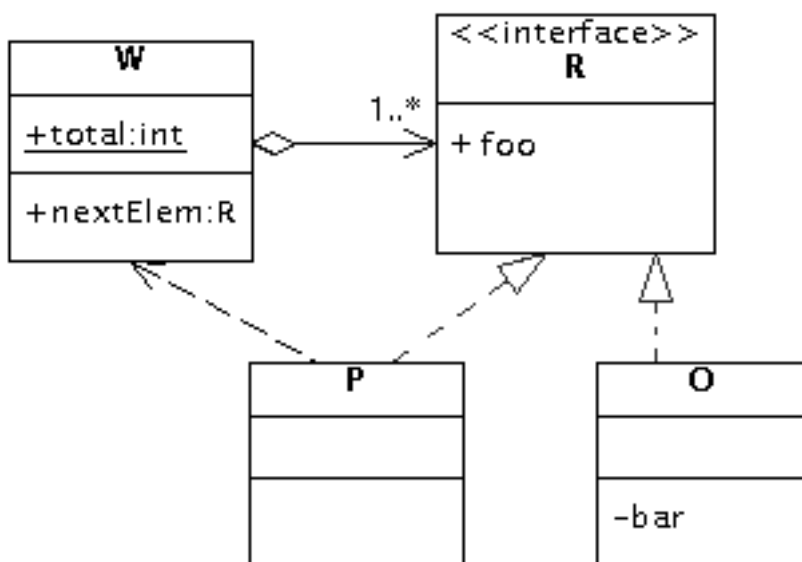
DEL(INS(b, i), j) = if (SIZE(b) >= 10) DEL(b, j)

else if (i == j) b else INS(DEL(b, j), i)

6. A *W* osztálynak van egy publikus, *total* nevű, osztályszerű attribútuma, valamint egy publikus *nextElem* metódusa, amely *R* interfészi objektumot ad visszatérési értéként. A *W* osztály tetszőlegesen sok (de legalább egy) *R* interfészi objektumot tartalmazhat.

Az *R* interfésznek egy publikus *foo()* metódusa van. Az interfészt két osztály valósítja meg, *O* és *P*. *O*-ban van egy privát *bar()* metódus, míg *P* függ *W*-től.

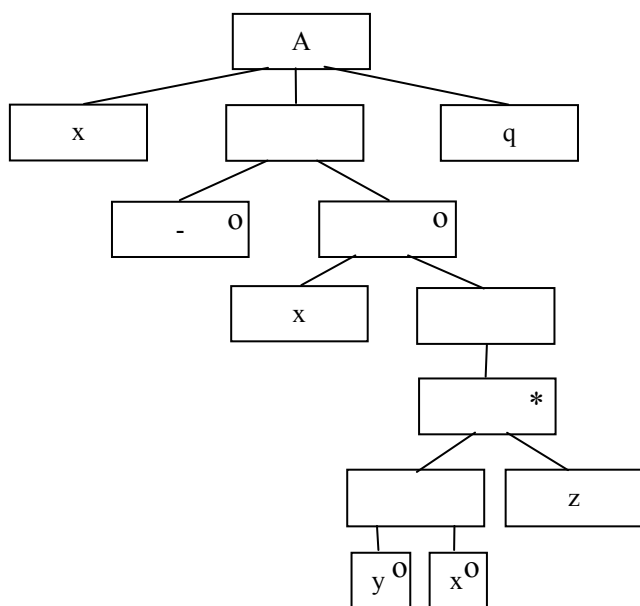
Készítsen a fenti leírás alapján **UML osztálydiagramot** ! (7 pont)



Total, nextElem, aggregáció, interfész, (foo, bar), 2implementálás, függőség 1-1 pont

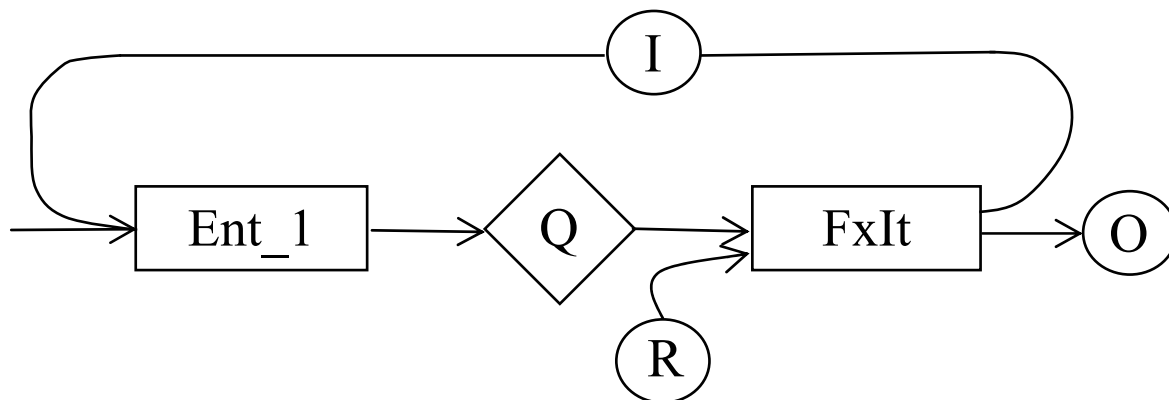
7. Legyen egy A entitás, amelyen az x, y, z és q események fordulhatnak elő. Rajzolja fel az A entitás élettörténetét a JSD szerint, ha az események sorrendjét az alábbi BNF leírással definiáltuk ! (6 pont)

$x[x\{[y|x]z\}]q$



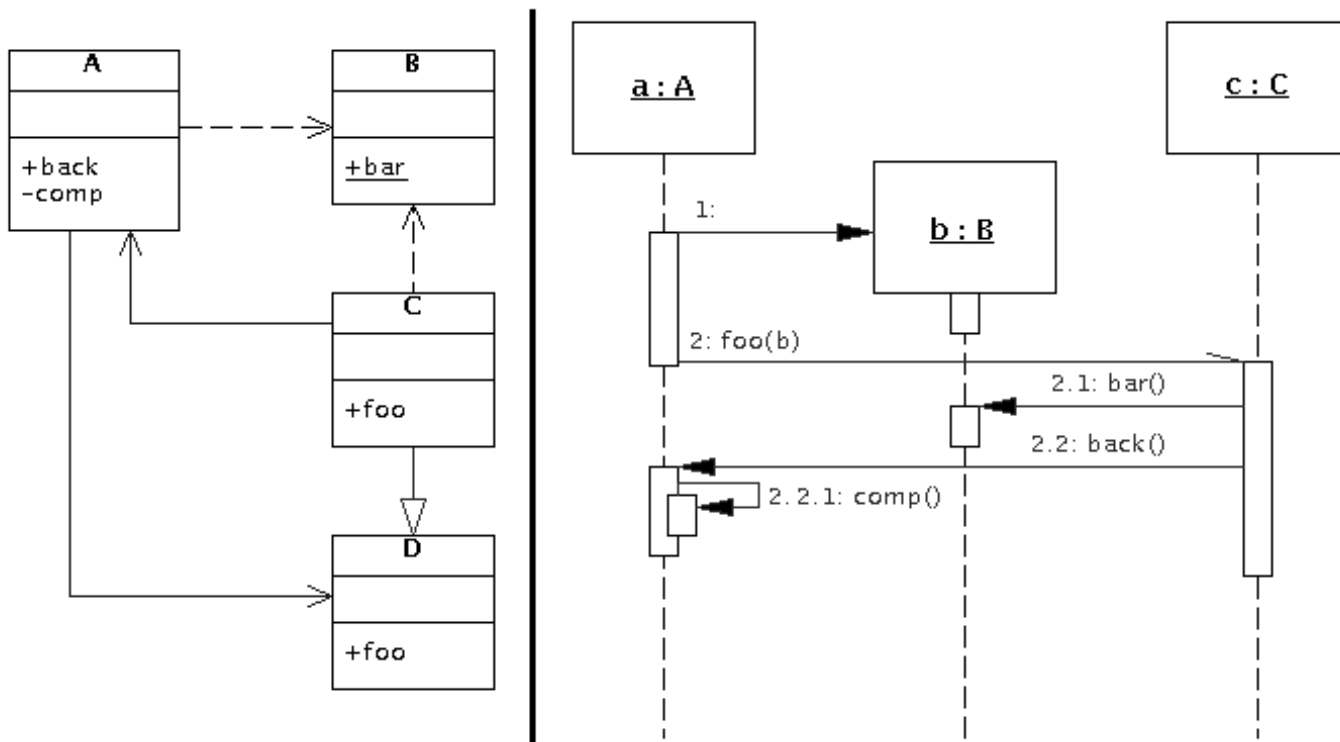
minden esemény a jó helyen (1 szinttel feljebb) 1 pont

8. A JSD-ben az Ent_1 modell processzhez kapcsoljon a modell állapotán alapuló, felhasználói kérésre induló FxIt iteratív funkciót ! Rajzoljon SSD-t ! (6 pont)



mindegyik korrektul kapcsolódó item 1 pont, rossz kapcsolattal 0,5 pont

9. Az alábbi (összefüggő) UML diagramok alapján értékeljen minden állítást az alábbi kulcs segítségével ! (7 pont)



- A - csak az első tagmondat igaz
- B - csak a második tagmondat igaz
- C - mindkét tagmondat igaz, de a következtetés hamis
- D - mindkét tagmondat igaz és a következtetés is helyes
- E - egyik tagmondat sem igaz
- (F - az ki van zárva, mert nem tudom)

- [B] az a objektum nem hozhatná létre b-t, mert az A osztály függ B osztálytól.
- [B] az a objektum nem hívhatná meg c foo () metódusát, mert csak D osztályt ismeri, C-t nem
- [C] c objektum a foo () metódushívás végrehajtása után nem őrzi meg b-t, mert c ismeri a-t.
- [D] D osztályú objektum helyettesíthető C osztályúval, mert C osztály a D leszármazottja.
- [A] foo () üzenet aszinkron, mert a C.foo () metódus protected
- [A] c küldhet b-nek bar () üzenetet, mert a B.bar () metódus nem osztályszintű
- [E] az a nem küldhet magának comp () üzenetet, mert az A.comp () nem privát.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

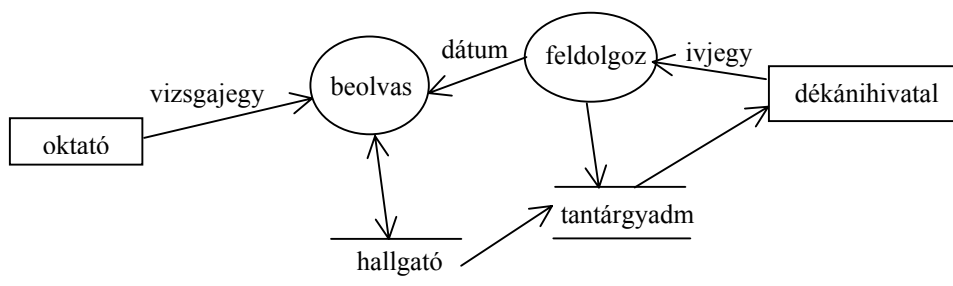
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2003. január 21.

1. Milyen szintaktikai hibákat talál az alábbi adatfolyam ábrán ? (6 pont)



- ha ez DFD, akkor nincs terminátor, ha ez ContextDiag, akkor nincs 2 processz és adattár
- **beolvasból** nincs kimenő adat, ezért a **hallgatóból beolvasba** mutató nyílhegy hiba.
- **tantárgyadm** (adattár) nem köthető a **dékánihivatalhoz** (terminátor)
- **hallgató** és **tantárgyadm** (két adattár) közvetlenül nem köthető össze.

minden felismert hiba 2 pont, minden tévedés -1 pont (2 bonus pont, ha megtalálta mind a 4 hibát)

2. A “végrehajtási szemantika (execution semantics)” mely OO alapfogalmakhoz társítható ? (2 pont)

- | | |
|---|--|
| <input type="checkbox"/> az osztályhoz | <input checked="" type="checkbox"/> az operációhoz |
| <input type="checkbox"/> az objektumhoz | <input type="checkbox"/> a metódushoz |

jó válasz 2 pont, egyébként 0

Az OO alapmodell milyen végrehajtási szemantikákat definiál és azokat hol és hogyan jelöli ? (4 pont)

at-most-once

—

best effort

oneway a szignatúrában

minden jó item 1 pont

3. Az UML komponensek fajtái: (6 pont)

deployment (a sw indításához kell, pl class)

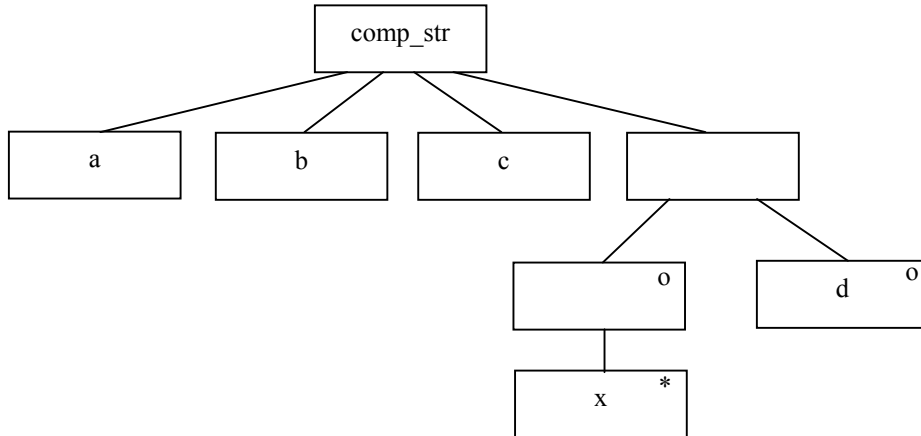
execution (a végrehajtás közben keletkezik, pl file)..

work product (a fejlesztő munka terméke, pl *.h) .

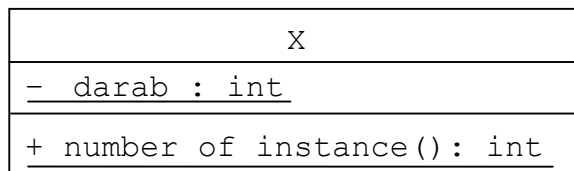
.....

minden helyes válasz 2 pont

4. Írja fel az alább algebrailag leírt `comp_str` nevű összetett adatstruktúrát a JSD-ben alkalmazott jelöléstechnikával ! Egy `z` attribútum egyszeri előfordulását jelölje a `z` esemény ! (3pont)

$$\text{comp_str} = a + b + c + [\{ x \} | d]$$


5. Az `X` osztályon értelmezünk egy `number_of_instance()` operációt, amely megadja, hogy az osztályból hány darab példány keletkezett. Rajzolja meg az `X` osztály **UML osztálydiagramját**, feltüntetve az operációt és az annak megvalósításához szükséges attribútumot ! (4 pont)



az attribútum és az operáció 2 – 2 pont

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

Szarumán elhatározta, hogy leszámol Lovasvéggel, ezért Vasudvard pincéinek legmélyén orkok gyártásába kezdett (mint tudjuk, az orkok csak sötétben gyárthatók). Minden új orkot próbának vetnek alá, hogy megállapítsák képességeit. Akik el tudják viselni a napfényt, azok uruk-hai orkok lesznek, és szekercét, a többiek görbe kardot kapnak fegyverül. A kiképzés során az orkok gyakorlatoznak, amelyek alkalmával gyakran okoznak maguknak vagy egymásnak sérüléseket, elveszítik valamely testrészüket (nagylábujj, orr, jobb kar), de a sérülésekbe nem hálnak bele. Az orkok között vannak olyan – különösen természetes – példányok, akik soha nem végeznek gyakorlatot. Minden uruk-hai ork közvetlenül a csata előtt a szekerce mellé kap még egy kardot is. Az olyan uruk-hai orkok, akik soha nem gyakorlatoztak, a csata előtt – az ellenfél elrettentése érdekében – a kapott karddal levágják saját jobb fülüket.

Szarumán a legvadabb orkok közül választja meg tisztjeit, akik a kinevezést követően is részt vesznek a gyakorlatozásban. Ha valamelyikük túl sok sérülést okoz másnak vagy kap, azt lefokozzák. Az egyszer lefokozott orkot többé nem nevezik ki. A legjobb tisztet Szarumán előlépteti főtisztté. Főtisztet már nem fokoznak le.

Az orkok egyszer használatosak, a csatában valamennyi ork elpusztul.

Definiálja az alábbi funkciókat ellátó processzeket:

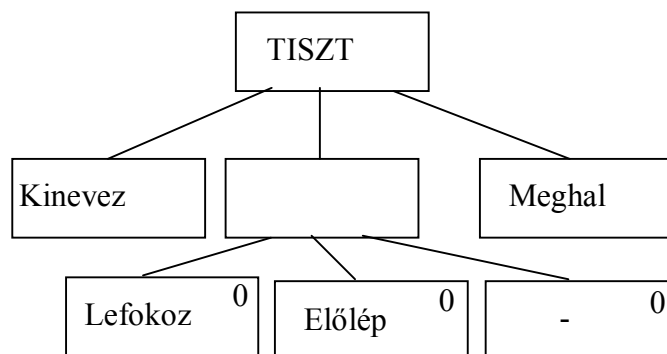
Azonnal jelezzen, ha egy ork az ötödik sérülését szerezte !

Kérésre listázza ki azokat a lefokozott tiszteteket, akik a csata előtt megkapták a kardot és esedékes a saját jobb fülük levágása !

Kérésre adja meg a főtiszték listáját !

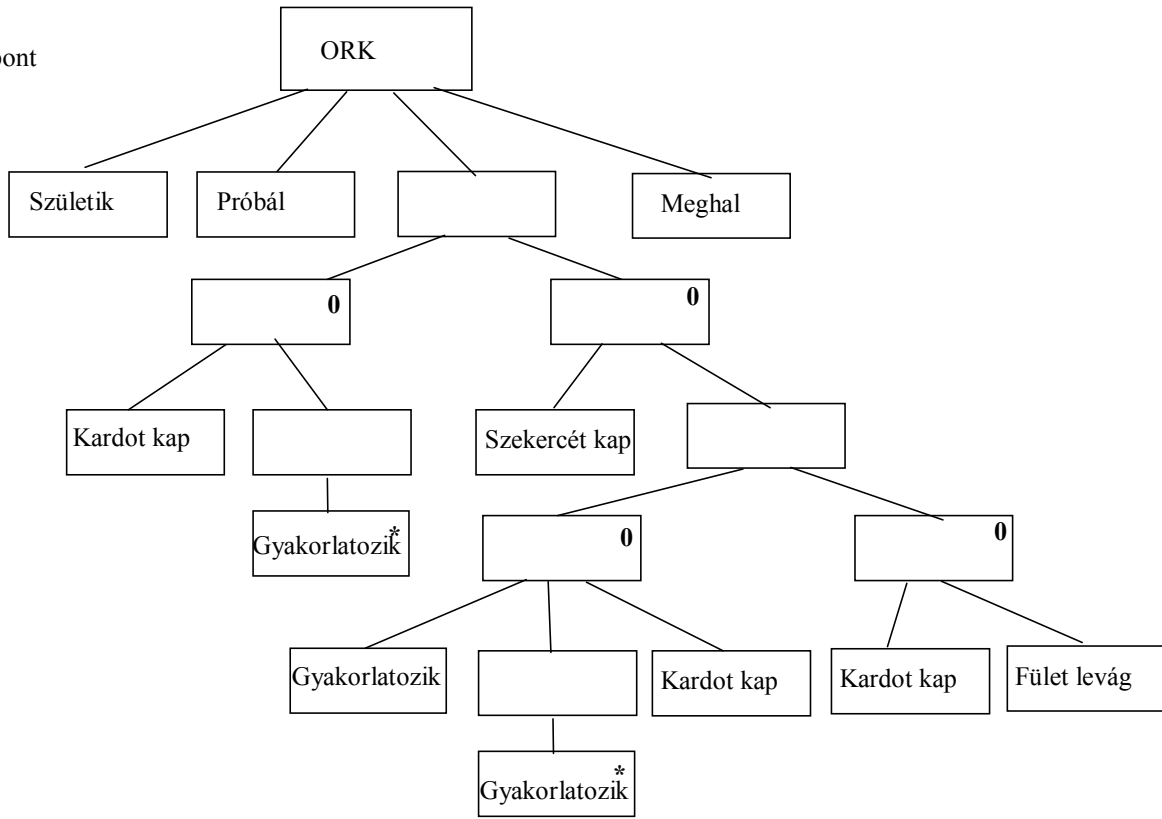
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

- születik (ORK) 2 pont
- próbál (ORK)
- kardot kap (ORK)
- szekercét kap (ORK)
- kinevez (TISZT)
- gyakorlatozik (ORK)
- fület levág (ORK)
- lefokoz (TISZT)
- előlép (TISZT)
- meghal (ORK, TISZT)

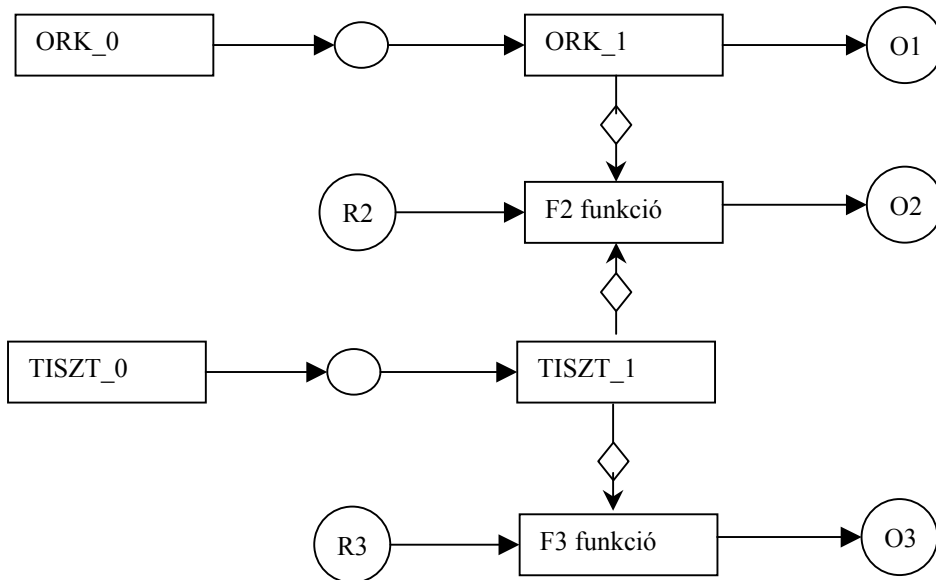


2 pont

5 pont

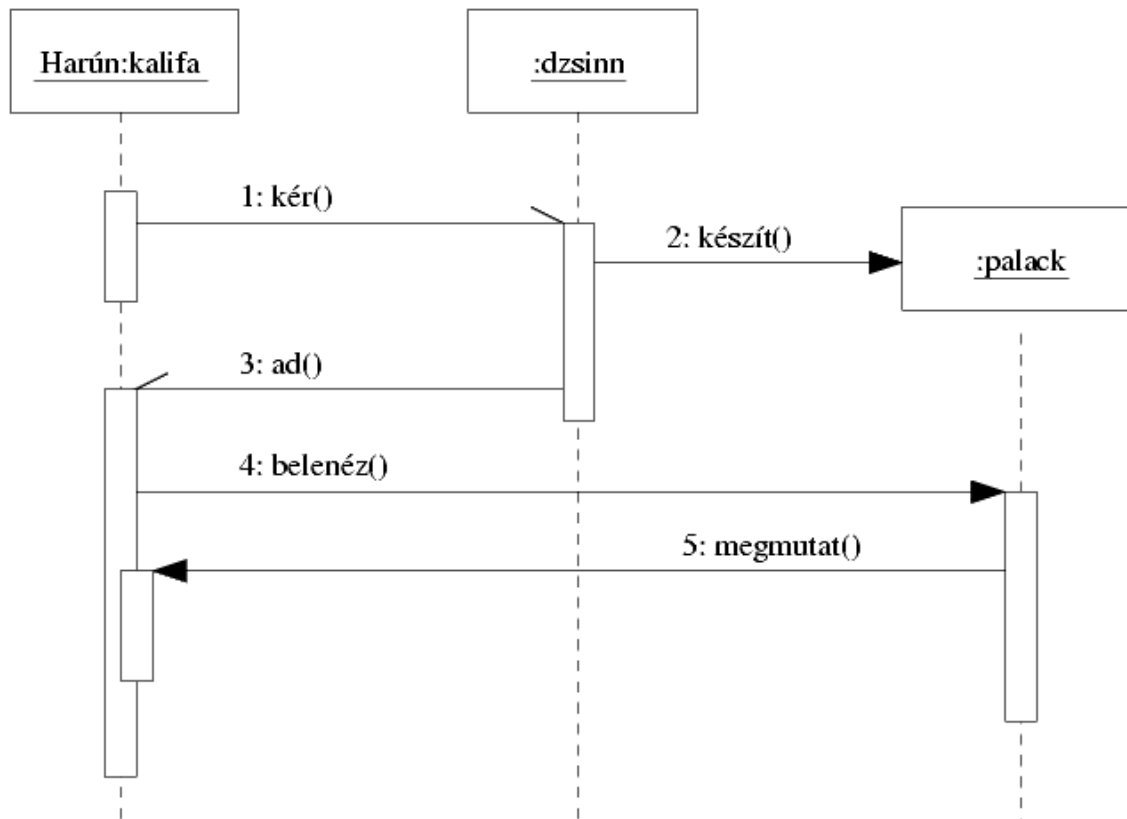


teljes SSD 2 pont



7. Harún al-Rasíd, a bagdadi kalifa üzen az egyik dzsinnjének, amelyben kéri, hogy az készítse neki egy varázspalackot. (Ez még a dzsinnek is több napot vesz igénybe.) Mikor a palack elkészül, a dzsinn átadja azt a kalifának, majd dolgára megy. A kalifa nem vár, hanem egyből belenéz a palackba, amely megmutatja neki, hogy hol van a Próféta saruja.

Készítsen a történet alapján **UML szekvencia-diagramot** ! (7 pont)



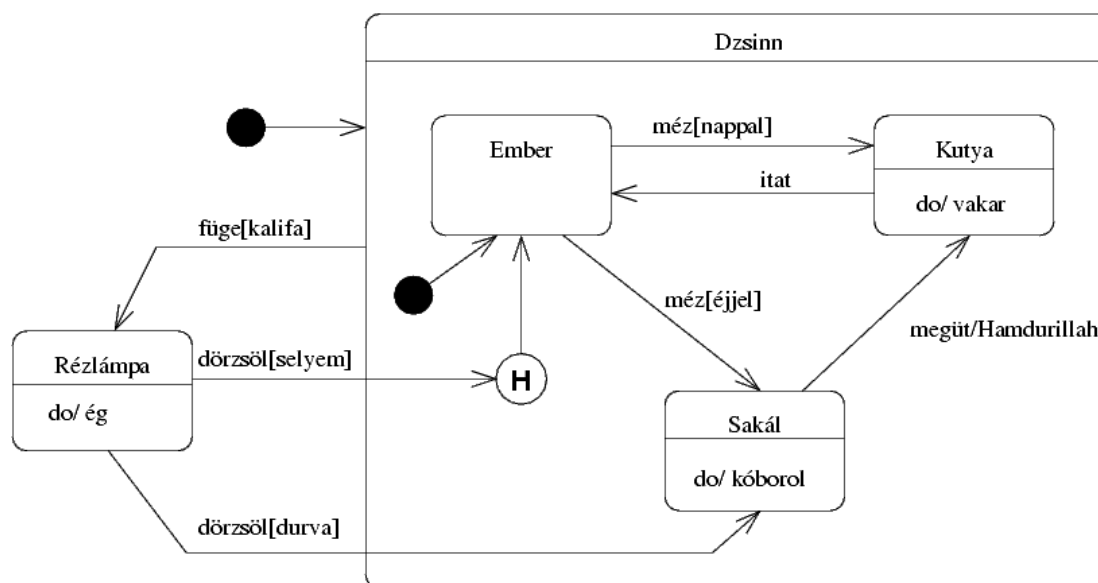
minden helyes üzenet, a hozzá tartozó korrekt bárral együtt 1 pont. 1-1 pont jár a :dzsinn-ért és Harún-ért.

8. Egy karaván Harún al-Rasíd kalifa udvarába hozott egy dzsinn. A dzsinn, elmondásuk szerint, a sivatagban, egy homokviharban született, ott leltek rá.

A dzsinn alapvetően egy ember alakját viselte. Ha nappal mézet kapott, akkor kutyává változott. Kutyaként addig vakarta a füle tövét, amíg meg nem itatták, mire ismét emberi formát öltött. Ha éjjel kapta a mézet, akkor viszont sakálként a sivatagban kóborolt, és csak akkor változott a fenti kutyává, ha egy nádpálcával megütötték. Az ütésre azt kiáltotta: Hamdurillah.

Mikor a kalifa a dzsinn (bármely formájában) fűgével etette, a dzsinn rézlámpássá változott, és kifogyhatatlanul égett. Másától kapott fűgére rá sem hederített. Ha a lámpást a legfinomabb selyemmel dörzsölték, visszanyerte a lámpássá alakulás előtti formáját. Ha durvább anyagot használtak, akkor a sakál képét vette fel.

Rajzolja meg a dzsinn **UML állapot-diagramját** ! (7 pont)



minden állapot, minden aktivitás, minden korrekt állapotátmenet 0,5 pont. A historyból kimutató predefinit belépés bárhová mutathat, mert az nem definiált.

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2003. január 7.

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (7 pont)

Egy videotékában DVD-ket lehet kölcsönözni. A DVD-kről nyilvántartják a címét, a kiadóját, a kiadás évét, a műsor hosszát, valamint kategóriáját (pl.: zene, akció, romantikus, vígjáték, krimi, thriller, sex, stb.). Egy DVD-ből több példány is lehet, de olyan DVD-k adatait is tárolják, amelyekből még vagy már nincs példány. Példányonként tárolják a vásárlás idejét, az árat, a szállítót és az adott példány kölcsönzéseinek számát. A szállítóról ismerik a nevét, címét és bankszámla számát. Az ügyfelet első kölcsönzésük alkalmával veszik nyilvántartásba. Az ügyfelet azonosító jellemzi, de nyilvántartják a nevet, címet, telefonszámot és az éves tagdíj lejárat dátumát is. Egy kölcsönzés alkalmával egy ügyfél egy DVD példányt megadott határidőig kölcsönbe vesz. A kölcsönzés megszűnik, ha az ügyfél a DVD-t visszahozta. Az ügyfelekről profilt készítenek, amely tartalmazza, hogy az ügyfél az élete során a különböző kategóriájú DVD-kből hány alkalommal kölcsönzött (Ennek alapján lehetséges az ügyfeleknek személyre szóló reklámokat küldeni).

DVD (cím, kiadó, kiadási év, hossz, kategória)

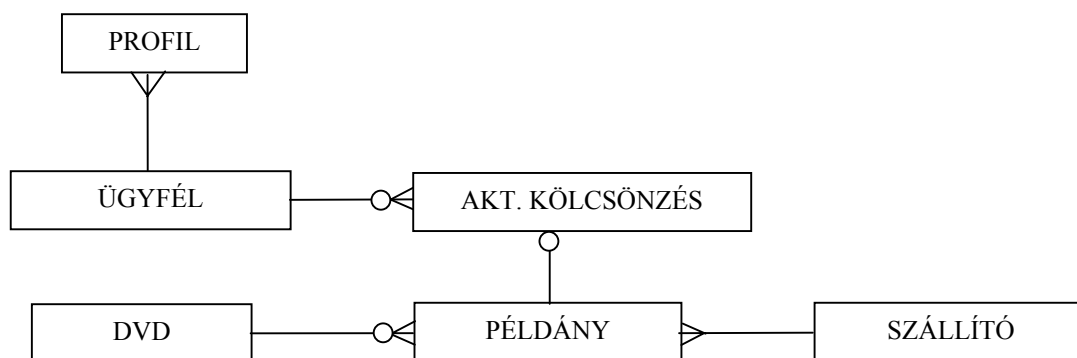
PÉLDÁNY (dvd, azonosító, vásárlás ideje, ára, kölcsönzések száma, *szállító*)

SZÁLLÍTÓ (név, cím, bankszámlaszám)

ÜGYFÉL (azonosító, név, cím, telefon, tagdíj lejárat)

AKT. KÖLCSÖNZÉS (példány, ügyfél, dátum, határidő)

PROFIL (ügyfél, kategória, kérések száma)



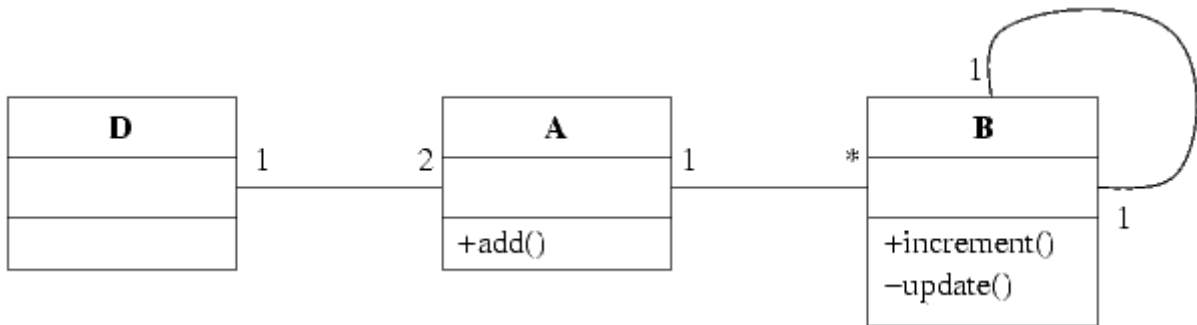
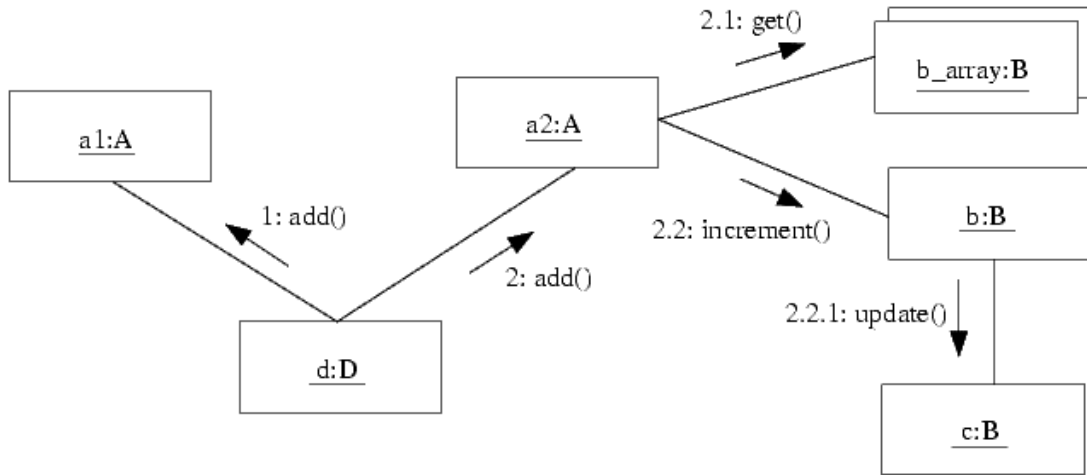
minden korrekt entitás 0,5 pont és minden jó reláció 1 pont, max. 4-et fogadunk el. Minden fölösleges reláció -0,5 pont

2. Jelölje meg a funkciópont elemzés (Function Point Analysis, FPA) komponenseit ! (5 pont)

- | | |
|--|--|
| <input checked="" type="checkbox"/> External Inputs | <input checked="" type="checkbox"/> External Interface Files |
| <input type="checkbox"/> External Memory | <input checked="" type="checkbox"/> Internal Logical Files |
| <input type="checkbox"/> Internal States | <input checked="" type="checkbox"/> External Outputs |
| <input checked="" type="checkbox"/> External Enquiry | <input type="checkbox"/> Internal Data Flows |
| <input type="checkbox"/> Internal Memory | <input type="checkbox"/> Internal Logical Interfaces |

minden helyes válasz 1 pont, minden hiba -1 pont, negatív nem lehet

3. Rajzolja meg az alábbi kollaborációs diagramban szereplő objektumok **UML osztálydiagramját**, a metódusok és azok láthatóságának feltüntetésével! A láthatóság legyen a lehető legszigorúbb, a multiplicitások esetén pedig a feltételezhető legkisebbet válassza ! (6 pont)



minden korrekt asszociáció és metódus 1 pont

4. Az UML komponensek fajtái: (6 pont)

deployment (a sw indításához kell, pl class)

execution (a végrehajtás közben keletkezik, pl file)..

work product (a fejlesztő munka terméke, pl *.h) .

.....

minden helyes válasz 2 pont

5. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A bergengóciai posta karácsonykor különleges szolgáltatással kedveskedik a bergengócoknak. Ha valaki zenés üdvözetet küld ismerősének, akkor azt legalább egyszer, de ha jut rá idő, többször is kézbesítik. (A zenés üdvözetet Vili a postás, Bergengócia legjobb torkú levélkézbesítője személyesen prezentálja.) Az első kézbesítést megelőzően még lehet extra változatot kérni. Ilyenkor az utolsó zenés üdvözet kézbesítése után (de még karácsony előtt) egy prózai üdvözet is kap a címzett (amit szintén Vili ad elő). A bergengócok persze nem tudják, hogy ez utóbbit Vili annyira utálja, hogy gyakran elszabotálja és ilyenkor még a zenés üdvözetet sem viszi ki. A posta igazgatójának utasítására karácsony után minden feladó a befizetett díj 20%-át visszakapja. A nagy rohamra való tekintettel az egyéb küldemények célba juttatásához a posta alkalmi kézbesítőket foglalkoztat (akár egyetlen napra is). Az alkalmi kézbesítők – amíg a munkaviszonyuk tart – a zenés üdvözet küldő szolgáltatást ingyen rendelkezésre tartják.

Definiálja az alábbi funkciókat ellátó processzeket:

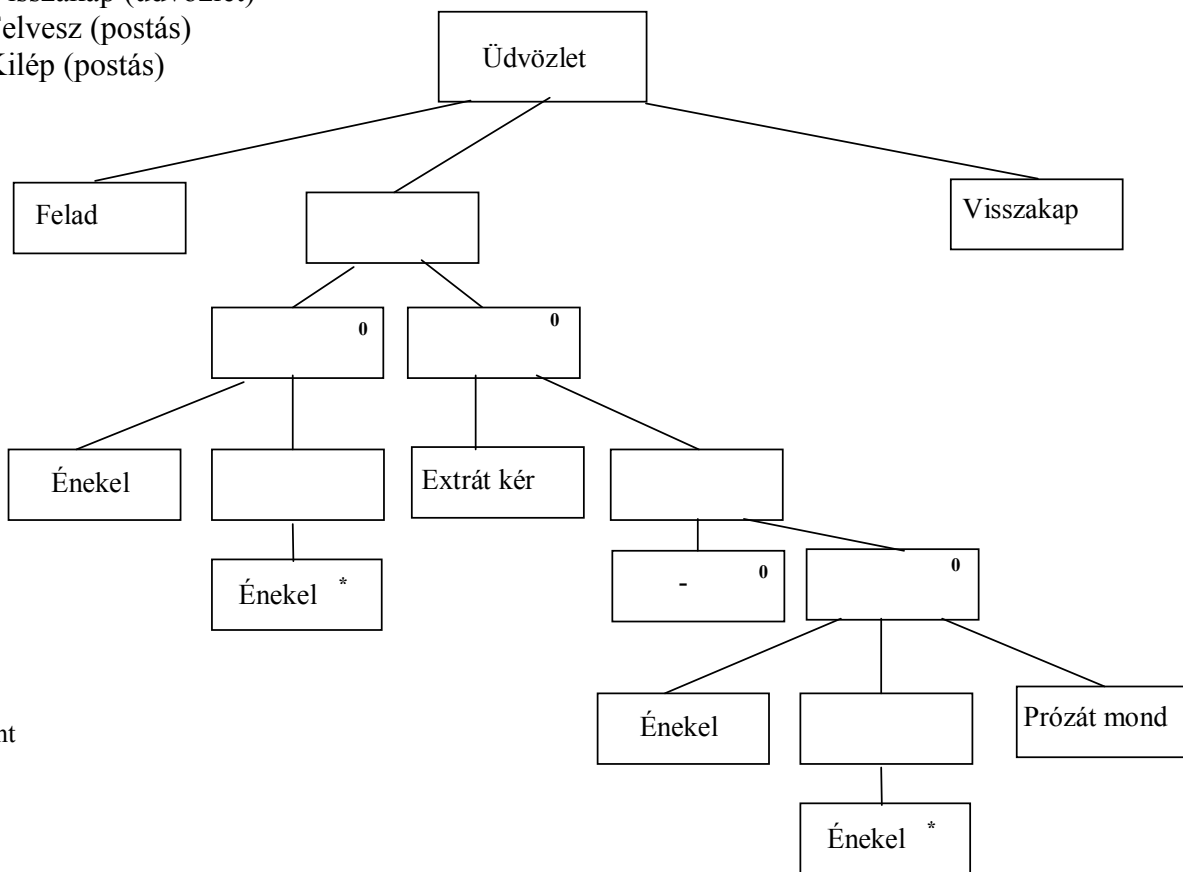
Azonnal jelezzen, ha egy alkalmi kézbesítő már az ötödik zenés üdvözetet adja fel.

Kérésre listázza ki az elszabotált üdvözetek feladóit.

Kérésre adja meg azokat a már alkalmazásban nem álló alkalmi kézbesítőket, akik zenés üdvözetet kaptak.

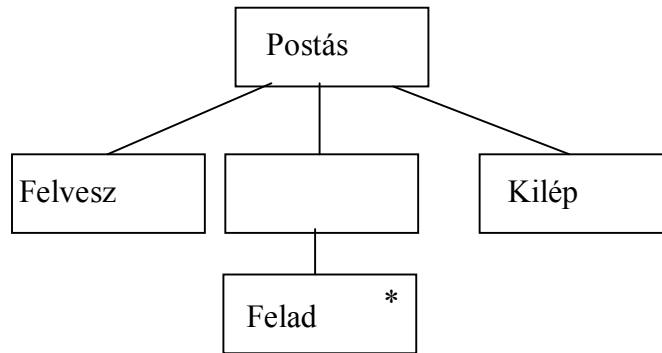
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
 kapcsolja a modellhez a **funkció processzeket.**

- Felad (üdvözet, postás) 2 pont
- Énekel (üdvözet)
- Prózát mond (üdvözet)
- Extrát kér (üdvözet)
- Visszakap (üdvözet)
- Felvesz (postás)
- Kilép (postás)

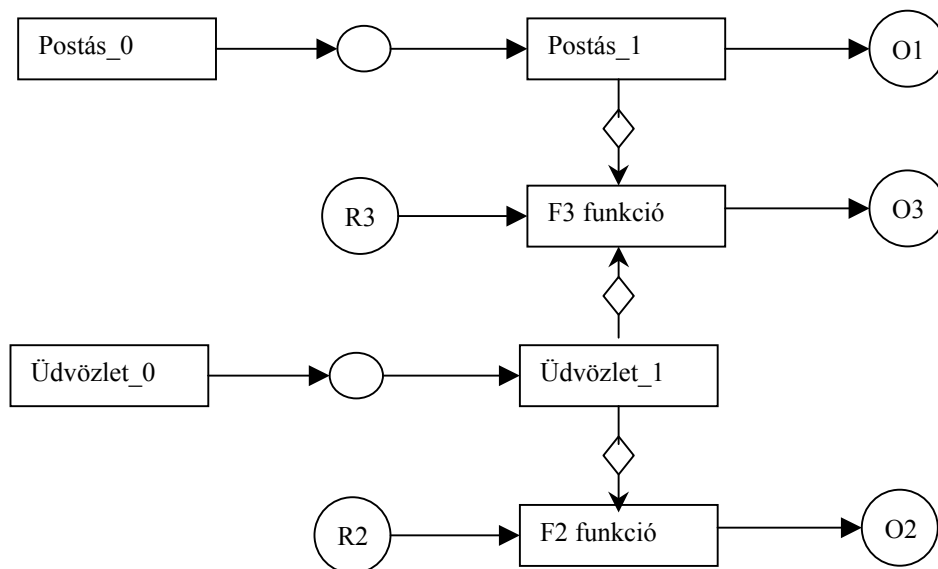


5 pont

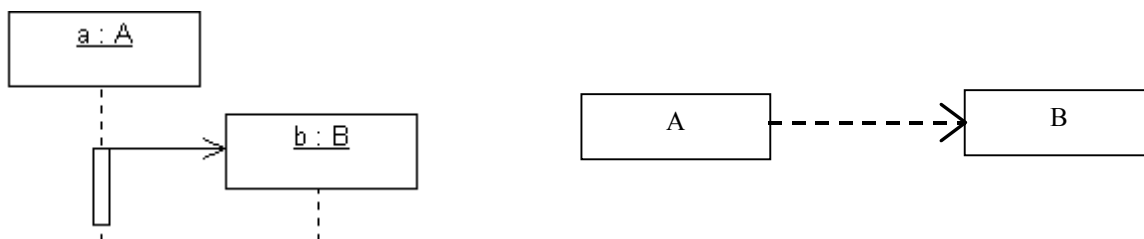
Alkalmi postás 2 pont



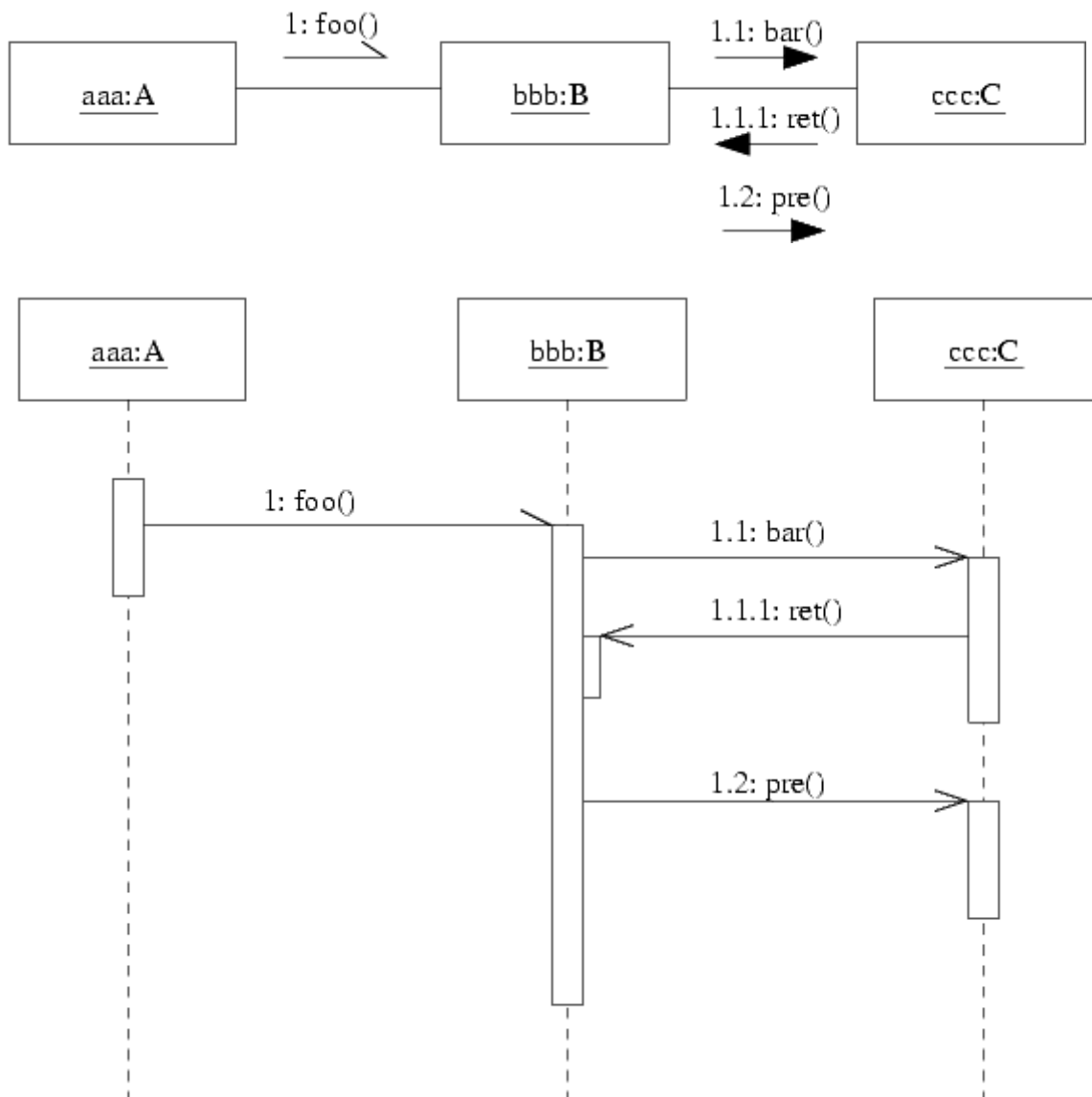
teljes SSD 2 pont



6. A szekvencia diagram alapján jelölje az A és B közötti kapcsolatot **UML osztálydiagramon!** (2 pont)

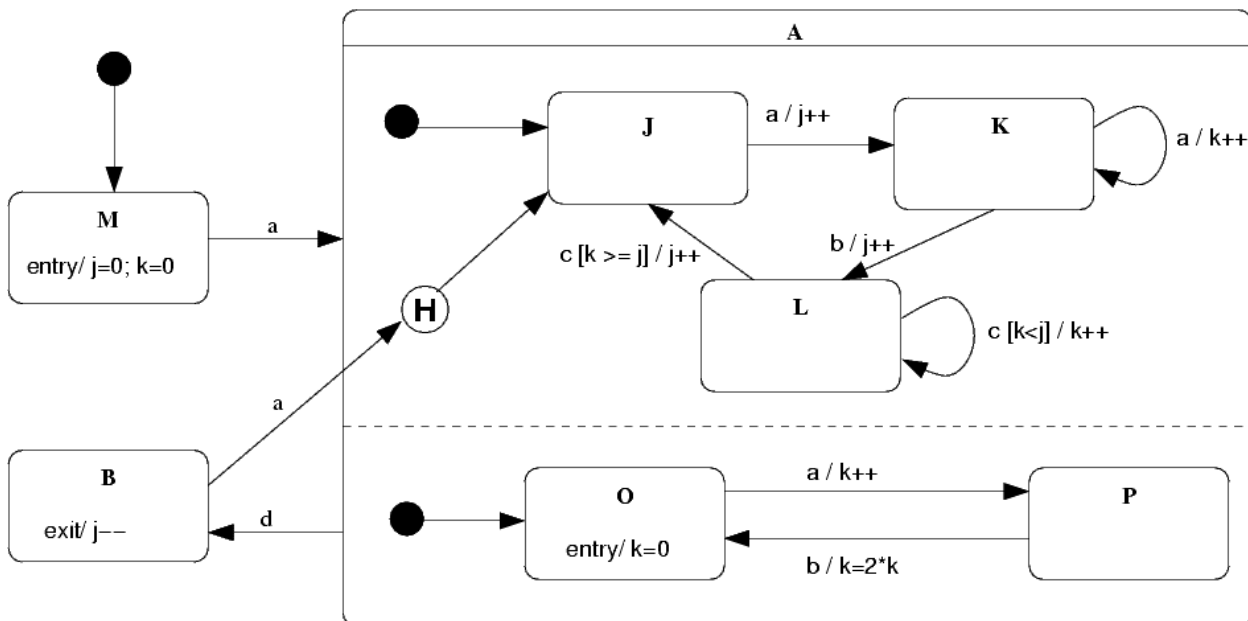


7. Az alábbi UML kollaborációs diagram alapján rajzoljon **UML szekvencia diagramot!** Ügyeljen az activation bar-ok helyes ábrázolására is! (8 pont)



minden helyes üzenet, a hozzá tartozó korrekt barral együtt 2 pont. Ha hiba van az objektummal (osztály), akkor az darabonként 1 pont levonás.

8. Az alábbi UML statechart egy objektum viselkedését írja le.



Az alábbi táblázat kitöltésével adja meg, hogy a táblázat első oszlopában álló események hatására mennyi lesz a j és k változó értéke, valamint mi lesz a következő állapot ! A táblázat első oszlopa az objektumot egymás után ért eseményeket mutatja. Az első sorban szereplő esemény előtt az objektum még nem kapott eseményt. A feladat a táblázat üres helyeinek kitöltése az adott sorban szereplő esemény teljes feldolgozása után. (5 pont)

esemény	j	k	állapot
a	0	0	A(J,O)
a	1	1	A(K,P)
a	1	2	A(K,P)
b	2	0	A(L,O)
c	2	1	A(L,O)
a	2	2	A(L,P)
d	2	2	B
a	1	0	A(L,O)
c	1	1	A(L,O)
c	2	1	A(J,O)

minden helyes sor 0,5 pont

Eredmények értékelése:

Pontszám Osztályzat

21 - 2
 28 - 3
 35 - 4
 42 - 5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2003. január 7.

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (7 pont)

A múzeumban tárgyakat állítanak ki. Minden tárgynak van egy egyedi azonosítója, de ismert a megnevezése, értéke és kategóriája (dísz tárgy, használati tárgy, fegyver, könyv, stb.) is. A tárgyakat tárolókban helyezik el. A tárolónak van egy száma, tudják róla, hogy melyik teremben van és milyen a fajtája (asztal, üveges vitrin, polc, stb.). Egy tárolóban legalább egy tárgy található. A kiállított tárgyak között vannak olyanok is, amelyeknek nem a múzeum a tulajdonosa, hanem kölcsönbe kapták. Ezekről ismert, hogy mettől meddig tart a kölcsönzés, melyik biztosító biztosítja a tárgyat és kitől kapták kölcsönbe. A kölcsönadó lehet magánszemély vagy intézmény. Magánszemély esetében nyilvántartják a személy nevét, címét és telefonszámát. Intézmény esetén annak nevét és a főhatóság elnevezését. Ugyanazon személytől vagy intézménytől több tárgy is lehet kölcsönben. Csak azon személyek és intézmények adatait tárolják, akiktől aktuálisan kölcsönöz a múzeum. A múzeum a tulajdonát képező tárgyakat alkalmanként átadja más kiállításokon történő bemutatásra. Ez esetben a tárgyat egy cédulával helyettesítik a tárolóban. A cédulán szerepel annak sorszáma, a kiállítás dátuma, a lejárat időpontja és az engedélyező személy neve.

TÁRGY (*tároló*, megnevezés, kategória, érték, azonosító)

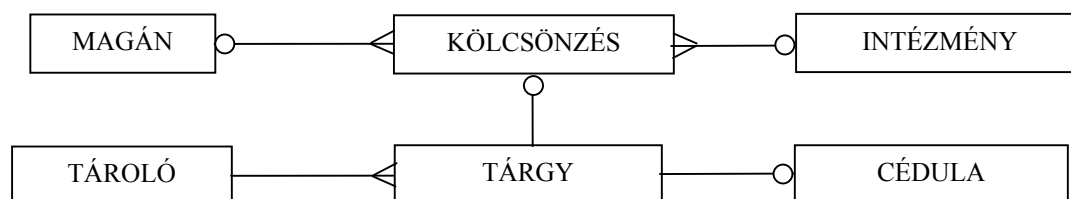
TÁROLÓ (szám, terem, fajta)

CÉDULA (*tárgy*, szám, dátum, határidő, engedélyező)

KÖLCSÖNZÉS (*tárgy*, mettől, meddig, biztosító, *magán*, *intézmény*)

MAGÁN (név, cím, telefon)

INTÉZMÉNY (név, főhatóság)



minden korrekt entitás 0,5 pont és minden jó reláció 1 pont, max. 4-et fogadunk el. Minden fölösleges reláció -0,5 pont

2. A minőség az alábbi nézőpontokból vizsgálható (views of quality): (5 points)

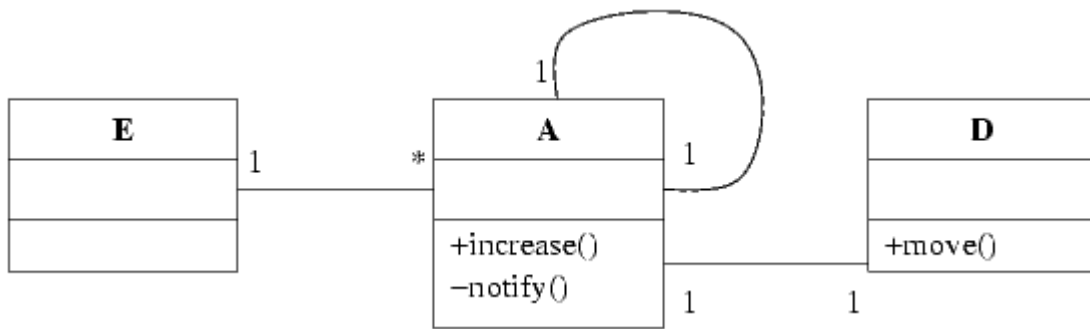
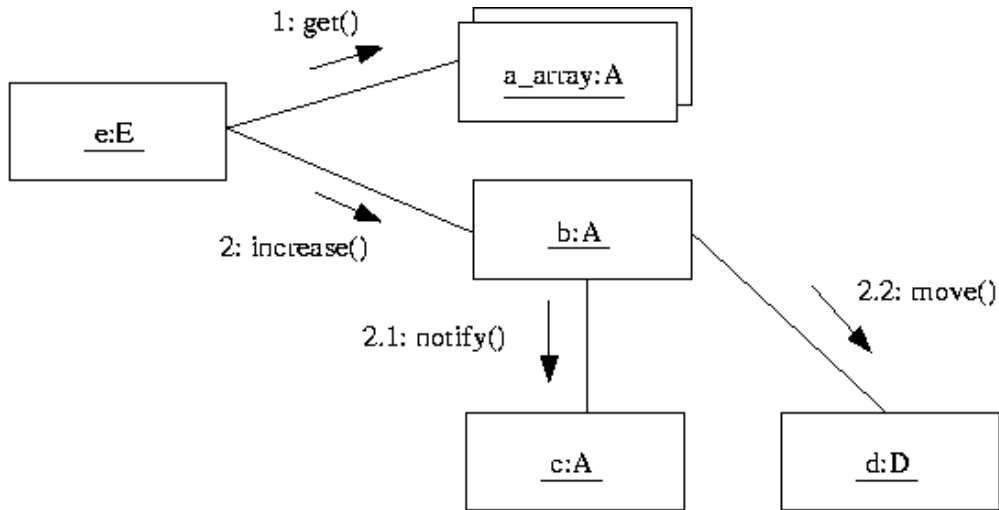
Transcendental (transzcendentális) User (felhasználói).....

Manufacturing (gyártási) Product (termék).....

Value-for-money (a pénzért kapható érték).. ..

minden helyes válasz 1 pont

3. Rajzolja meg az alábbi kollaborációs diagramban szereplő objektumok **UML osztálydiagramját**, a metódusok és azok láthatóságának feltüntetésével! A láthatóság legyen a lehető legszigorúbb, a multiplicitások esetén pedig a feltételezhető legkisebbet válassza ! (6 pont)



minden korrekt asszociáció és metódus 1 pont

4. Az UML szerint az operációk konkurenciájának szemantikája lehet: (6 pont)

sequential (szekvenciális) guarded (őrzött).....

concurrent (konkurrens)..... ..

minden helyes válasz 2 pont

5. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A bergengócok szilveszterkor gigantikus bulit rendeznek, amelyen bergengócia minden felnőtt lakója részt vesz. A szervezők (hogy senki se feledkezzen meg a nagy eseményről) mindenkinek meghívót küldenek. A buli fénypontja a petárdadobálás. Mindenki legalább egy, de ha az idő engedi több petárdát is eldob. A zsűri értékeli, hogy az egyes petárdától hány ember kapott kisebb szívinfarktust. A profi petárda-hajigálók az első petárda elhajítása előtt bejelenthetik a "Nagy Durranást". Ez azt jelenti, hogy miután a profi az összes petárdáját eldobta, kötelezően egy kisebb rakétát is fel kell lőnie, amit a zsűri szintén értékkel. Minden évben van olyan profi, aki amire sorra kerülne úgy elázik, hogy a petárdáit sem tudja eldobni. A buli végén a zsűritől mindenki díszes oklevélben kapja meg az eredményét. A helyi rozsmárklub tiszteletbeli tagja lesz az, aki a petárdáit a bergengóc tó jeges vizében úszva hajítja el. Ők a rendezőktől a vízből kimászás után azonnal egy bögre forró rumos teát kapnak.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha egy rozsmárklub tag már a harmadik petárdát dobja a vízből.

Kérésre listázza ki az elázott profikat.

Kérésre adja meg azokat, akik bejelentették a "Nagy Durranást" és teáztak.

Készítse el az

**entitás/akció leírást,
az entitás struktúra diagramot,
az initial modelt,**

kapcsolja a modellhez a

funkció processzeket.

Meghív (bergengóc) 2 pont

Petárdát dob (bergengóc, rozsmár)

Rakétát lő (bergengóc)

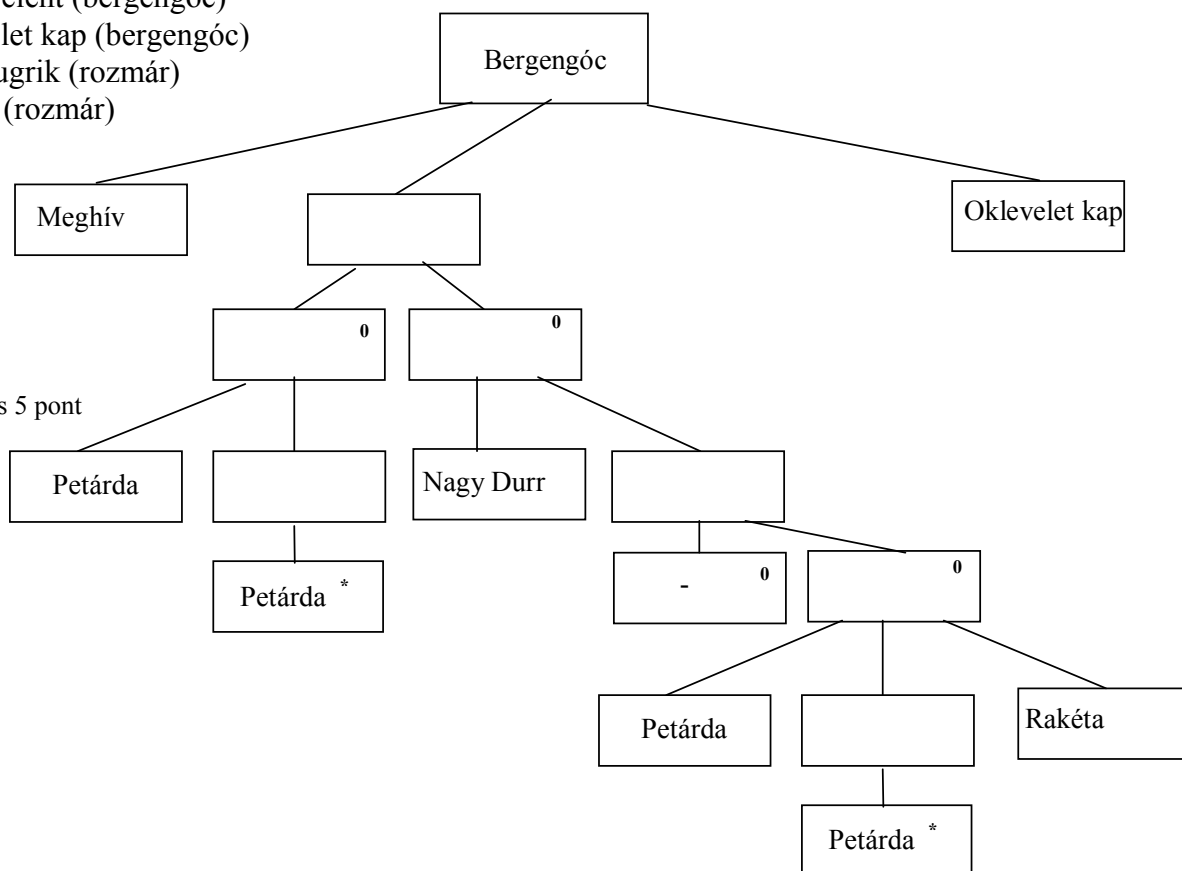
ND bejelent (bergengóc)

Oklevelet kap (bergengóc)

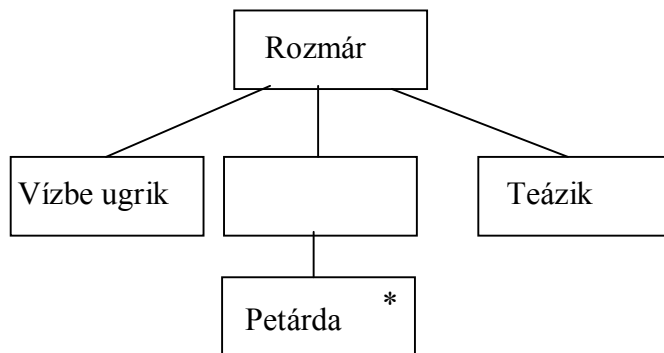
Vízbe ugrik (rozsmár)

Teázik (rozsmár)

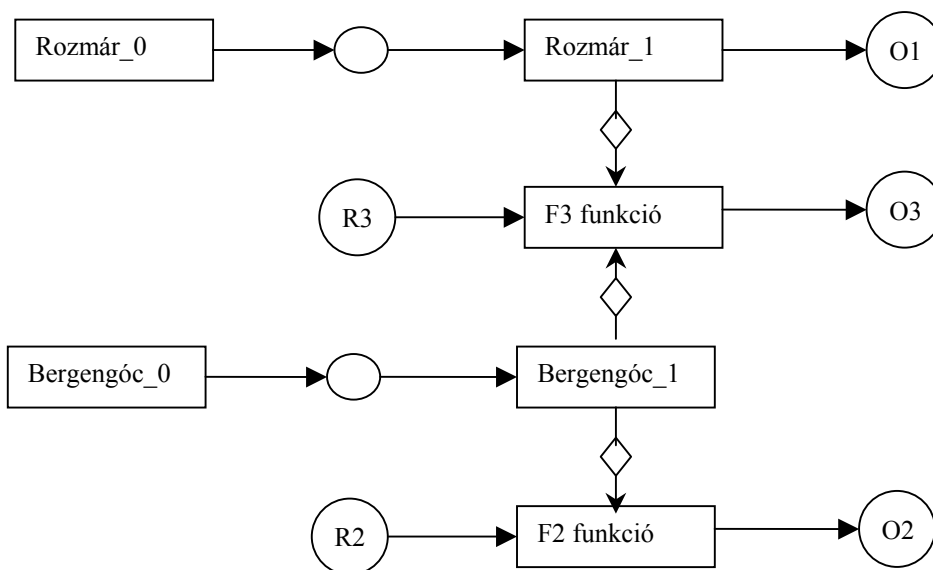
bergengóc entitás 5 pont



Rozmár entitás 2 pont

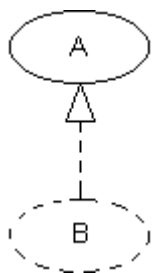


teljes SSD 2 pont



6. Mi A és B az alábbi UML diagrammon ? (2 pont)

A



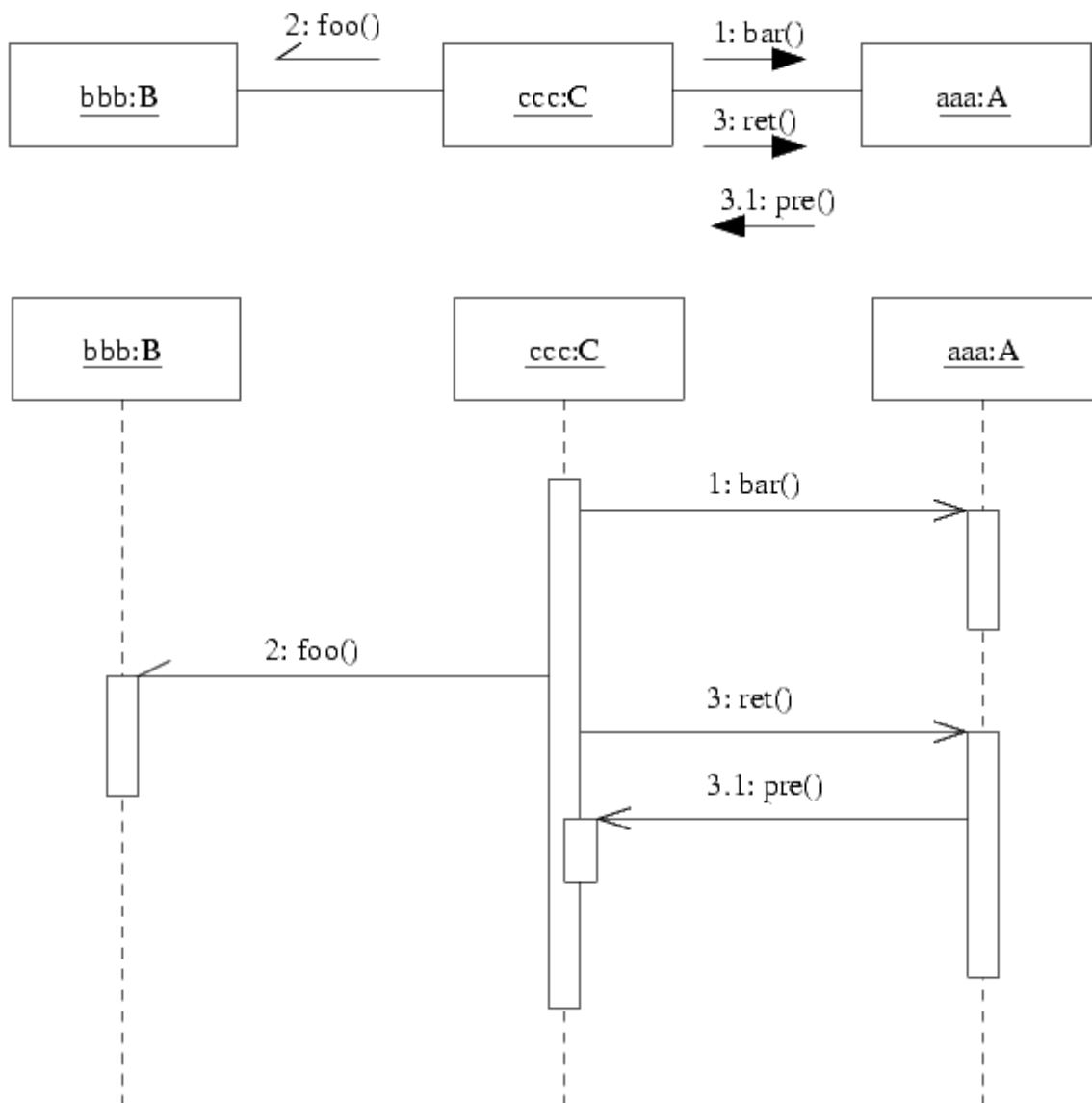
- osztály
- use-case
- kollaboráció
- komponens

B

- leszármazott osztály
- leszármazott use-case
- kollaboráció
- kollaborációs diagram

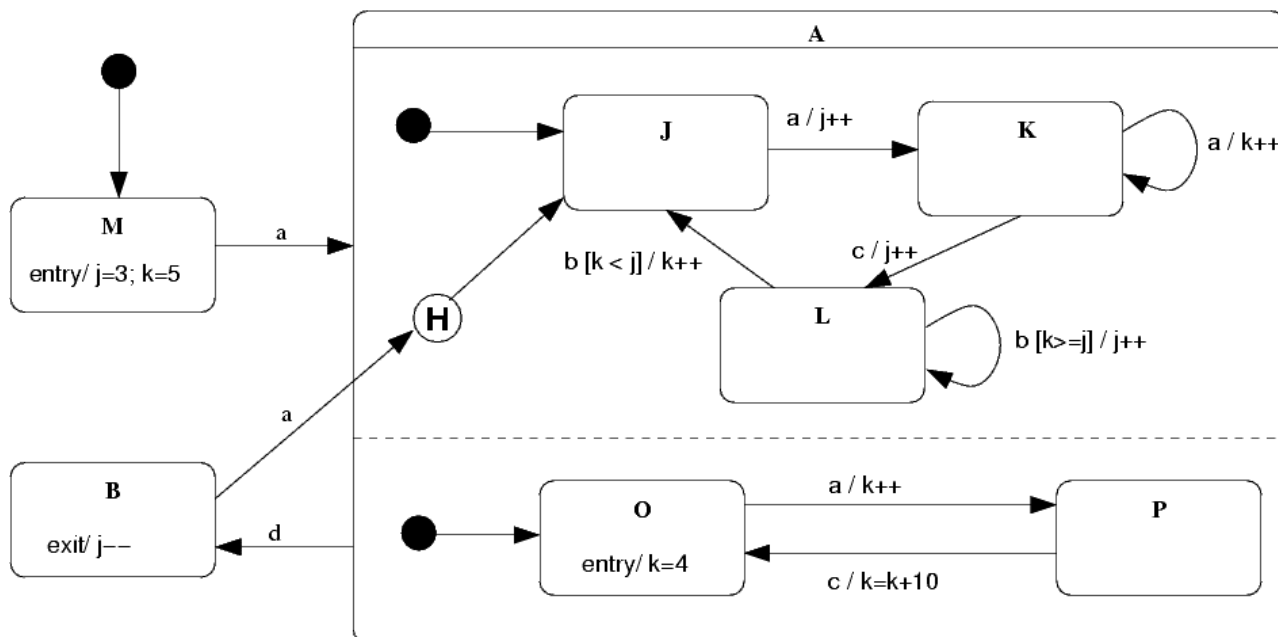
minden helyes válasz 1 pont, hiba -1 pont, negatív nem lehet

7. Az alábbi UML kollaborációs diagram alapján rajzoljon **UML szekvencia diagramot!** Ügyeljen az activation bar-ok helyes ábrázolására is! (8 pont)



minden helyes üzenet, a hozzá tartozó korrekt barral együtt 2 pont. Ha hiba van az objektummal (osztály), akkor az darabonként 1 pont levonás.

8. Az alábbi UML statechart egy objektum viselkedését írja le.



Az alábbi táblázat kitöltésével adja meg, hogy a táblázat első oszlopában álló események hatására mennyi lesz a j és k változó értéke, valamint mi lesz a következő állapot! A táblázat első oszlopa az objektumot egymás után ért eseményeket mutatja. Az első sorban szereplő esemény előtt az objektum még nem kapott eseményt. A feladat a táblázat üres helyeinek kitöltése az adott sorban szereplő esemény teljes feldolgozása után. (5 pont)

esemény	j	k	állapot
a	3	4	A(J,O)
a	4	5	A(K,P)
c	5	4	A(L,O)
a	5	5	A(L,P)
b	6	5	A(L,P)
d	6	5	B
a	5	4	A(L,O)
a	5	5	A(L,P)
b	6	5	A(L,P)
b	6	6	A(J,P)

minden helyes sor 0,5 pont

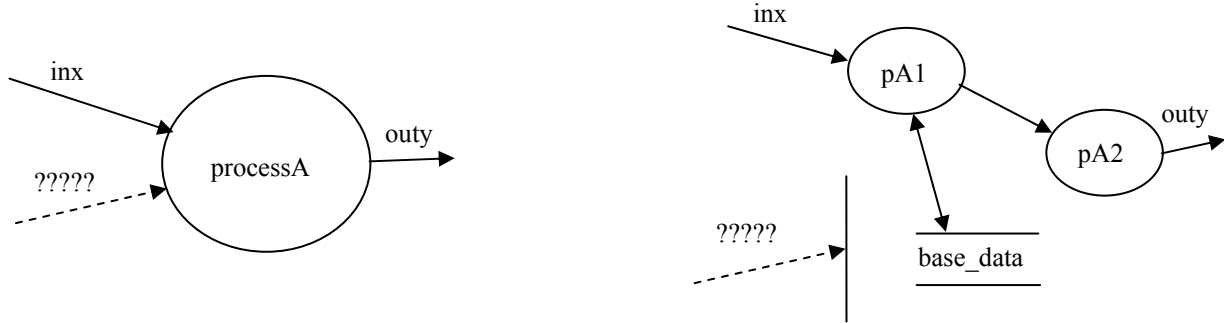
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2003. június 14.

1. A baloldalon álló adatfolyamára részletesen mi a ?????-el jelölt ábraelem ? (2 pont)
Tételezzük fel, hogy a processA folyamatot DFD-ként akarjuk megjeleníteni a jobboldali ábrarészlet szerint.
Kapcsolja a kért (????) ábraelemet korrekt módon az ábrához ! (2 pont)



Vezérlő folyam, vezérlés

2. Milyen típusú kohéziós osztályba sorolható a Pascal nyelv `write()` funkciója ? (2 pont)

- | | |
|--|--|
| <input type="checkbox"/> funkcionális | <input type="checkbox"/> kommunikációs |
| <input type="checkbox"/> szekvenciális | <input checked="" type="checkbox"/> procedurális |

helyes válasz 2 pont, minden rossz -1 pont, de ≥ 0

Nevezzen meg egy olyan kohéziós osztályt, amely a fentiek között nem szerepel ! (1 pont)

temporális vagy logikai vagy esetleges

első helyes 1 pont, minden további helyes 0.5 pont, rossz -1 pont, de total ≥ 0

3. Legyen egy az alábbi metódusokat megvalósító osztályunk ! Az azonos nevű attribútumok szemantikailag is azonosak. Mekkora lesz az osztályban a Lack of Cohesion Metric (a kohézió hiányának mértéke) ? (4pont)

```
opA(a, x, c2)
opB(b, c2, t)
opC(c1, z, g)
opD(a, x5, y)
```

	B	C	D
A	1	0	1
B		0	0
C			0

$$LCOM = |P| - |Q| = 4 - 2 = 2$$

4. Nevezze meg és jellemezze 1-1 mondattal a RUP életciklus modelljének fázisait ! (8 pont)

Inception phase: establishing the business rationale for the project, and decide on the scope of the project. Specifying the project vision.

Elaboration phase: planning the necessary activities and required resources; specifying the features and designing the architecture.

Construction phase: building and testing a working system according to the previously elaborated plans as a series of incremental iterations.

Transition phase: supplying the product to the user community (manufacturing, delivering, and training)

5. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet ! Az axiómák felírásakor megengedett a két karaktert összehasonlító művelet használata. (6 pont)

CRT()	új (üres) stringet hoz létre.
ADD(s, x)	az s string végére rakja az x karaktert.
LAST(s)	az s string végén álló karaktert adja.
END(s1, s2)	igaz, ha az s2 string az s1 string végén áll.
DUPLO(s)	igaz, ha az s stringben legalább egyszer dupla karakter fordul elő. Például: aggódik

LAST(CRT()) = undefined	0,5 pont
LAST(ADD(s, x)) = x	0,5 pont
END(s, CRT()) = true	1 pont
END(CRT(), ADD(s, x)) = false	1 pont
END(ADD(s1, x1), ADD(s2, x2)) = (x1 == x2) and END(s1, s2)	1 pont
DUPLO(CRT()) = false	1 pont
DUPLO(ADD(s,x)) = (x == LAST(s)) or DUPLO(s)	1 pont

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A Zweite Bank folyószámlát vezet az ügyfeleknek. A folyószámla megnyitásakor az ügyfél megadja a jellemző adatait. Az ügyfél a kártya átvételét követően használhatja a számlát. Készpénzt fizethet be a fiókban, kártyával vásárolhat, pénzt vehet fel ATM-ből, és átutalást hajthat végre más bankszámlákra. A bank különféle (áru, gépkocsi, lakás stb.) hitelek is ad a folyószámlás ügyfeleinek. Egy ügyfélnek egyszerre esetleg többet is. Miután az ügyfél benyújtotta hitelkérelmét, a bank dönt annak elfogadásáról vagy elvetéséről. Ha a bank a kérelmet befogadja, akkor folyósítja a hitelt (átadja a készpénzt, átutalja a hitelezett összeget az eladónak, stb.), majd megkezdődik a törlesztés. Az esedékes törlesztő részleteket a bank leemeli az ügyfél folyószámlájáról. A hitel és kamatainak megfizetése után, a hitelügylet lezárásakor a bank elszámolást küld az ügyfélnek. A folyószámlát csak akkor mondhatja fel az ügyfél, ha nincs hiteltartozása.

Definiálja az alábbi funkciókat ellátó processzeket:

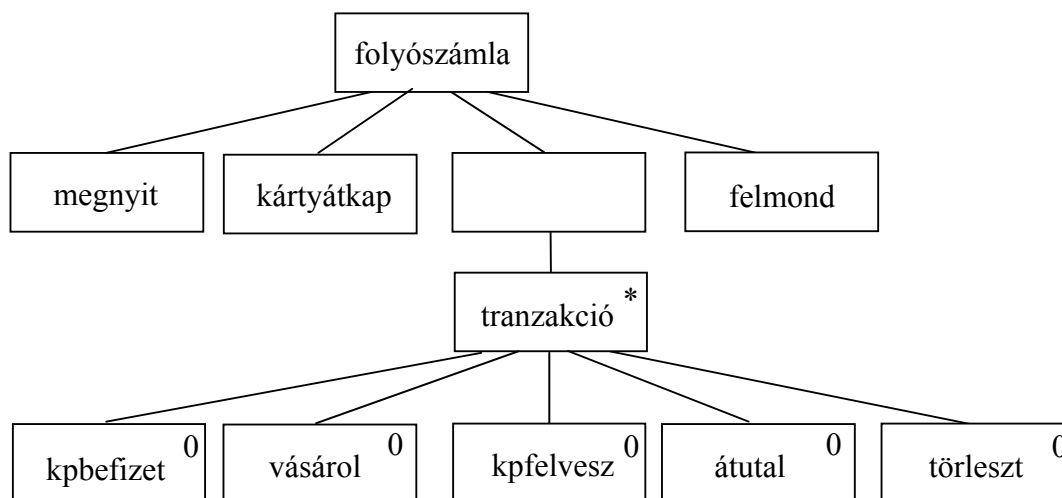
Készpénz felvételek vagy vásárláskor küldjön az ügyfélnek SMS-t !

Kérésre listázza ki azokat a hitelek, amelyeknél kevesebb, mint egy év van hátra a teljes törlesztésig !

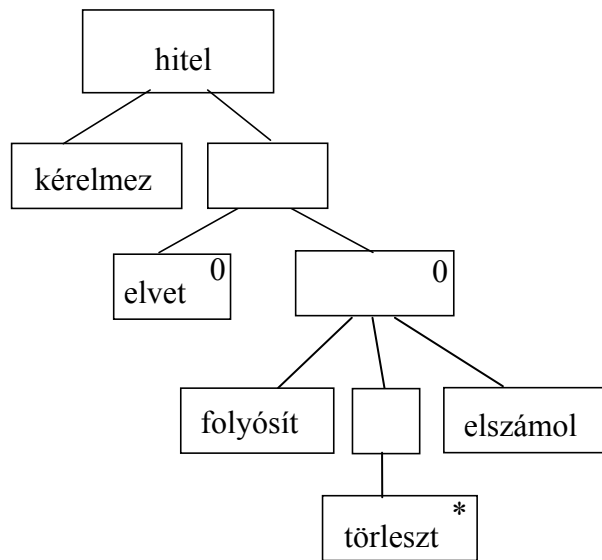
Kérésre listázza ki azokat az ügyfeleket, akiknek a folyószámláján kevesebb pénz van, mint amennyi az éppen esedékes hiteltörlesztés(ek) kifizetéséhez szükséges !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

megnyit (folyószámla) 2 pont
kártyátkap (folyószámla)
kpbefizet (folyószámla)
vásárol (folyószámla)
kpfelvesz (folyószámla)
átutal (folyószámla)
kérelmez (hitel)
elvet (hitel)
folyósít (hitel)
törleszt (folyószámla, hitel)
elszámol (hitel)
felmond (folyószámla)

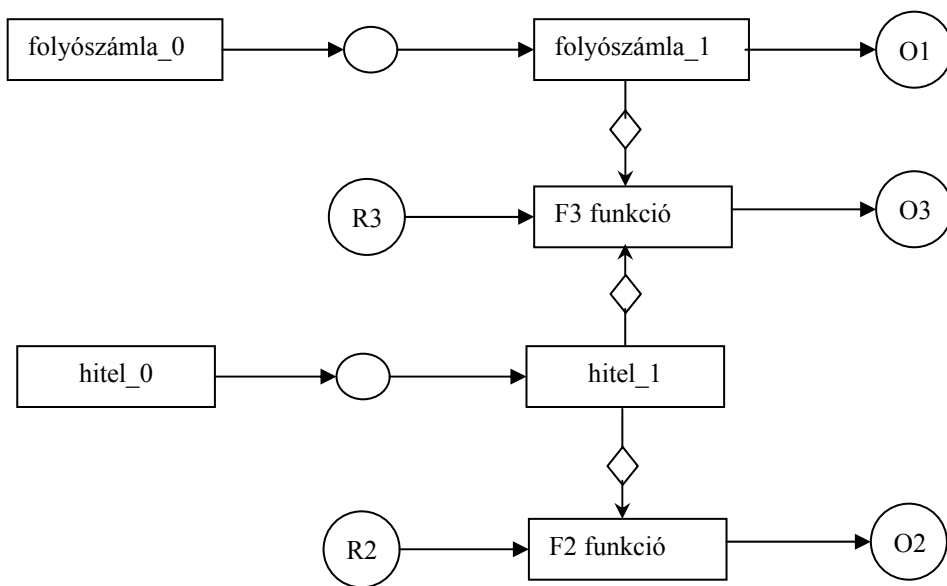


4 pont

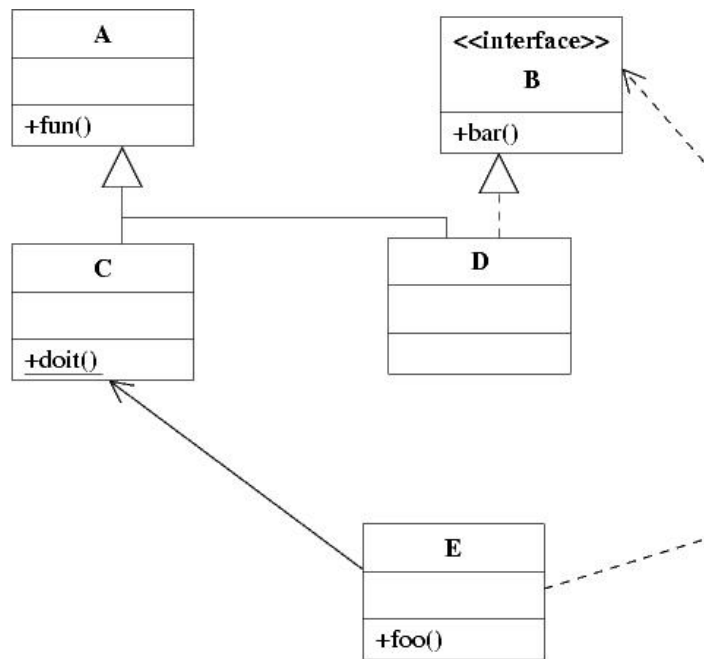


3 pont

teljes SSD 2 pont



7. Az UML osztálydiagram alapján értékeljen minden állítást az alábbi kulcs segítségével ! (7 pont)



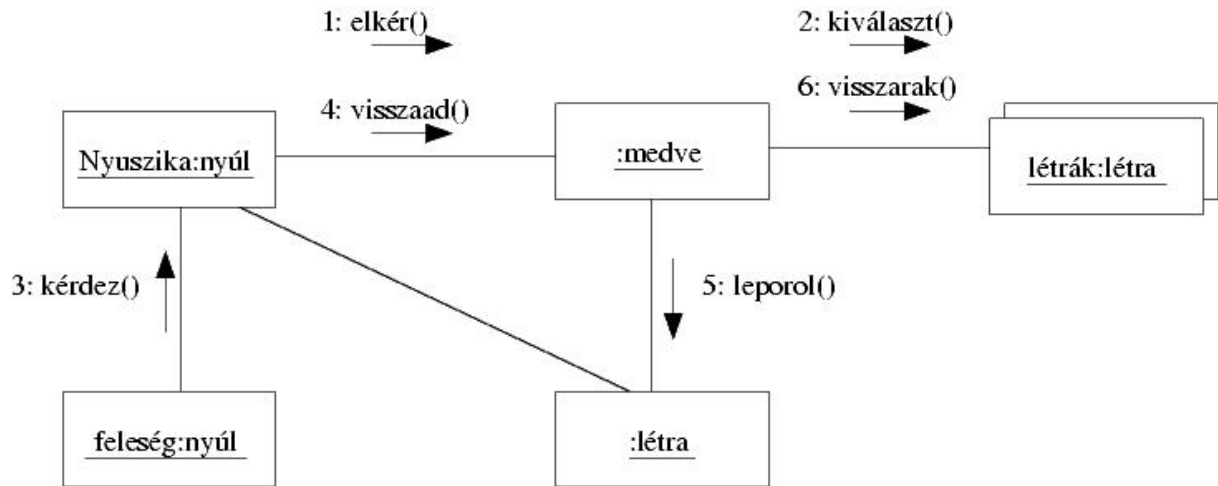
- A - csak az első tagmondat igaz
- B - csak a második tagmondat igaz
- C - mindkét tagmondat igaz, de a következtetés hamis
- D - mindkét tagmondat igaz és a következtetés is helyes
- E - egyik tagmondat sem igaz
- (F - az ki van zárva, mert nem tudom)

- [A] Egy A osztályú objektum helyettesíthető C osztályú objektummal, mert az A osztály C leszármazottja.
- [B] Egy C osztályú objektum helyettesíthető D osztályúval, mert van közös ősük.
- [E] Egy C osztályú objektum meg tudja hívni egy E osztályú objektum `foo()` metódusát, mert a metódus privát.
- [B] Egy E osztályú objektum létrehozhat egy D osztályú objektumot, mert D megvalósítja B-t.
- [D] Egy E objektum egy paraméterül kapott D osztályú objektumnak meg tudja hívni a `bar()` metódusát, mert ismeri a D által megvalósított B interfészt.
- [B] Egy E objektum nem tudja meghívni C `doit()` metódusát, mert a metódus osztály-metódus.
- [C] Egy D osztályú objektum nem ismeri E-t, mert az E osztály nem A leszármazottja.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total ≥ 0

8. Nyuszika elmegy a medvéhez, és kér tőle egy létrát. A medve kiválasztja a legszebb létráját, és odaadja nyuszikának. Mikor hazaér, Nyuszikától a felesége megkérdi, kitől van a létra. Nyuszika munkája végeztével visszaadja a létrát a medvének, és azután visszatesz a többi létra közé.

Rajzoljon a történet alapján **UML kollaborációs diagramot** ! (7 pont)



Minden elem 0,5 pont, kivéve a multiobjectet és a 2 és 6 üzeneteket, mert azok 1 pontot érnek

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2003. május 21.

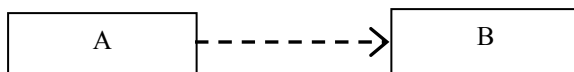
1. Mi a különbség a Funkciópont elemzésben (FPA) alkalmazott External Output (EO) és External Inquiry (EQ) között ? (4 pont)

	EO	EQ
számított eredményeket ad	igen	nem
módosítja valamely ILF-et	igen	nem

minden találat 1 pont

2. Mit jelent az, hogy A függ B-től (A depends on B) és hogyan jelöli az UML ? (3 pont)

Reláció. Ha B specifikációjának változása A megváltozását vonja maga után
jó válasz 1 pont



jó rajz 2 pont

3. A C osztály M metódusából a Demeter törvény szerint az alábbi program-elemek érhetők el: (6 pont)

az M paraméterei..... `this, super`

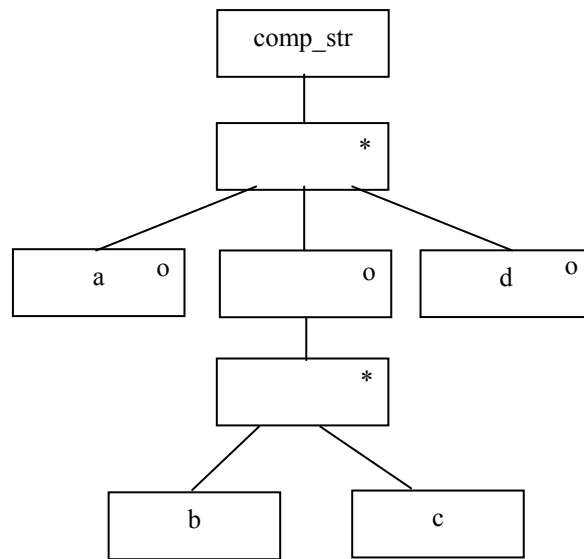
a C osztály és példány változói (attribútumok)..... globális változók.....

Az M -ben létrehozott átmeneti változók..... az M által létrehozott objektumok.....

minden helyes válasz 1 pont

4. Írja fel az alább algebrailag leírt `comp_str` nevű összetett adatstruktúrát a JSD-ben alkalmazott jelöléstechnikával ! Egy `z` attribútum egyszeri előfordulását jelölje a `z` esemény ! (4pont)

`comp_str = {[a | { b + c } | d]}`



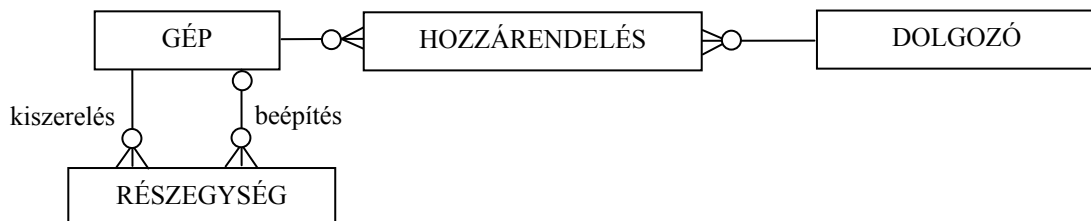
5. Az X cégnél nyilvántartják, hogy a dolgozóknál (név, adószám) milyen számítógép van. A gépet jellemzi típusa, gyári száma és ára. Aki új gépet kap, annak korábbi gépét általában más dolgozónak adják tovább, vagyis a gépek vándorolnak. A nyilvántartásból vissza kell tudni keresni, hogy egy gép mely időszakban melyik dolgozónál volt és a gépet ki installálta ott. Vannak olyan dolgozók, akiknek nincs gépük, és vannak tartalék gépek is. Bármely gépből (különböző okokból) részegységeket (tápegység, alaplap, diszk stb.) szerelhetnek ki. A részegységet jellemzi a megnevezése és száma, valamint az, hogy melyik gépben volt eredetileg. A működőképes részegységek más gépekbe beépíthetők. Egy gépből több részegységet is kiserelhetnek, egy gépben több beépített részegység is lehet. A részegységről azt is tudni kell, hogy aktuálisan melyik gépben van, mikor és ki installálta ott. Rajzoljon entitás-relációs diagramot !(8 pont)

GÉP (típus, gyári szám, ár)

DOLGOZÓ (név, adószám)

HOZZÁRENDELÉS (gép, dolgozó, időszak, installátor)

RÉSZEGYSÉG (megnevezés, szám, eredeti gép, aktuális gép, beépítési_idő, installátor)



Minden entitás és reláció 1 pont

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

Egy jelmezkölcsönzőből ruhák kölcsönözhetők. A ruhákat a kölcsönző csináltatja. Az új ruhák a bemutatóterembe kerülnek. A ruhák kölcsönzésére – foglalási díj megfizetésével – előjegyzést vesznek fel. Az előjegyzett ruhákat a kölcsönbe vétel napján a nyitáskor félreteszik és csak az előjegyzőnek adják ki. Ha az előjegyző a kért időpontig nem jelentkezik, akkor felhívják telefonon és emlékeztetik a kölcsönzésre. Olyan eset nem fordul elő, hogy az előjegyzett ruháért végül nem jönnek. Az aznapra elő nem jegyzett ruhák a bemutatóteremben vannak kiakasztva. Ezekből azonnal, előjegyzés nélkül is lehet kölcsönözni, feltéve, hogy a kölcsönzendő ruhára vonatkozó legkorábbi előjegyzés napján nyitásig visszahozza azt a kölcsönbe vevő. A visszahozott ruhát félreteszik, ha volt rá aznapra előjegyzés, egyébként pedig a bemutatóteremben kiakasztják. A tönkrement ruhákat kiselejtezik. Egy ruhát legalább egyszer kikölcsönöznek. A visszatérő, jó ügyfelek részére a kölcsönző kedvezményt szokott adni. Egy ügyfél egyidejűleg több ruhát is kölcsönözhet.

Definiálja az alábbi funkciókat ellátó processzeket:

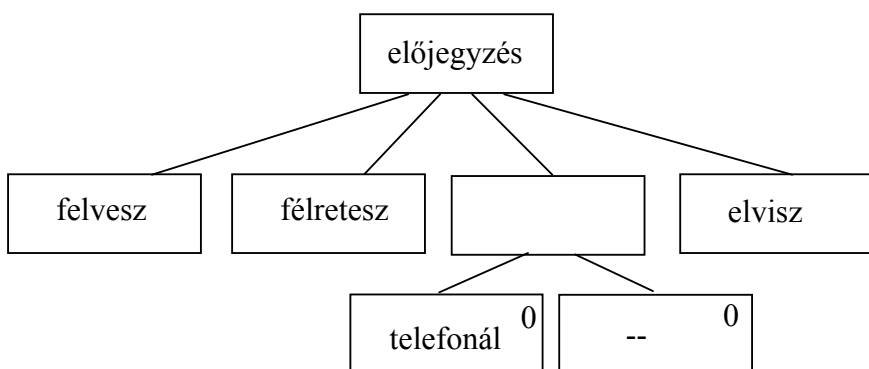
Azonnal jelezzen, ha egy visszatérő ügyfél akar előjegyezni !

Kérésre listázza ki azon ruhákat, amelyek jelenleg a bemutatóteremben vannak, de holnap reggel félre kell tenni !

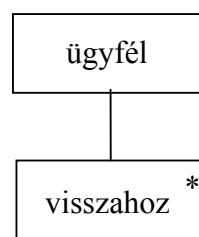
Kérésre adja meg azt a ruhát, amelyet leggyakrabban visznek el előjegyzés nélkül !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

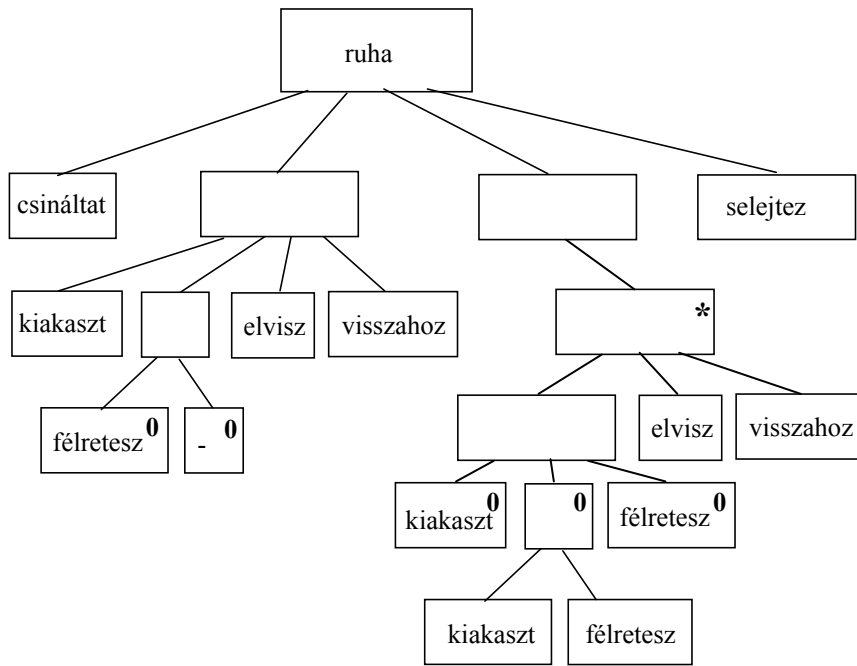
csináltat (ruha) 2 pont
felvesz (előjegyzés)
félretesz (ruha, előjegyzés)
telefonál (előjegyzés)
kiakaszt (ruha)
elvisz (ruha, előjegyzés)
visszahoz (ruha, ügyfél)
selejtez (ruha)



2 pont

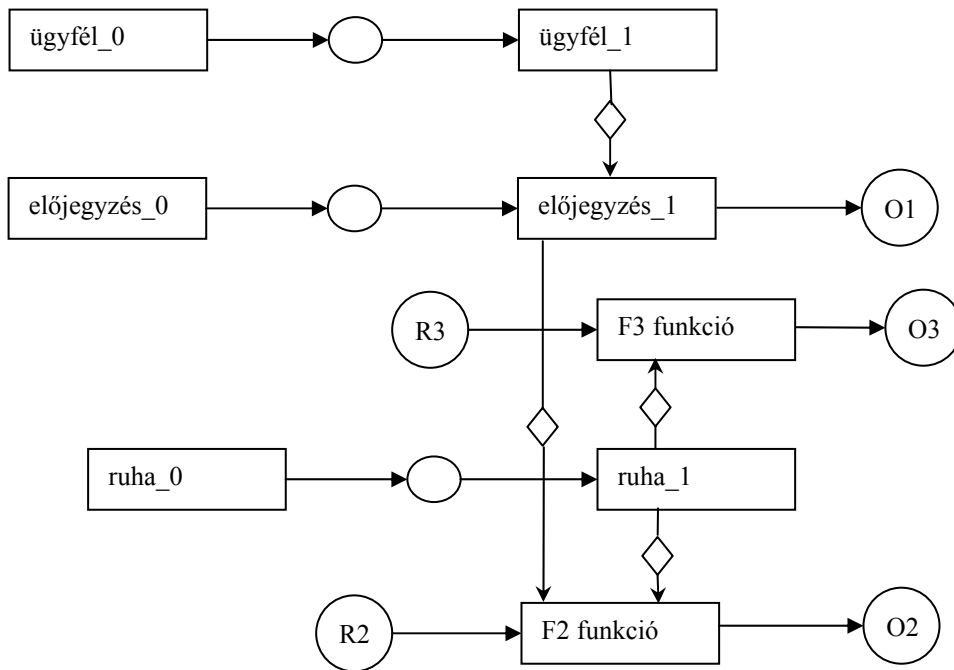


1 pont



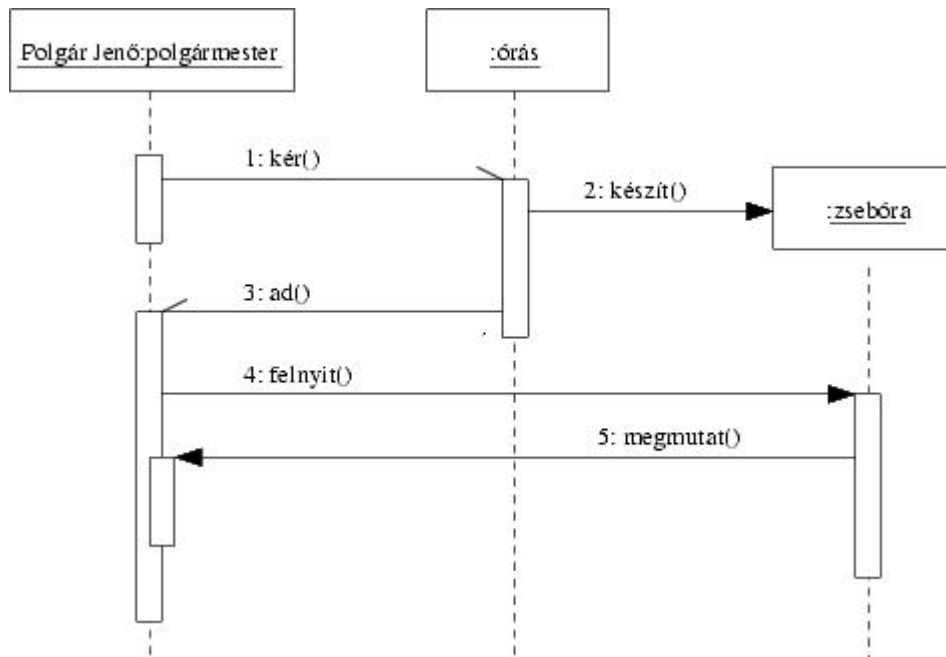
4 pont

teljes SSD 2 pont



7. Polgár Jenő polgármester üzen a város legjobb órásának, amelyben kéri, hogy az készítse neki egy zsebórát. (Ez még az órásmesternek is több napot vesz igénybe.) Mikor az óra elkészül, az órás átadja azt Polgárnak, majd dolgozóra megy. A polgármester nem vár, hanem egyből felnyitja az órát, amely megmutatja neki, hogy mennyi a pontos idő.

Készítsen a történet alapján **UML szekvencia-diagramot** ! (7 pont)



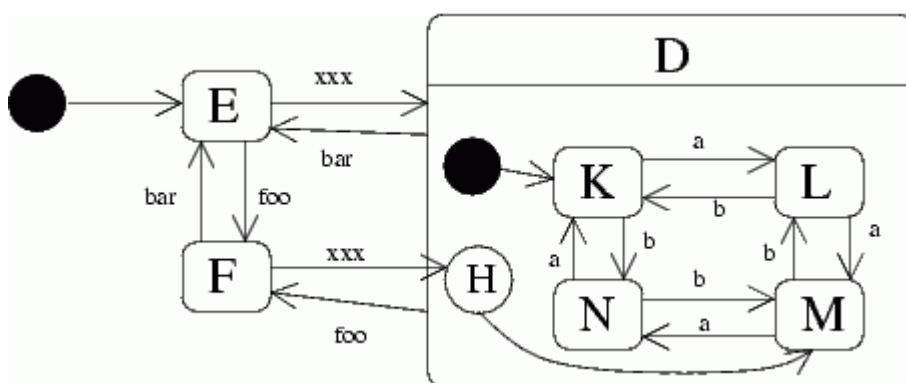
minden helyes üzenet, a hozzá tartozó korrekt bárral együtt 1 pont. 1-1 pont jár a :polgármesterért-ért és :órák-ért.

8. Egy objektum 3 fő állapottal rendelkezik: E, D és F. A kezdőállapot az E. A foo esemény hatására mind E-ből, mind D-ből F-be kerül. A bar esemény lendíti D-ből és F-ből E-be. A D-nek 4 belső állapota van: K, L, M, N. A kezdőállapot a K. A D viselkedését a következő tábla írja le:

	a	b
K	L	N
L	M	K
M	N	L
N	K	M

Ha xxx eseményt kap, és E-ben van, akkor D-be lép, és D kezdőállapotába kerül. Ha F-ben van, akkor xxx-re szintén D-be lép, de akkor a D korábbi belső állapotát veszi fel. Ha ilyen nem volt, akkor M-be jut.

Rajzolja meg a feladat **UML állapot-diagramját** ! (7 pont)



D skatulyázása korrekt induló állapottal - 1 pont

a gyűrű - 1 pont

b gyűrű - 1 pont

H és átvezetés M-re - 2 pont

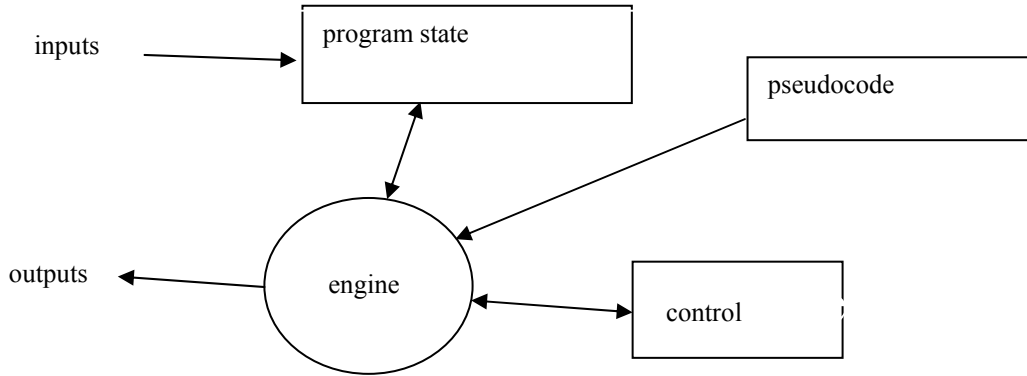
többi állapotváltás - 2 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
 c. tárgyból
 2003. május 28.

1. Rajzolja fel az interpreter szoftver architektúra mintát ! (4 pont)



minden korrekt elem 0.5 pont (4.5 pont van benne)

2. A RUP a fogalmi modell megalkotásával kapcsolatosan a “térképész-elv” követését javasolja. Mit mond ki a “térképész-elv” ? (3 pont)

a valóságban létező elnevezéseket használd !

ne foglalkozz a lényegtelen, jelentéktelen dolgokkal !

ne foglalkozz a valóságban nem létező dolgokkal !

minden helyes válasz 1 pont

3. Legyen egy `O` objektumnak egy `int calc(int x)` metódusa, ahol csak $10 < x < 60$ megengedett, és akkor a $100 < calc < 200$ igaz. Az `O` leszármazottjában felüldefiniáljuk `calc`-ot. Az alábbi állítások közül jelölje meg azokat, amelyek ha igazak a felüldefiniált `calc`-on, akkor az kielégíti a Liskov-elvet ! (6 pont)

- | | |
|--|---|
| <input type="checkbox"/> <code>..20 < x < 50</code> | <input checked="" type="checkbox"/> <code>..120 < calc < 180</code> |
| <input checked="" type="checkbox"/> <code>..00 < x < 100</code> | <input type="checkbox"/> <code>..000 < calc < 200</code> |
| <input checked="" type="checkbox"/> <code>..00 < x < 1000</code> | <input type="checkbox"/> <code>..200 < calc < 1200</code> |
| <input type="checkbox"/> <code>..30 < x < 60</code> | <input type="checkbox"/> <code>..150 < calc < 250</code> |

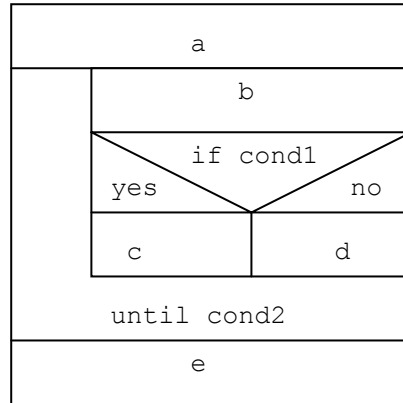
minden helyes válasz 2 pont, minden rossz -1 pont

4. Rajzolja fel az alábbi programrészlet NS (Nassi-Schneiderman) diagramját! (4pont)

```

a;
repeat
    b;
    if cond1 then c else d
until cond2;
e;

```

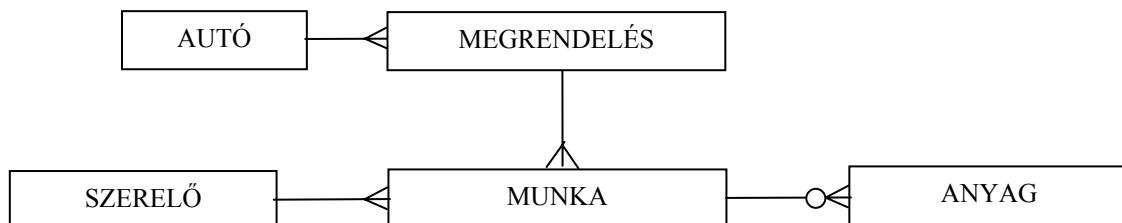


korrekt IF 2 pont
korrekt UNTIL 2 pont

5. Egy autószervezben az autót azonosítja az alvázszám, de rögzítik a rendszámot, a gyártót és a típust is. A szerelőkről tudjuk nevüket, végzettségüket, korukat és a munkaviszonyuk kezdetét. Az ügyfél megrendelése tartalmazza a kérést (pl. motor beállítás, olajcsere), a megrendelés felvételének időpontját, a várható költséget. A megrendelést annak száma azonosítja. A megrendelés teljesítése munkával történik. A munkavégzés kapcsán nyilvántartjuk az elvégzésre ténylegesen ráfordított időt és azt, hogy a munkát ki végezte. A munka közben felhasznált anyagokról tudjuk azok megnevezését és mennyiségét. Egy munkához nem mindig használunk anyagot, viszont az anyagfelhasználás mindig munkához kapcsolódik. Egy megrendelés teljesítéséhez legalább egy munkát el kell végezni. Egy autón egy munkát egy szerelő végez. Munkavégzés csak megrendelés alapján történhet. Egy megrendelés mindig egy autóhoz kapcsolódik, egy autóhoz az idők során több megrendelés tartozhat.

Rajzoljon entitás-relációs diagramot !(8 pont)

AUTÓ (rendszám, alvázszám, gyártó, típus)
SZERELŐ (név, végzettség, életkor, munkaviszony kezdete)
MEGRENDELÉS (autó, kérés, felvétel ideje, szám, várható költség)
MUNKA (megrendelés, tevékenység, tényleges idő, szerelő)
ANYAG (munka, megnevezése, mennyisége)



Minden entitás és reláció 1 pont. Pontszám = szumma pont - 1

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

Egy webes áruházban történő vásárlásnál az ügyfél először kiválasztja a termékeket. Ezt követően a fizetéshez és szállításhoz szükséges adatok megadásával feladja a rendelést. Az áruház zárja (ha tudja) a vevő bankszámláján a vásárlás összegét. Számla vagy fedezet hiánya esetén az áruház a rendelést eldobja. A rendelés alapján az áruház termékenként ajánlatkérőt küld egy raktárnak. A raktár a válaszában közli, hogy tudja-e és ha igen hány nap alatt küldeni a terméket az elosztóba. Ha a raktár által adott határidő megfelelő, akkor az áruház a terméket bekéri. Amennyiben a raktár nem tud szállítani, vagy a határidő nem megfelelő, az áruház másik raktártól kér ajánlatot az előzővel megegyező módon. Mikor az elosztóba beérkezik a rendeléshez tartozó valamennyi termék, azokat ott csomagolják és postázzák. Ha a vevő a csomagot átvette, a számlán zárolt összeget az áruház lehívja. Az áruház a jó ügyfélnek időnként extra kedvezményt ad.

Definiálja az alábbi funkciókat ellátó processzeket:

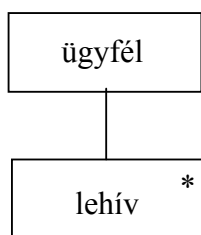
Azonnal jelezzen, ha egy jó ügyfél ad fel nagyobb összegű rendelést !

Kérésre listázza ki azokat a raktárakat, amelyek az ígért határidőnél később szállítottak !

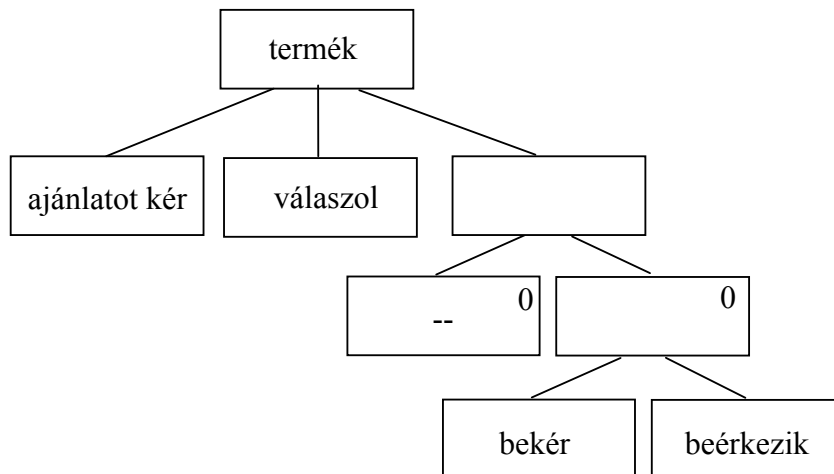
Kérésre listázza ki azon megrendeléseket, amelyeket már postáztak, de az ügyfelek még nem vettek át !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
 kapcsolja a modellhez a **funkció processzeket.**

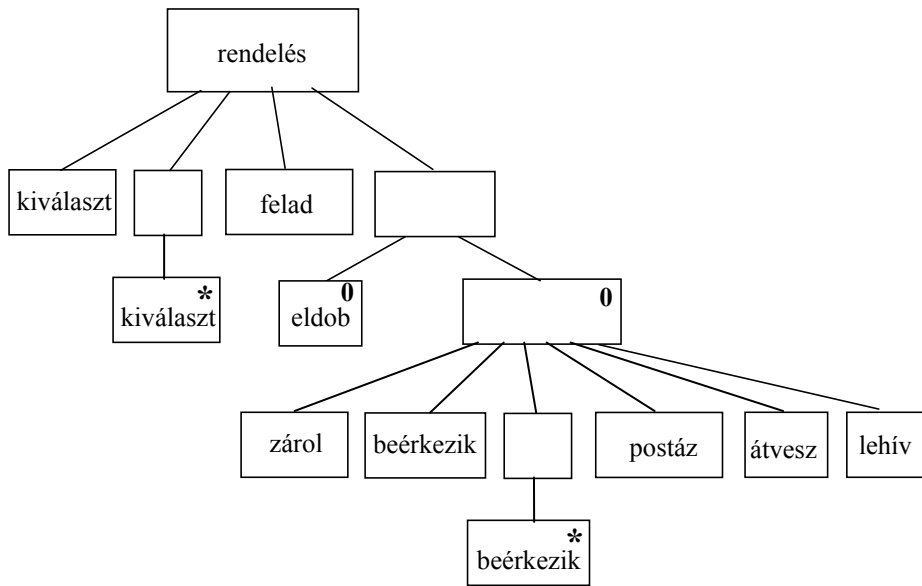
- kiválaszt (rendelés) 2 pont
- felad (rendelés)
- zárol (rendelés)
- eldob (rendelés)
- ajánlatot kér (termék)
- válaszol (termék)
- bekér (termék)
- beérkezik (termék, rendelés)
- postáz (rendelés)
- átvesz (rendelés)
- lehív (rendelés, ügyfél)



1 pont

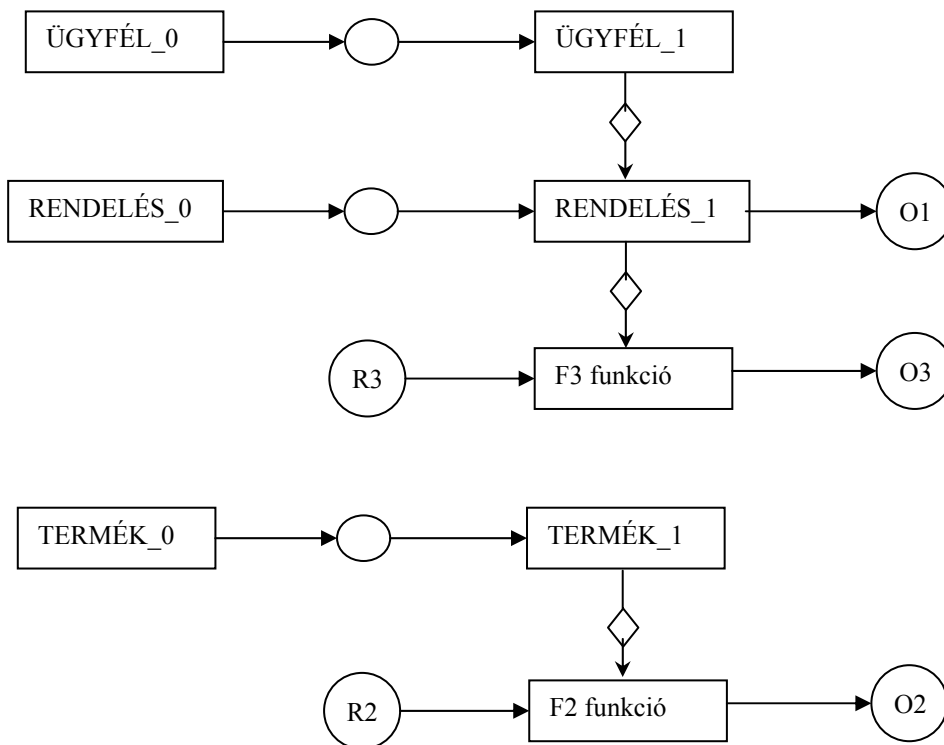


2 pont



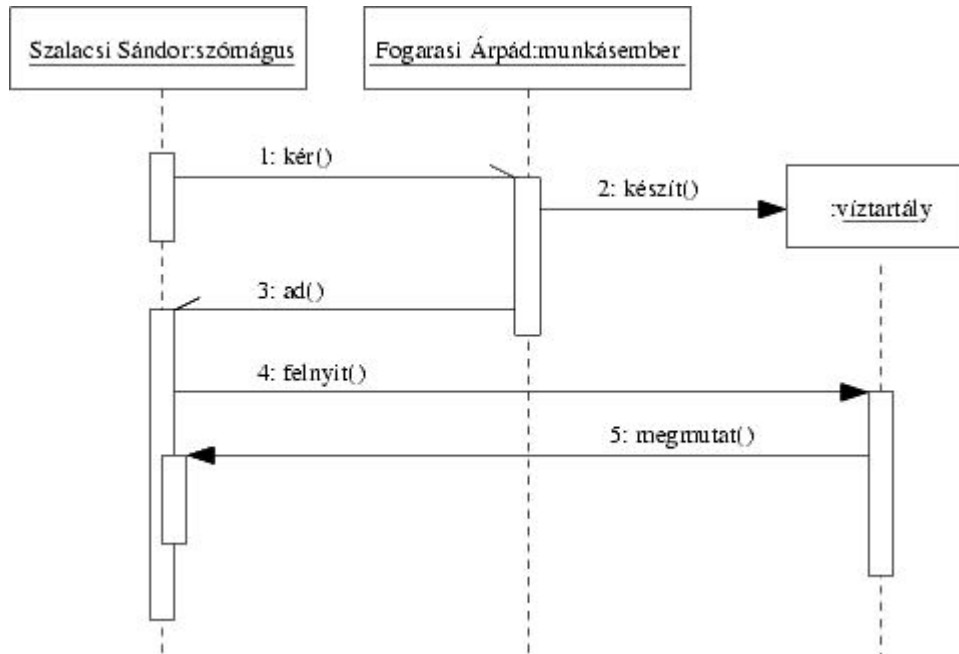
4 pont

teljes SSD 2 pont



7. Szalacsi Sándor, a szavak mágusa üzen Fogarasi Árpádnak, a jó munkásembernek, amelyben kéri, hogy az készítse neki egy víztartályt. (Ez még a jó munkásembernek is több hetet vesz igénybe.) Mikor a tartály elkészül, Árpád átadja Sándornak, majd dolgára megy. Sándor nem vár, hanem egyből felnyitja a tartályt, amely megmutatja neki, hogy mennyi víz (30000 vagy 300000 liter) fér bele.

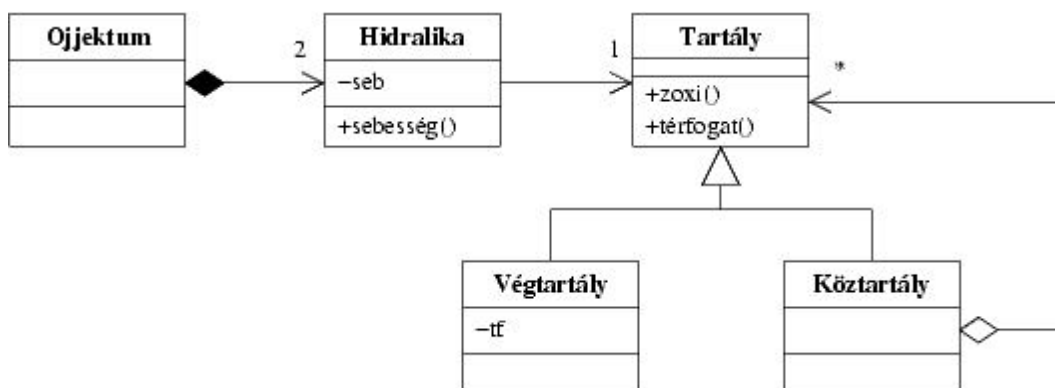
Készítsen a történet alapján **UML szekvencia-diagramot** ! (7 pont)



minden helyes üzenet, a hozzá tartozó korrekt bárral együtt 1 pont. 1-1 pont jár a :szómagusért-ért és :munkásemberért-ért.

8. Árpád objektuma két hidralikából áll. A hidralika pontosan egy tartályhoz kapcsolódik, belőle a zoxigént tudja felvenni (a tartály zoxi metódusának meghívásával). Teccik érteni? Hogy ezt milyen sebességgel tudja csinálni, az a neki (mármint a hidralikának) küldött sebesség() üzenettel állítható, amit az a privát seb attribútumában tárol. A tartály vagy végtartály, vagy köztartály. A köztartályban további tartályok lehetnek. A végtartály térfogatát (pl. 30000 vagy 300000 liter) egy privát attribútum (tf) tárolja. A tartályoktól le lehet kérdezni, hogy mekkora a térfogatuk (térfogat() metódus). A köztartály térfogata a benne levő tartályok térfogatának összege.

Rajzolja meg a fenti leírásnak megfelelő **UML osztálydiagramot** az attribútumok és a metódusok, valamint azok láthatóságának feltüntetésével! Teccik érteni? (7 pont)



Minden elem (class, kapcsolat, attribútum, metódus) 0,5 pont (7,5 pont van benne)

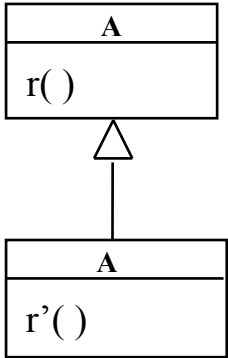
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

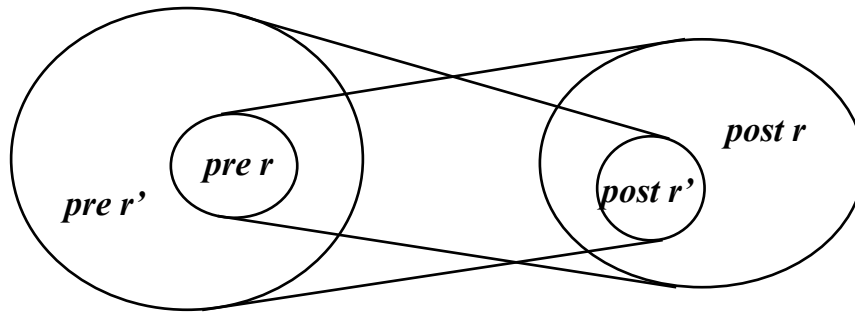
c. tárgyból
2004. január 20.

1. A és A' között a Liskov-féle helyettesítési elv szerinti öröklést definiálunk az ábra szerint. Definiálja az r és r' -re vonatkozó elő- és utófeltételek közötti összefüggést ! (2 pont).



$pre\ r'$ gyengébb vagy egyenlő mint $pre\ r$
 $post\ r'$ erősebb vagy egyenlő mint $post\ r$

Ábrázolja az összefüggéseket halmaz-ábrán is ! (2 pont)



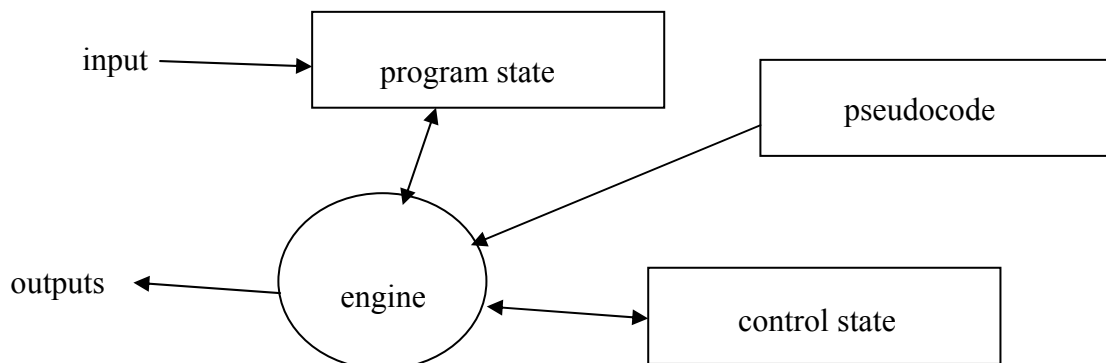
2. Adja meg a fejlesztési folyamatban előforduló tesztelési fokozatokat (stages) (6 pont)

unit (egység, modul) test integration (integrációs) test

system (rendszer) test acceptance (átadás, elfogadás) test

.....
minden helyes válasz 1,5 pont

3. Rajzolja fel az interpreter szoftver architektúra vázlatát ! (3 pont):



4. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (5 pont)

Egy könyvtárban könyveket lehet kölcsönözni. A könyvekről nyilvántartják a címét, a kiadóját, a kiadás évét valamint kategóriáját (pl.: klasszikus, krimi, gyerek, tudományos stb.). Egy könyvből több példány is lehet, de olyan könyvek adatait is tárolják, amelyekből még vagy már nincs példány. Példányonként tárolják a vásárlás idejét, az árat és az adott példány kölcsönzéseinek számát. Az ügyfeleket első kölcsönzésük alkalmával veszik nyilvántartásba. Az ügyfelet azonosító jellemzi, de nyilvántartják a nevet, e-mail címet, telefonszámot és az éves tagdíj lejárat dátumát is. Egy kölcsönzésben egy ügyfél egy könyv példányt megadott határidőig kölcsönbe vesz. A kölcsönzés megszűnik, ha az ügyfél a könyvet visszahozta. Egy időben egy ügyfél több könyvet is kölcsönözhet. Az ügyfelekről profilt készítenek, amely tartalmazza, hogy az ügyfél az élete során a különböző kategóriájú könyvekből hány alkalommal kölcsönözött (Ennek alapján lehetséges az ügyfeleknek e-mailben személyre szóló reklámokat küldeni).

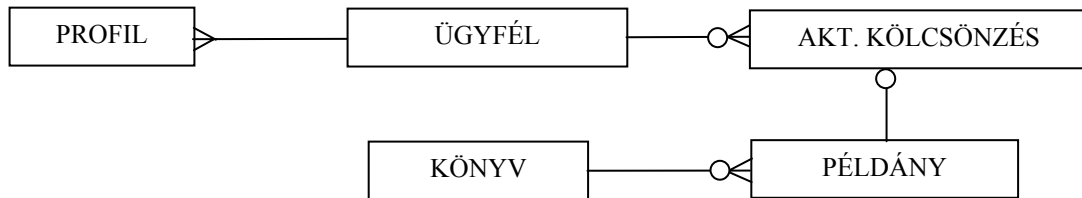
KÖNYV (cím, kiadó, kiadási év, kategória)

PÉLDÁNY (könyv, azonosító, vásárlás ideje, ára, kölcsönzések száma)

ÜGYFÉL (azonosító, név, e-mail, telefon, tagdíj lejárat)

AKT. KÖLCSÖNZÉS (példány, ügyfél, dátum, határidő)

PROFIL (ügyfél, kategória, kérések száma)



mindegyik entitás és mindegyik reláció 0.5 pontot ér 4,5 pontot 5-re kerekít.

5. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

gyakorlati problémák megoldására szolgáló **dolgok**

a **társadalom** által szabályozott **tudományos** alapokon álló

gazdaságos

előállításának mikéntje

mindegyik komponens 1 pont.

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A HP (Henrik-Plotter) Rt. végzi a zsupszkulcsok javítását, karbantartását. A zsupszkulcsot használó cégek együttműködési szerződést kötnek a HP-vel. Ezt követően ha bármi probléma adódik, a cég képviselője telefonál a HP gyorsszolgálatának. A szerelő azonnal kivonul a helyszínre és a hiba természetétől függően a hibás zsupszkulcsot megjavítja, vagy kicseréli. Az elvégzett munkának megfelelő javítási költséget a cégnek kiszámlázzák. Ha a cég a számlán megadott fizetési határidőig nem fizet, akkor a HP inkasszóval leemeli a pénzt a cég bankszámlájáról. Ha egy céggel kapcsolatosan háromszor előfordul, hogy nem fizet, akkor a HP az együttműködést felmondja. A HP a cégekkel az együttműködést évente megújítja, de ez nem vonatkozik azokra, akikkel az együttműködést a HP felmondta. Velük a továbbiakban nem tartanak fenn üzleti kapcsolatot.

Definiálja az alábbi funkciókat ellátó processzeket:

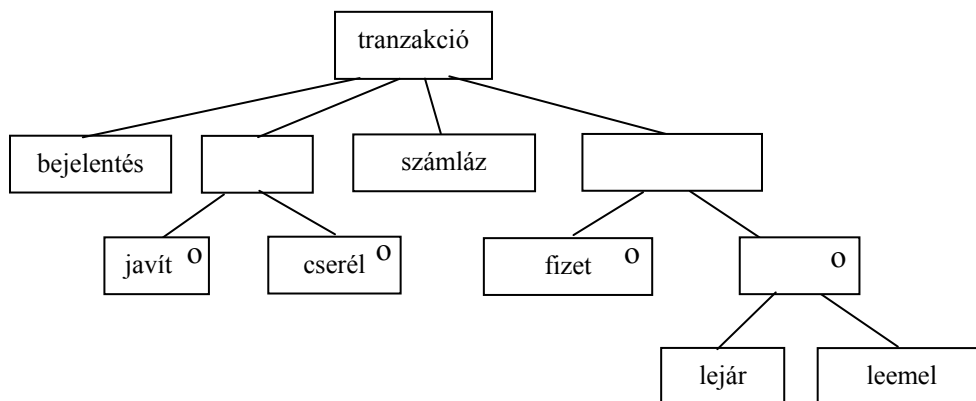
Automatikusan generáljon felmondást, amikor harmadik alkalommal kell leemelni a cégtől a pénzt !

Listázza ki azokat a kivonulásokat, ahol cserére volt szükség és még legalább 3 nap van hátra a fizetési határidő lejártáig !

Azonnal jelezzen, ha olyan ügyfél telefonál, akivel a HP az együttműködést felmondta !

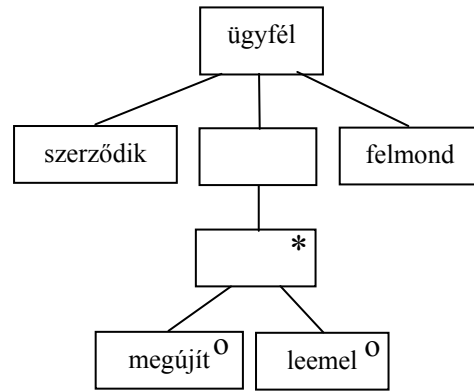
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

bejelent (tranzakció) 2 pont
javít (tranzakció)
cserél (tranzakció)
számláz (tranzakció)
fizet (tranzakció)
lejár (tranzakció) - **nem fizet –NEM JÓ**
leemel (tranzakció, ügyfél)
szerződik (ügyfél)
megújít (ügyfél)
felmond (ügyfél)

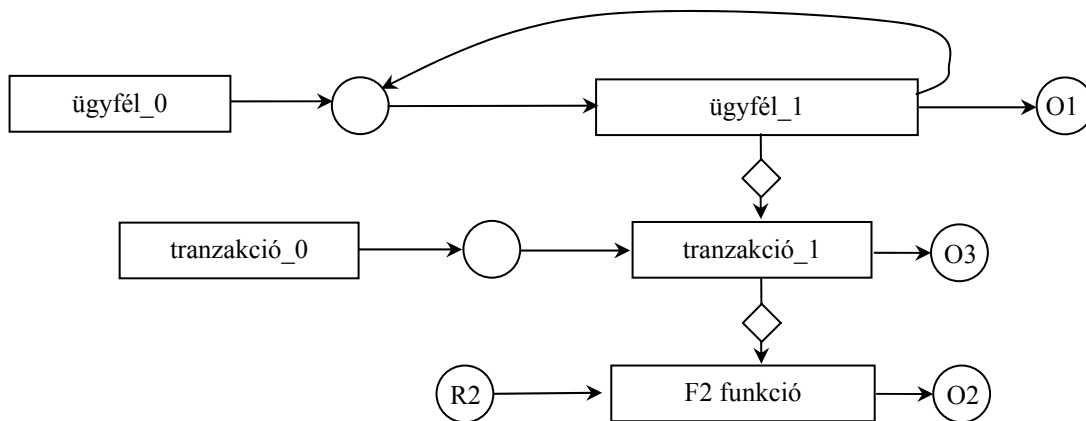


4 pont

3 pont

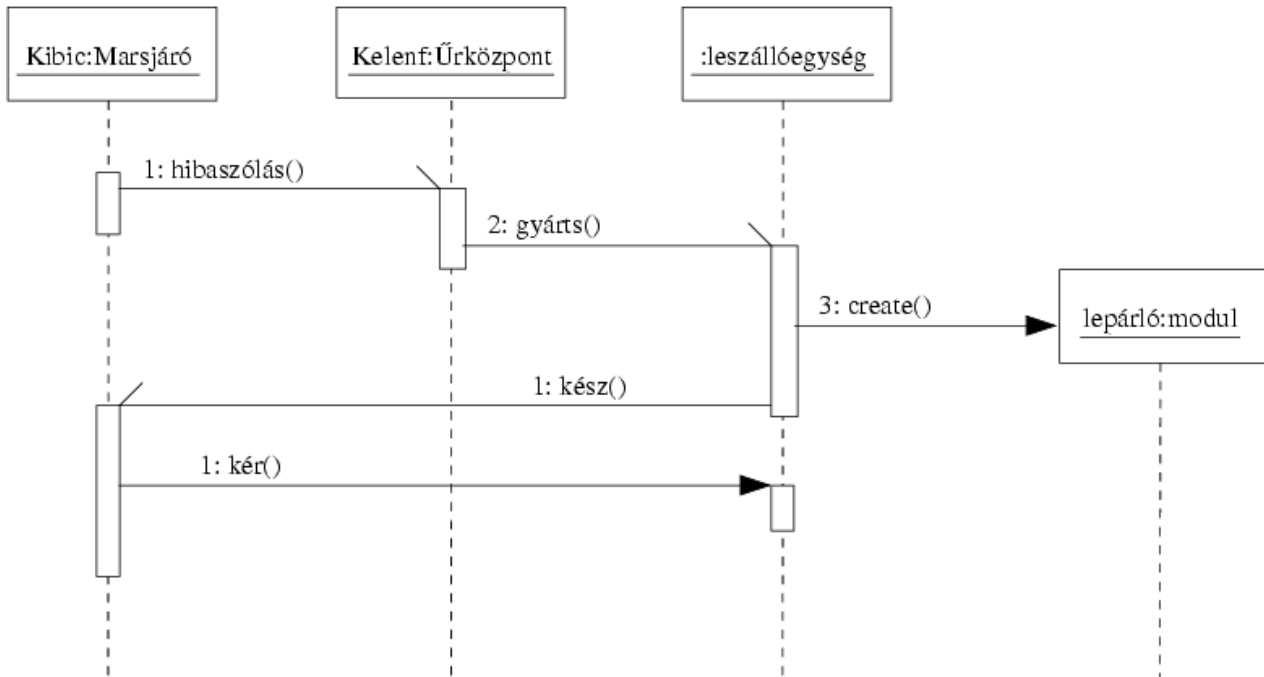


2 pont

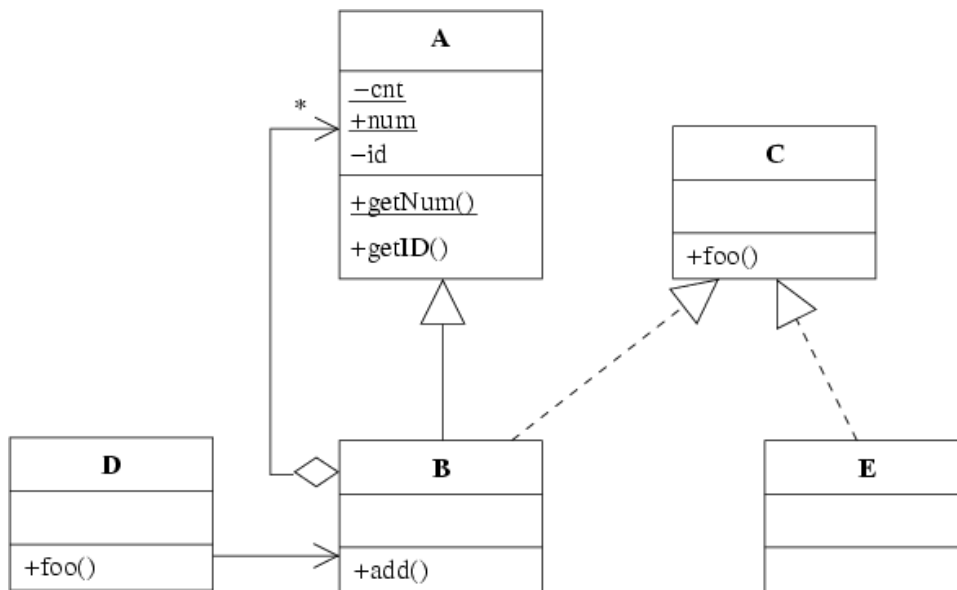


7. Kibicnek, a magyar marsjárónak elromlik a pálinkalepárló modulja, erről üzenetet küld a kelenföldi űrközpontba, majd folytatja tevékenységét (vörös kőzet gyűjtése a lágymányosi teniszpályák talajához) a Marson. Az üzenet feldolgozása után az űrközpont egyből utasítást ad a leszállóegységnek, hogy készítsen új pálinkalepárló modult. Ennek elkészültekor az egység szól Kibicnek, aki odagurul, és elkéri az új modult.

A fenti történet alapján rajzoljon **UML szekvencia diagramot** ! (9 pont)



8. Az UML diagram alapján értékeljen minden állítást az alábbi kulcs segítségével ! (7 pont)



- A - csak az első tagmondat igaz
- B - csak a második tagmondat igaz
- C - mindkét tagmondat igaz, de a következtetés hamis
- D - mindkét tagmondat igaz és a következtetés is helyes
- E - egyik tagmondat sem igaz
- (F - az ki van zárva, mert nem tudom)

- [B] egy D osztályú d objektum nem mindig hívhatja meg egy B osztályú b objektum `getNum()` metódusát, mert előfordulhat, hogy b nem aggregál egy A osztályú objektumot sem
- [B] B nem örökli A `num` attribútumát, mert az A.`num` statikus
- [B] B osztályú objektum helyén bárhol használhatok E osztályút, mert mindkettő a C interfészt valósítja meg
- [D] B osztályú objektum nem látja D osztályú objektum `foo()` metódusát, mert a navigáció iránya D-től B-re mutat
- [C] E nem látja B `add()` metódusát, mert E-nek nem őse A
- [A] B nem látja az aggregált A-k `id` attribútumát, mert az `id` statikus
- [B] D nem hívhatja B `foo()` metódusát, mert a C interfészt nem ismeri

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total ≥ 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2004. január 6.

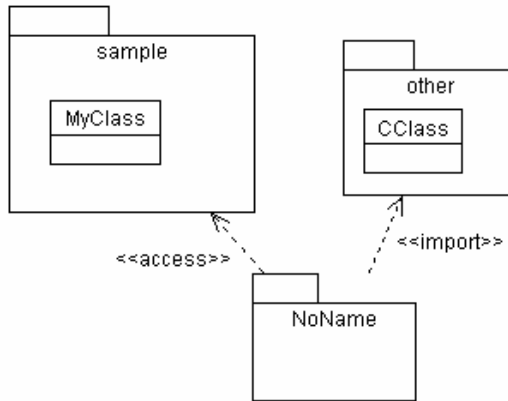
1. Milyenek a mellékelt UML ábrán szereplő relációk (1-1 pont) ?

<<access>>

- öröklés (inheritance)
- tartalmazás (aggregation)
- függőség (dependency)
- példányosítás (instantiation)

<<import>>

- öröklés (inheritance)
- asszociáció (association)
- függőség (dependency)
- minősítés (qualification)



Mi a két reláció jelentése? (1-1 pont)

<<access>> NoName-ből teljes útvonallal látjuk sample elemeit

<<import>> NoName címtérébe bekerül other tartalma

2. Az Aspektus Oriénált Programozásban (6 pont)

mi biztosítja a genericitást: aspektusok szövése

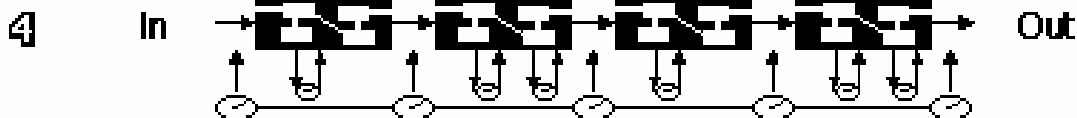
mik az interfészek: a csatlakozó pontok (join points) a szövéshez

hogyan adaptálunk: ragasztó kódok és csomagolók szövése, aspektusok változtatása

3. Mi a CMM negyedik szintjének neve (1 pont): Managed (menedzselt, irányított)

Jellemezze röviden (3 pont):

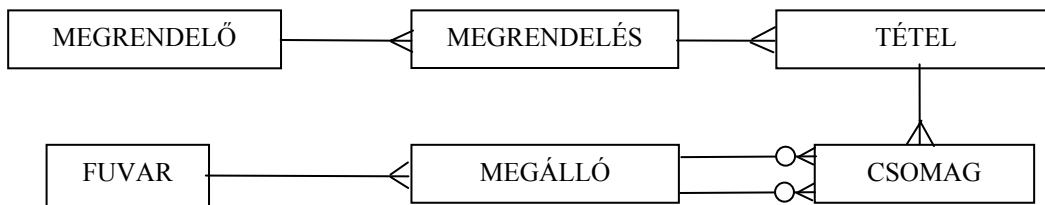
Both the software process and products are quantitatively understood and controlled.



4. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A SPEdit Kft. szállítmányozást végez a saját teherautóival. A beérkező megrendelés tartalmazza, hogy ki a megrendelő, honnan, hová, milyen határidőn belül kell szállítani. A megrendelőről nyilvántartják annak nevét, címét, adószámát, valamint az általa korábban összesen adott megbízások értékét. Egy megrendelés legalább egy, de inkább több tételt tartalmaz. A tételnek van megnevezése, darabszáma, és darabsúlya. Egy tételt egy vagy több csomagba csomagolnak a szállíthatóság miatt. A csomagot jellemzi egy egyedi sorszám és a csomag súlya. Egy gyűjtő fuvar egy adott napon egy teherautó a vezetőjével együtt teljesít. A fuvar a Kft. telephelyén üres raktérrel kezdődik (az induló km állás rögzítésével) és ott is végződik, de több megállóból áll. Minden megálló valamely szállítási feladathoz kapcsolódik, ahol a teherautóról csomagokat raknak fel és/vagy le, jellemzője az autó kmórájának állása. Egy csomag több megállón keresztül is a raktérben maradhat, ez az útvonaltól függ.

- MEGRENDELÉS(*megrendelő, honnan, hová, határidő*)
- MEGRENDELŐ(*név, cím, adószám, érték*)
- TÉTEL(*megnevezés, darabszám, darabsúly, megrendelés*)
- CSOMAG(*sorszám, súly, tétel, felvesz_megálló, letesz_megálló*)
- FUVAR(*autó, vezető, induló km, dátum*)
- MEGÁLLÓ(*hely, óraállás, fuvar*)



mindegyik entitás és mindegyik reláció 0.5 pontot ér.

5. Milyen minőségi jellemzőket tesztlünk a különböző (FURPS) típusú tesztekkel ? (4 pont)

- Functionality, funkcionalitás..... Usability, használhatóság
- Reliability, megbízhatóság..... Performance, teljesítőképesség
- Supportability, támogatottság

minden helyes válasz 1 pont, van benne 1 plusz pont is

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A SíAkadémia sítanfolyamokat szervez. Az ügyfél a tanfolyam keretében a jelentkezéskor befizetett díj fejében meghatározott számú – az ügyfél által választott napokon történő – oktatáson vehet részt. Minden nap reggel a SíAkadémia vezető edzői az aznapon megjelent ügyfelekből a sítudásuk szerint csoportokat alakítanak ki. A csoportokba sorolt ügyfelekkel az edzők az oktatási feladatnak, a hó- és időjárási viszonyoknak megfelelő pályára mennek és ott folyik az oktatás. A nap végén a csoporton belül az edző háziversenyt is rendezhet, ha ehhez a csoport tudása elegendő és az időjárás is megengedi. Az edző a foglalkozás végén minden hozzá besorolt ügyfél tudását minősíti. Ennek figyelembe vételével történik majd a következő alkalommal a besorolás (az első alkalommal az ügyfél elmondása szerint sorolnak). Az ügyfél a tanfolyam végén – vagy bármelyik befejezett oktatási napot követően, a tanfolyamot a maga részéről befejezve – kérheti bizonyítvány kiállítását. Előfordulhat, hogy az ügyfél balesetet szenved oktatás közben. Ekkor az edző nem minősíti az ügyfelet, hanem az ügyfél jelenti a balesetet és visszakapja a tanfolyami díj arányos részét. Ez esetben az ügyfél nem kap bizonyítványt.

Definiálja az alábbi funkciókat ellátó processzeket:

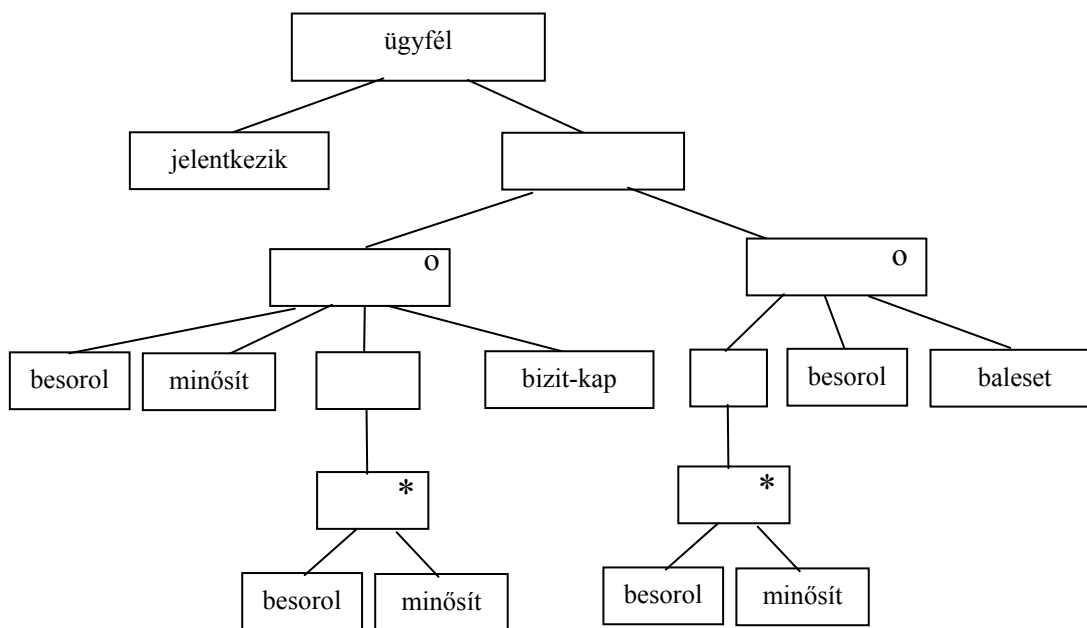
Azonnal jelezzen, ha baleset történt !

Kérésre listázza ki azokat az edzőket, akik a kezdőket gyakran viszik nehéz pályákra !

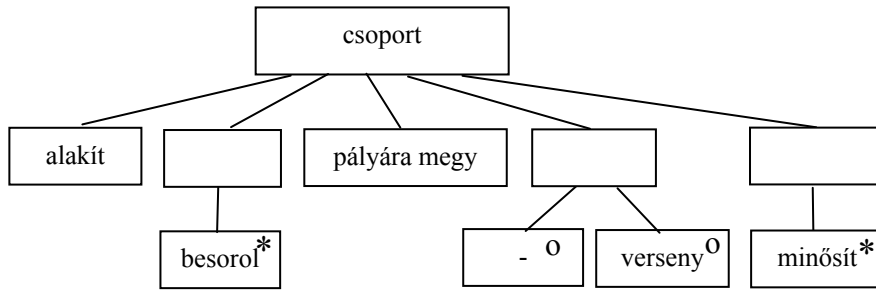
Amikor tanfolyam végén bizonyítványt állít ki, készítsen listát azokról az edzőkről, akik az ügyfelet tanították és versenyt is rendeztek !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
 kapcsolja a modellhez a **funkció processzeket.**

- jelentkezik (ügyfél) 2 pont
- alakít (csoport)
- besorol (ügyfél, csoport)
- pályára megy (csoport)
- verseny (csoport)
- minősít (ügyfél, csoport)
- bizit-kap (ügyfél)
- baleset, pénz vissza (ügyfél)

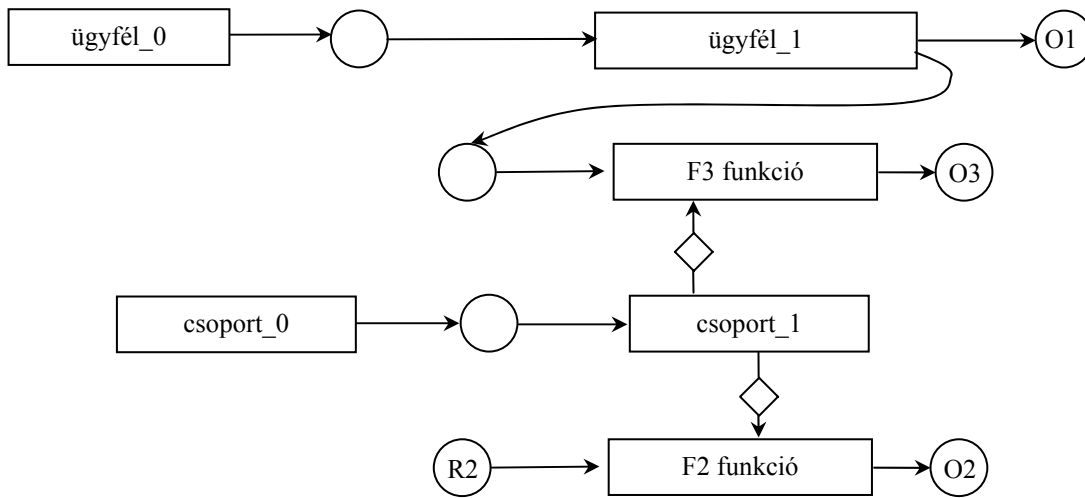


4 pont



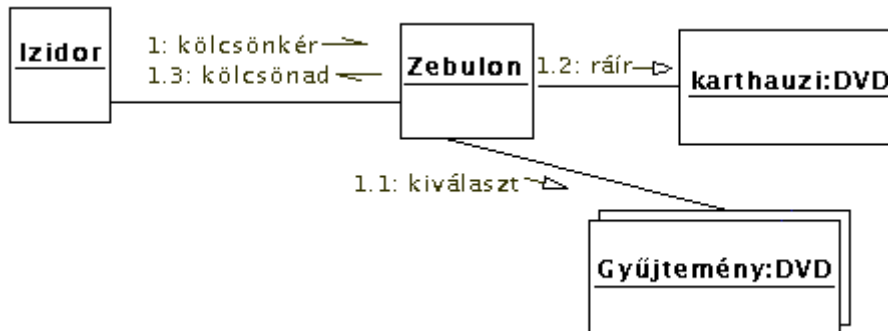
3 pont

F3 1 pont, összes többi 1 pont



7. Izidor az iskolában kölcsönkéri Zebulon egyik DVD-jét, amin karthauzi szerzetesek olvassák fel a rend reguláját. Otthon Zebulon kiválasztja gyűjteményéből a kért DVD-t, amire ráírja, hogy az övé, majd másnap odaadja Izidornak.

Rajzoljon UML kollaborációs diagramot a fenti történetre ! (7 pont)



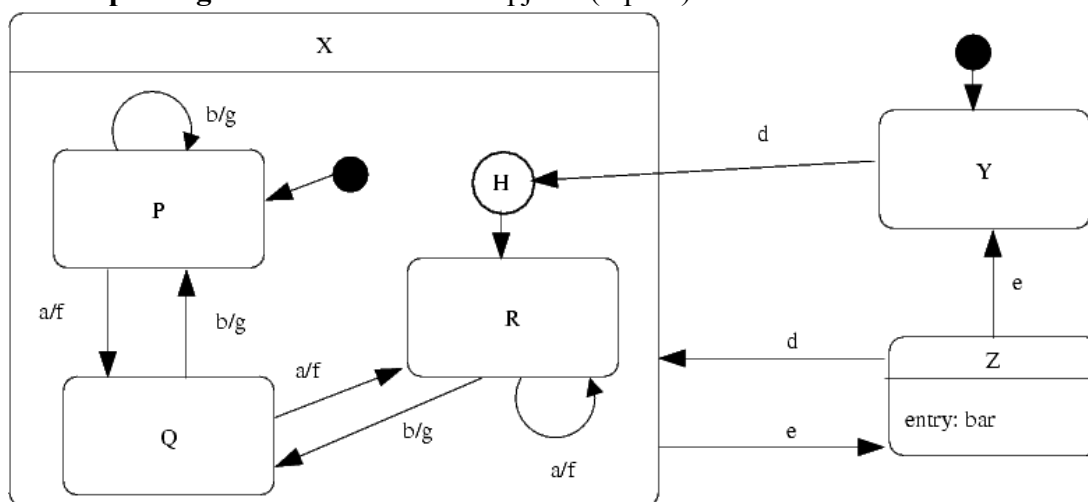
- Multiobject 1 pont
- Kölcsönkér aszinkron 1 pont
- Kölcsönad aszinkron 1 pont
- Kiválaszt 1 pont
- Masik 3 objetum 1-1 pont

8. Az Antipeca WC-tisztító robot szenzorai a beérkező jeleket a, b, d, e eseményként továbbítják a robot agyába. Ez utóbbiban egy állapotgép működik, melynek X, Y és Z főállapotai vannak. Az X főállapot belső működését a következő állapottábla írja le:

	a	b
P	Q/f	P/g
Q	R/f	P/g
R	R/f	Q/g

X-ben a kezdőállapot a P. Ha X-ben az e esemény történik, akkor a Z állapotba jutunk. Az Y állapotból d esemény bekövetkeztére az X legutóbbi állapotába jutunk. Ha még nem jártunk X-ben, akkor R-be. Z-ből d eseményre X-be, e eseményre Y-ba jutunk. Z-be való belépéskor mindig lefut a bar akció. A főállapotok közül a kezdőállapot az Y.

Rajzoljon UML állapotdiagramot a fenti leírás alapján ! (8 pont)



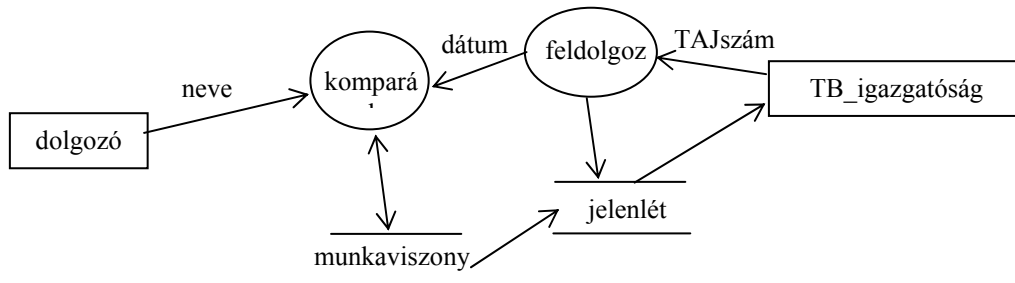
- X skatulyázás 1 pont
- Z entry 1 pont
- H korrekt (d és R kapcsolat) 1-1 pont
- Z és d 1 pont
- Z és e 1 pont
- 2 korrekt kezdőpont 1-1 pont
- X belseje rossz, 1 pont levonás

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból
2004. június 12.

1. Milyen szintaktikai hibákat talál az alábbi adatfolyam ábrán ? (4 pont)



- ha ez DFD, akkor nincs terminátor, ha ez ContextDiag, akkor nincs 2 processz és adattár
- **komparál**ból nincs kimenő adat, pedig a **munkaviszony**ból **komparál**ba mutató nyílhegy miatt kellene
- **jelenlét** (adattár) nem köthető a **TB_igazgatóság**hoz (terminátor)
- **munkaviszony** és **jelenlét** (két adattár) közvetlenül nem köthető össze

minden felismert hiba 1 pont, minden tévedés -0,5 pont

2. Melyek a fő típusai a kohézív osztályokat összefogó **domain**-eknek ? (6 pont)

Foundation Abstract/Math Unit Struct.....

Business Role Relationship

Architectural HCI DB Com

Application

minden helyes válasz 1,5 pont, minden tévedés -0,5 pont

3. Mit jelentenek az alábbi fogalmak ? (5 pont)

At-most-once: if an operation request returns successfully, it was performed exactly once; if it returns an exception indication, it was performed at-most-once. A client is able to invoke an at-most-once operation in a synchronous or deferred-synchronous manner

Best-effort: is a request-only operation (i.e., it cannot return any results and the requester never synchronizes with the completion, if any, of the request)

Minek a végrehajtásával kapcsolatosan definiáltuk őket: **Operation**.....

Fogalommagyarázat 2-2 pontot ér, az Operation (NEM METHOD) 1 pont

4. Készítsen **entitás-relációs modellt** az alábbi problémára! (6 pont)

A MegvAir légitársaság repülőjáratokat üzemeltet. A járat jellemzője a járatszám, az induló és célállomás. Egy adott napon a konkrét járatra vonatkozóan nyilvántartjuk az utasok számát és a késést. Egy napon nem indul többször ugyanazon számú járat. A járathoz a repülőt az aktuális utaslétszámot figyelembe véve rendelik hozzá. A repülőket leírja a lajstromjel (LA-DSE), a gép repült óráinak száma és a gép típusa. A gép típusát definiálja a típusazonosító (B767), a hossz és a maximális utaslétszám. A társaságnak különböző típusú gépei vannak, de a típus jellemző adatait az általa nem használt gépekre vonatkozóan is nyilvántartja. Minden repülőgépet két pilóta (első és másod) vezet. A pilótákat jellemzi a nevük és az összes repült órák száma. A pilóták csak azokat a típusú repülőket vezethetik, amelyekre érvényes típusvizsgájuk van. Olyan pilóta nincs, aki nem repül. A típusvizsga jellemzője a megszerzés és az érvényesség dátuma.

JÁRAT (szám, indulás, cél)

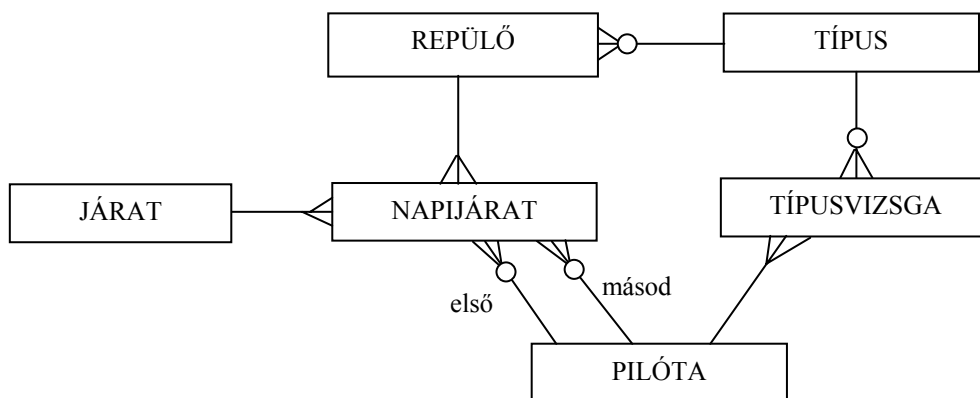
NAPIJÁRAT (járat, dátum, utasok száma, késés, repülő, pilóta(első), pilóta(másod))

REPÜLŐ (lajstromjel, repült óra, típus)

TÍPUS (azonosító, hossz, maxutasszám)

PILÓTA (név, repült órák)

TÍPUSVIZSGA (típus, pilóta, megszerzés, érvényesség)



mindegyik entitás és mindegyik reláció 0.5 pontot ér, max. 6 pont (6.5 pont van benne !!!)

5. Milyen típusú kohézióba sorolható a C nyelv `printf()` függvénye ? (2 pont)

procedurális

kommunikációs

esetleges

funkcionális

helyes válasz 2 pont, minden rossz -1 pont, de ≥ 0

Nevezzen meg legalább egy olyan kohéziós típust, amely a fentiek között nem szerepel ! (1 pont)

temporális vagy logikai vagy szekvenciális

első helyes 1 pont, minden további helyes 0.5 pont, rossz -1 pont, de total ≥ 0

6. Készítsen **JSD** modellt az alábbi problémára ! (11 pont)

A Lehetetlenségtudományi Tanszéken minden évben január végén kezdődik a diplomatervezés azzal, hogy a diplomázók beadják a - minden lényeges információt tartalmazó – adatlapot az adminisztrációban. Ezt követően a hallgató dolgozik, jár konzultálni, amiről az oktató rendszeresen jelentést készít az adminisztrációnak. A diplomaterv beadását (május közepe) követően a diplomatervet több bírálónak kiküldik bíráltni. Minimum két bírálat készül, de inkább három. Sajnos a bírálókkal előfordul, hogy a megadott határidőre nem készítik el a bírálatot. Ha letelt a határidő, akkor a bírálót SMS-ben figyelmeztetik. Előfordul, hogy a figyelmeztetés ellenére sem készül el végül a bírálat. Az államvizsgán a hallgatónak be kell mutatnia munkáját és a bírálatokban szereplő megjegyzésekre reagálni kell. A bírálók a bírálat elkészítéséért utólag díjazásban részesülnek, ha nem késtek a bírálással.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha olyan hallgató akarja beadni a diplomatervét, aki nem járt rendszeresen konzultálni !

Kérésre listázza ki azokat a hallgatókat, akiknek már legalább két bírálatuk bérkezett !

Kérésre adja meg azokat a bírálókat, akik késve küldték meg a bírálatot !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
kapcsolja a modellhez a **funkció processzeket.**

adatlapot bead (hallgató) eseménylista 2 pont

konzultál (hallgató)

diplomatervet bead (hallgató)

bírálatra megy (bíráló)

határidő lejár (bíráló)

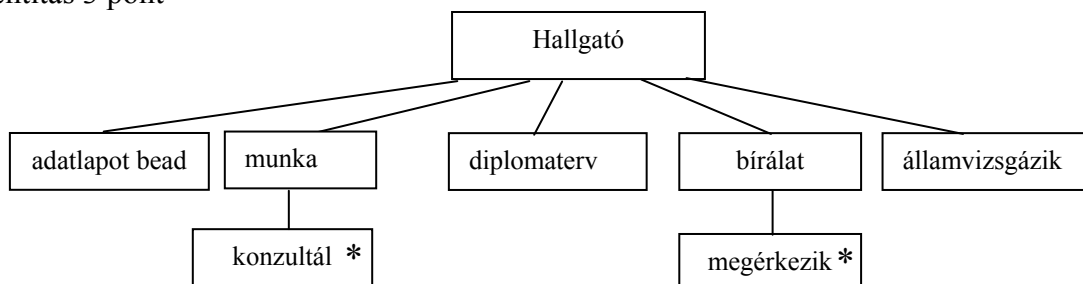
SMS-t küld (bíráló)

bírálat megérkezik (bíráló, hallgató)

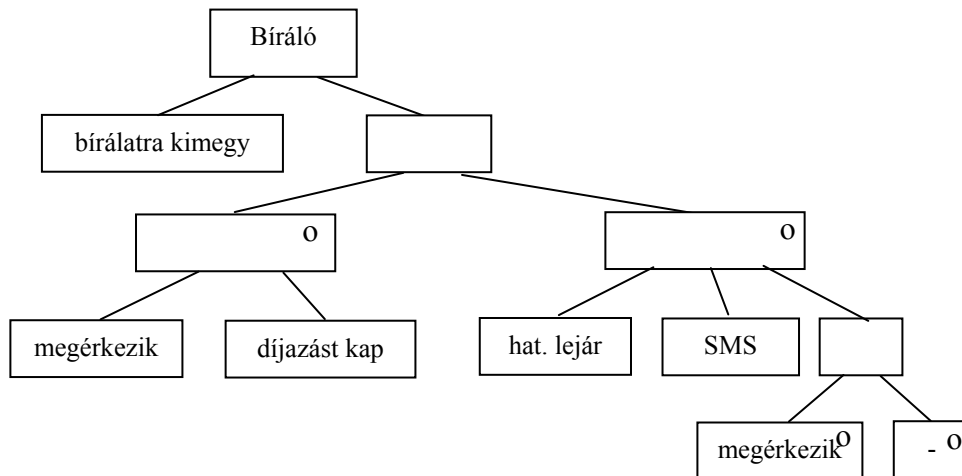
államvizgázik (hallgató)

díjazást kap (bíráló)

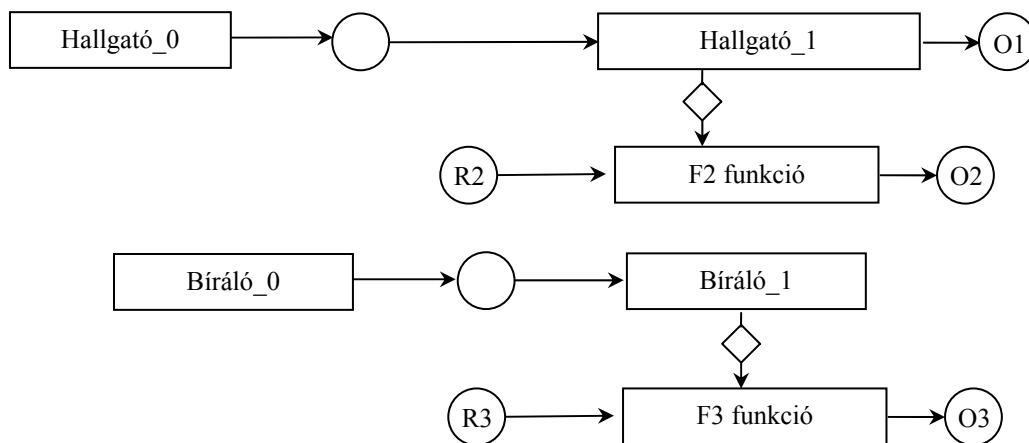
Hallgató entitás 3 pont



Bíráló entitás 4 pont

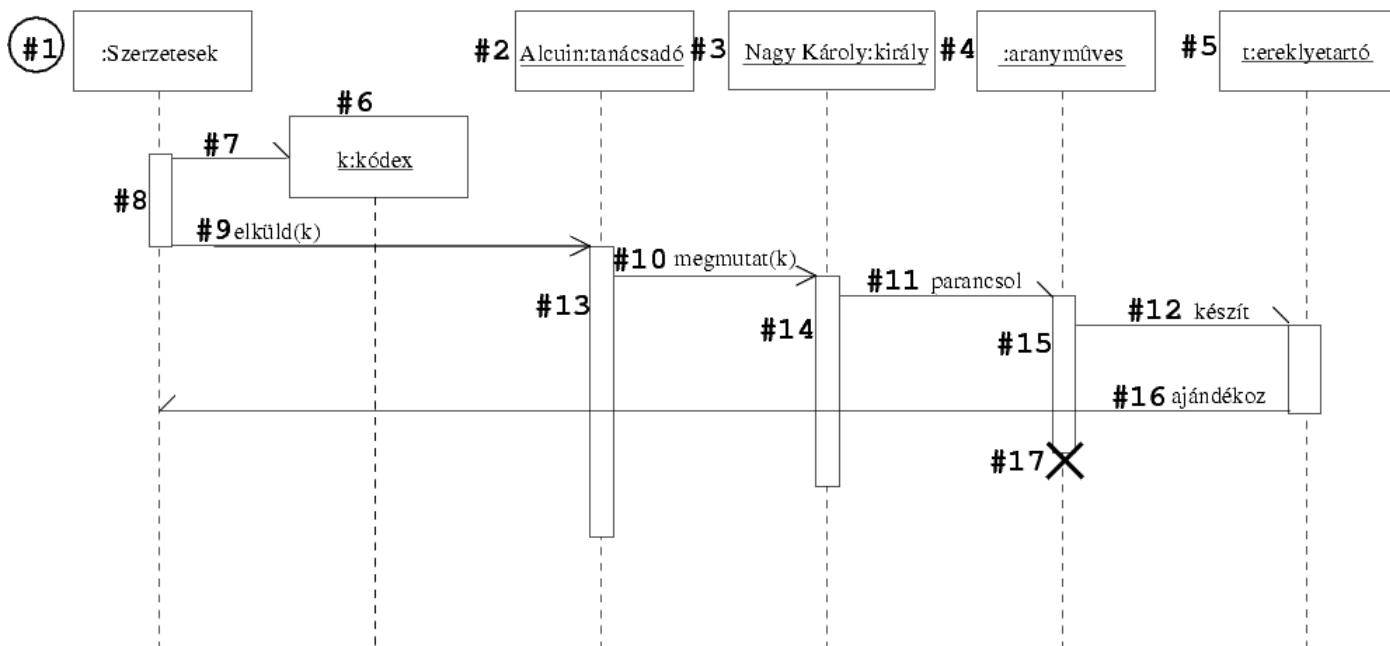


teljes SSD 2 pont



7. Ionai szerzetesek 794. körül Donaldusnak, a lúdtalpasok védőszentjének életéről készítenek gyönyörű kódexet, amit elküldenek Alcuinnak, Nagy Károly tanácsadójának. Kézhezvételkor Alcuin egyből a királyhoz siet, és megmutatja neki a művet. Károlynak annyira tetszik a kódex, hogy udvari aranyműveséhez a következő parancsot juttatja el: készítsen arany ereklyetartót Donaldus szent bokacsontjának, és ha elkészült, ajándékként küldje el a szerzeteseknek.

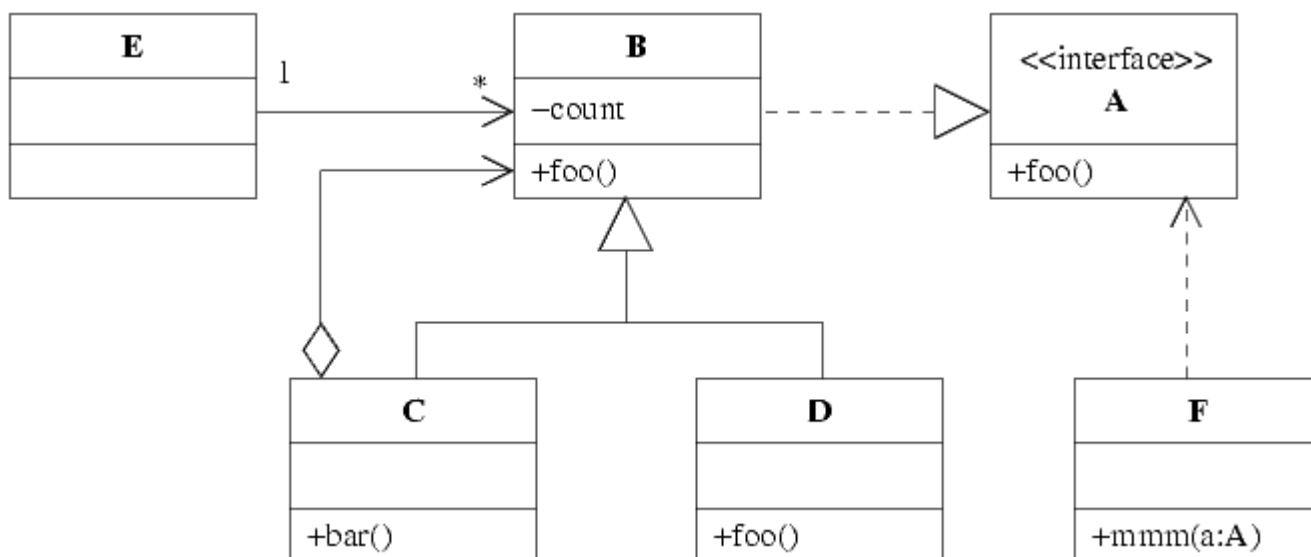
A fenti történetre Gáz Géza az alábbi, több hibát is tartalmazó UML szekvencia diagramot készítette. A diagramon néhány markert (#X) helyeztünk el. Karikázza be azokat a markereket, amelyek hibákat jelölnek. (Példa: #1-nél hiba van, mert az objektum neve nincs aláhúzva.) (7 pont)



Hiba: 5, 7, 9, 12, 14, 16, 17

Minden eltalált hiba 1 pont, minden tévedés -1

8. Az alábbi UML osztálydiagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz
- B - csak a második tagmondat igaz
- C - mindkét tagmondat igaz, de a következtetés hamis
- D - mindkét tagmondat igaz és a következtetés is helyes
- E - egyik tagmondat sem igaz
- (F - az ki van zárva, mert nem tudom)

- [B] D osztály nem definiálhatja felül a `foo()` metódust, mert nem ismeri az A interfészt.
- [D] E osztályú objektum nem hívhatja meg C osztályú objektum `bar()` metódusát, mert csak a B osztályt ismeri.
- [B] C osztály helyettesíthető D-vel, mert közös az ősök.
- [B] E osztály ismeri az A interfészt, mert ismeri utóbbi egy megvalósítását.
- [A] C osztály nem ismeri D osztályt, ezért C osztályú objektum nem hívhatja meg D osztályú objektum `foo()` metódusát.
- [D] F osztály `mmm(a:A)` metódusa paraméterül kaphat D osztályú objektumot, mert utóbbi megvalósítja az A interfészt.
- [A] C osztály nem látja B osztály `count` attribútumát, mert az attribútum `protected`.
- [E] F osztályú objektum tartalmazhat tetszőleges számú A interfészű objektumot, mert F megvalósítja A interfészt.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total ≥ 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2004. május 19.

1. Mi(k)hez kapcsolódik a CORBA rendszerben az "IDL váz" (angolul: "IDL Skeleton") ? (4 pont)

- | | |
|--|---|
| <input type="checkbox"/> implementációs háttértár (repository) | <input type="checkbox"/> entitás bean |
| <input type="checkbox"/> ORB mag (core) | <input checked="" type="checkbox"/> szervant |
| <input checked="" type="checkbox"/> objektum adapter | <input type="checkbox"/> kliens |
| <input type="checkbox"/> objektum attribútum | <input type="checkbox"/> interfész háttértár (repository) |

mindegyik jó 2 pont, mindegyik hiba -1, de total >= 0

2. A kockázat tervezése során milyen stratégiákat választunk ? (4 pont)

A kockázat valószínűségének csökkentése
A bekövetkező káros hatások minimalizálása
A nem kívánatos eseményt követően felmerülő tevékenységek tervezése

egy jó 2 pont, a háromból kettő már elég

3. A C osztály M metódusából a Demeter törvény szerint az alábbi program-elemek érhetők el: (6 pont)

az M paraméterei.....	this, super
a C osztály és példány változói (attribútumok).....	globális változók.....
Az M -ben létrehozott átmeneti változók.....	az M által létrehozott objektumok.....

mindegyik jó 1 pont

4. Elkészítjük az alábbi X osztály két példányát, x1-et és x2-t. Ezt követően végrehajtjuk a következő két műveletet:

x2.a = x1.a + 5;
x1.a +=5;

X
<u>int a = 0</u>
private aX():

Mennyi lesz x1.a és x2.a ? (3 pont)

x1.a = **10** x2.a = **10**.....

mindegyik jó 1,5 pont

5. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (7 pont)

Egy légitársaság több repülőgéptípust is használ. A típusokra jellemző a gyártó, a típuszám, és a gép súlya, valamint névleges fogyasztása. Döntéselőkészítési és összehasonlítási célból olyan típusok adatait is tárolják, amilyen repülőkkel nem rendelkeznek. A repülőket a lajstromjel azonosítja (pl. HA-LGO), jellemzőjük még a gyártás éve, és a vételára. A repülők járatokon repülnek. A járat jellemzője a járatszám, a dátum és a késés. Ugyanazzal a járatszámmal nem indulhat két járat ugyanazon a napon. A járatszám (pl. MA 091) tartalmazza a menetrend szerinti indulási és érkezési helyet és időt. A repülőgépet a kapitány vezeti, akit jellemez a neve, kora, engedélyének száma és érvényességi ideje. Csak az lehet kapitány, akinek legalább egy repülőgéptípusra van ún. típusvizsgálója (fontos a vizsga napja és eredménye), amely szükséges ahhoz, hogy az adott típusú gépet a kapitány vezesse.

TÍPUS (gyártó, típuszám, súly, fogyasztás)

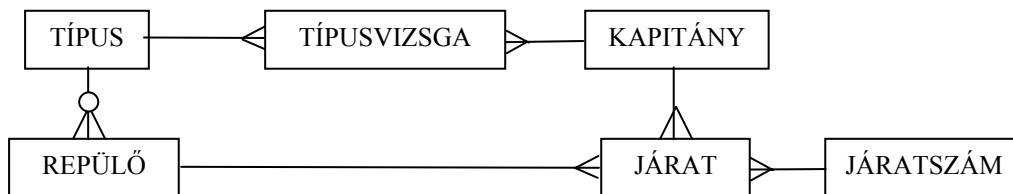
REPÜLŐ (típus, lajstromjel, gyártási év, vételár)

JÁRAT (járatszám, dátum, késés, kapitány, repülő)

JÁRATSZÁM (szám, indulás hely, érkezési hely, indulási idő, érkezési idő)

KAPITÁNY (név, kor, engedély száma, érvényességi ideje)

TÍPUSVIZSGA (kapitány, típus, vizsga napja, eredménye)



mindegyik entitás és mindegyik reláció 0.5 pontot ér.

6. Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

Az ABS (Advanced Business School) féléves tanfolyamokat tart. A hallgatók megadott határidőig jelentkeznek, majd befizetik a tandíjat. Ezt követően minden hallgatónak kötelező felvenni öt tantárgyat. A tantárgyak óráin való megjelenést regisztrálják. Minden tárgyból félév közben zárthelyit íratnak (pótlás nincs, aki nem jön el egyest kap). Aki az órák több mint 80 %-án megjelenik és a zárthelyit legalább 4-re megírja, annak vizsga nélkül adnak félévi jegyet. A többieknek vizsgázni kell. Egyetlen vizsgát tartanak tárgyanként. Az ABS nagy figyelmet fordít az egészséges életmódra, ezért a szorgalmi időszakban kéthetente teniszversenyt rendeznek. Akik mind a sportban, mind a tárgyakban kiváló eredményt érnek el, megkapják a „Jó tanuló – jó sportoló” kitüntetését.

Definiálja az alábbi funkciókat ellátó processzeket:

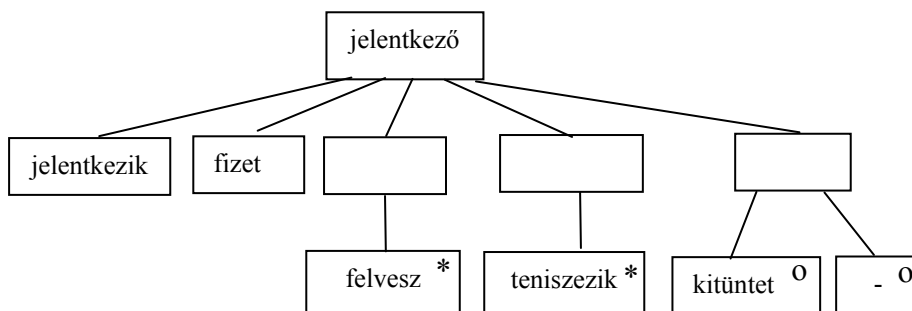
Kérésre készítse el a kitüntetésre javasolt hallgatók listáját ! Azokat válogassa ki, akik csak jó és jeles eredményt értek el, legalább két tárgyból megajánlott jegyet kaptak és minimum két alkalommal bejutottak a legjobb négy közé a teniszversenyen.

Kérésre adjon megajánlott jegyet a feltételeknek megfelelő hallgatóknak !

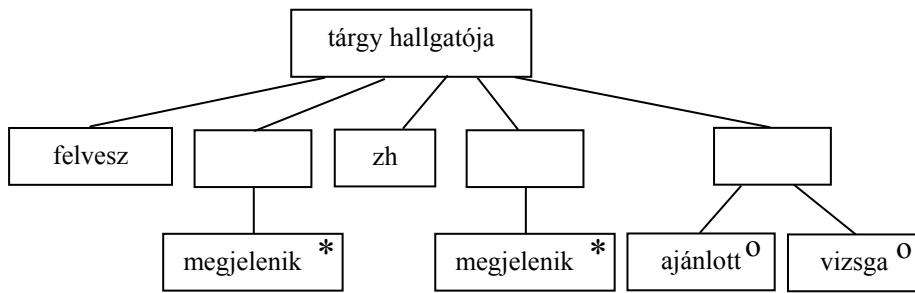
Azonnal jelezzen, ha valaki harmadik alkalommal végez az első négy között a teniszversenyen !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

jelentkezik (jelentkező) 2 pont
fizet (jelentkező)
felvesz (jelentkező, tárgyhallgatója)
teniszezik (jelentkező)
kitüntet (jelentkező)
megjelenik (tárgyhallgatója)
zh (tárgyhallgatója)
megajánlott (tárgyhallgatója)
vizsga (tárgyhallgatója)

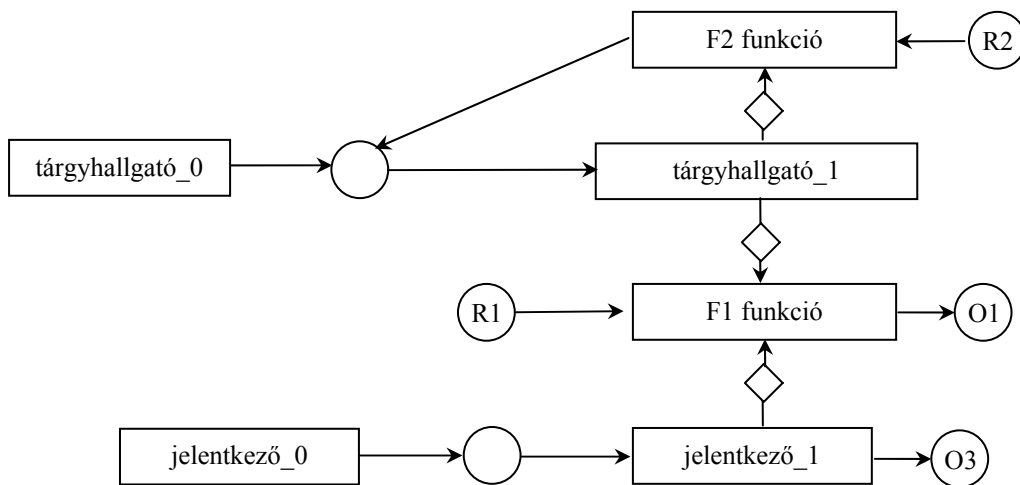


4 pont



3 pont

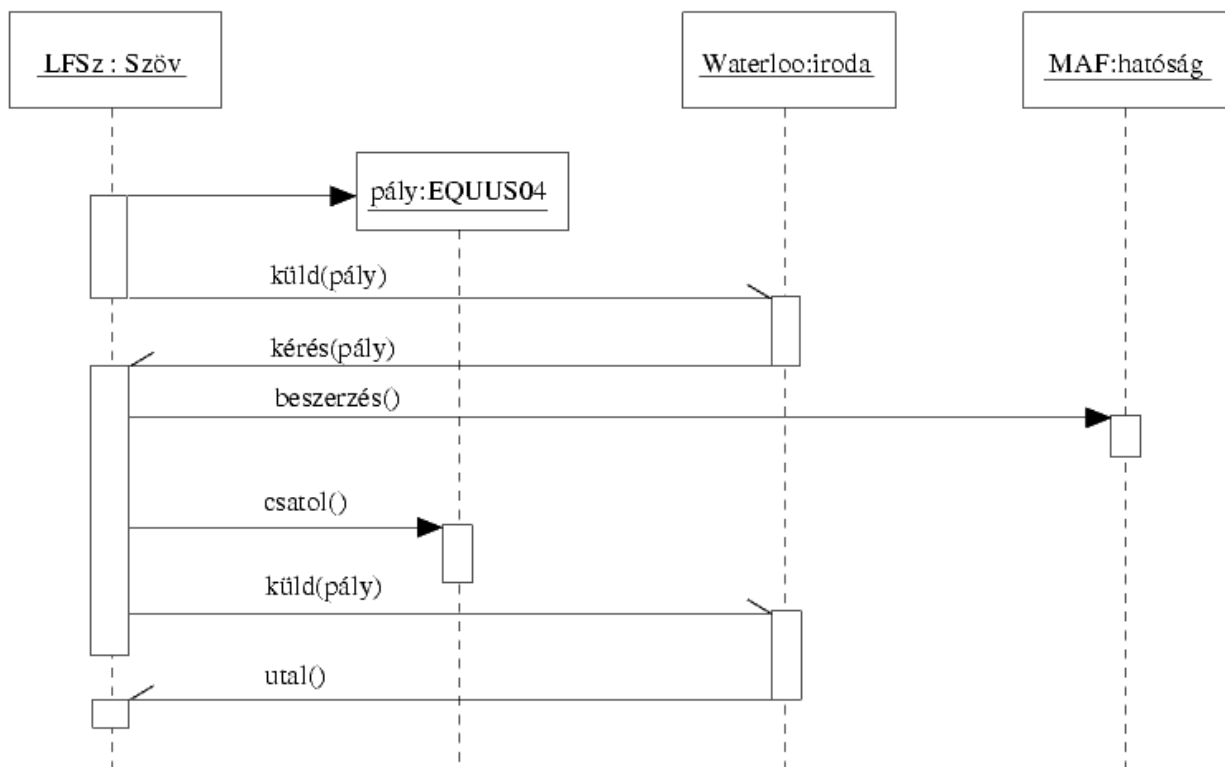
2 pont



7. A Lótornyai Faipari Szövetkezet európai uniós támogatást szeretne kapni korhú kuruckori lószerszámok készítéséhez. Ezért elkészítik az EQUUS04 kiírásnak megfelelő pályázatukat, amit el is küldenek az EU waterloo-i irodájába.

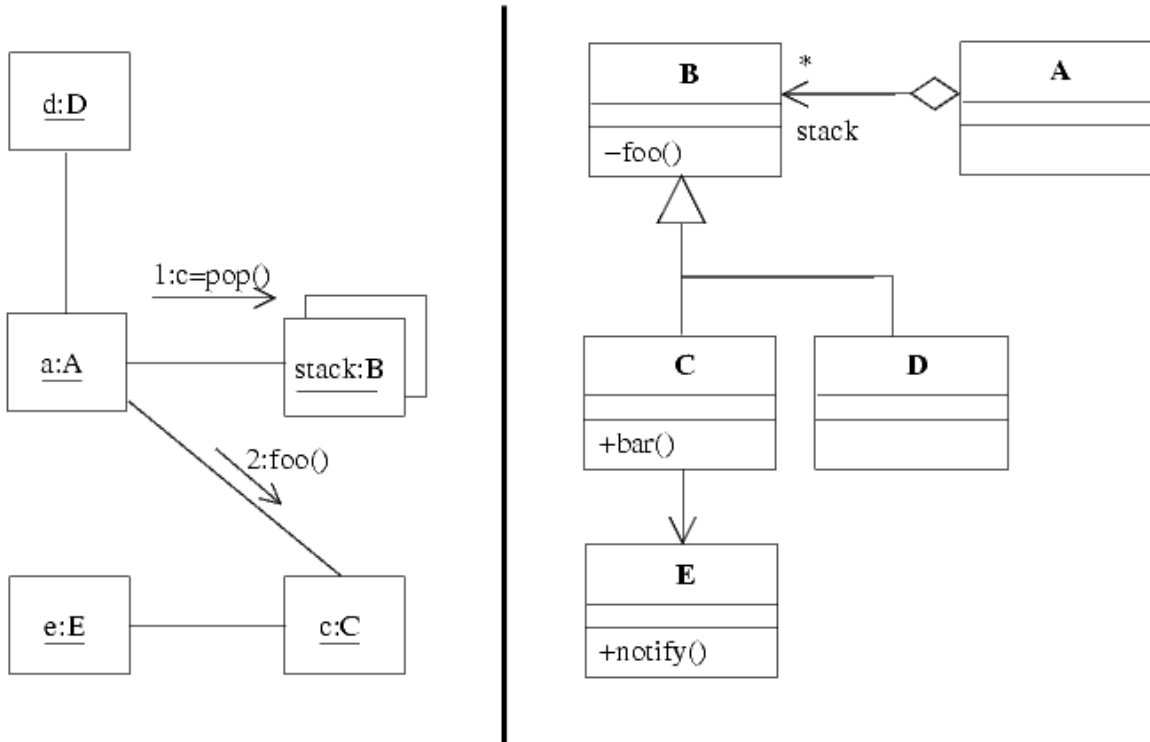
Egy hét múlva visszakapják az anyagot, azzal a felszólítással, hogy fűzzék hozzá a cég lóherékkel kapcsolatos nukleáris tevékenységét igazoló hatósági iratot. Az iratot a Magyar Atomenergia Felügyeletől szerzik be, csatolják, és az így kibővített anyagot ismét elküldik Waterloo-ba. Egy hónappal később az iroda a támogatást megküldi a szövetkezetnek.

A fenti történet alapján rajzoljon **UML szekvencia diagramot** ! (8 pont)



mindegyik aszinkron és pályázat 1 pont, a többi objektum és üzenet 0,5 pont

8. Az alábbi összefüggő UML osztálydiagram és kollaborációs diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (7 pont)



- A - csak az első tagmondat igaz
- B - csak a második tagmondat igaz
- C - mindkét tagmondat igaz, de a következtetés hamis
- D - mindkét tagmondat igaz és a következtetés is helyes
- E - egyik tagmondat sem igaz
- (F - az ki van zárva, mert nem tudom)

- [C] a nem küldhet `notify()`-t e-nek, mert a B osztály nem ismeri E osztályt
- [B] a nem teheti d-t a stack-be, mert stack nem D osztályú
- [D] a nem küldhet c-nek `bar()` üzenetet, mert az A osztály nem ismeri C osztályt
- [B] d meghívhatja c `foo()` metódusát, mert közös az ősük
- [E] A osztályú objektum helyettesíthető B osztályúval, mert B A leszármazottja
- [C] e nem hívhatja meg c `bar()` metódusát, mert nem ismeri C ősosztályát
- [B] e meghívhatja c `bar()` metódusát, mert a metódus publikus

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból
2004. május 25.

1. Az Aspektus Oriénált Programozásban (3 pont)

mi biztosítja a genericitást: aspektusok szövése

mik az interfészek: a csatlakozó pontok (join points) a szövéshez

hogyan adaptálunk: ragasztó kódok és csomagolók szövése, aspektusok változtatása

2. Egy adatfolyam-ábrán a DB1 és DB2 adattárak, valamint az X1, X2 és X3 processzek szerepelnek. A vezérlőegység bemenetét a következő specifikáció írja le: [alfa|beta|gamma]. Töltse ki a vezérlőegység állapottábláját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) az adatfolyam és adatspecifikációval ! (5 pont)

BB				
AA				

az oszlopok paraméterezése (sorrend lényegtelen) csak a bemenet konstansai lehetnek 1 pont

a következő állapotok csak a bejelölt két állapot (BB, AA) vagy don't care lehet, kell benne lenni állapotváltásnak is. 2 pont

az akció ha nem üres, csak processz (X1, X2, X3) lehet és mindegyiknek legalább egyszer szerepelni kell. 2 pont

3. Legyen egy X objektumnak egy `int vizsga(int x)` metódusa, ahol csak $10 < x < 50$ megengedett, és akkor a $50 < vizsga < 100$ igaz. Az X leszármazottjában felüldefiniáljuk `vizsga-t`. Az alábbi állítások közül jelölje meg azokat, amelyek ha igazak a felüldefiniált `vizsga-n`, akkor az kielégíti a Liskov-elvet ! (6 pont)

$20 < x < 40$

$60 < vizsga < 90$

$00 < x < 50$

$00 < vizsga < 120$

$00 < x < 1000$

$50 < vizsga < 1000$

$30 < x < 60$

$40 < vizsga < 110$

minden helyes válasz 2 pont, minden rossz -1 pont

4. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet ! Az axiómák felírásakor megengedett a két karaktert összehasonlító művelet használata. (5 pont)

CRT()	új (üres) stringet hoz létre.
LGTH(s)	az s string karaktereinek számát adja.
ADD(s, x)	az s string végére rakja az x karaktert.
LAST(s)	az s string végén álló karaktert adja.
SUBS(s,i,x)	az s string i -ik pozícióján (i pozitív egész) álló karaktert x karakterre változtatja. Ha a string rövidebb, mint i , akkor a művelet hatástalan. Az első karakter pozíciója 1.
DUPLO(s)	igaz, ha az s stringben legalább egyszer dupla karakter fordul elő. Például: aggódik

LAST(CRT()) = undefined	- 0,5 pont, ha hiányzik
LAST(ADD(s, x)) = x	- 0,5 pont, ha hiányzik
LGTH(CRT()) = 0	- 0,5 pont, ha hiányzik
LGTH(ADD(s, x)) = LGTH(s) + 1	- 0,5 pont, ha hiányzik
SUBS(CRT(), i, x) = CRT()	0,5 pont
SUBS(ADD(s, x), i, y) = if (i > LGTH(s) + 1) ADD(s, x)	1 pont
if (i == LGTH(s) + 1) ADD (s, y)	1 pont
if (i < LGTH(s) + 1) ADD(SUBS(s, i, y), x)	1 pont
DUPLO(CRT()) = false	0,5 pont
DUPLO(ADD(s,x)) = (x == LAST(s)) or DUPLO(s)	1 pont

5. Nevezze meg sorrendben a CMM (Capability Maturity Model) szintjeit ! (5 pont)

1. kezdetleges (initial)
2. ismétlődő (repeatable)
3. definiált (defined).....
4. irányított (managed)
5. optimalizált (optimizing).....

minden találat 1, minden hiba -0.5 pont

6. Készítsen **JSD** modellt az alábbi problémára ! (11 pont)

Az XXX cég autókölcsönzéssel foglalkozik. A bérelt autót, csak az ügyfél vezetheti, következésképp egy ügyfél egyszerre csak egy autót bérelhet. A bérlet normális esetben úgy ér véget, hogy az ügyfél a lejárat határidő előtt leadja az autót. Előfordulhat, hogy az autó baleset vagy műszaki okok miatt mozgásképtelenné válik. Ebben az esetben az ügyfél a céget értesíti, a továbbiakat a cég rendezi. Ha az autót az ügyfél a megadott határidőig nem adja le, akkor a cég SMS-ben üzenetet küld az ügyfélnek. Ha egy ügyfél többször is késedelmesen hozza vissza az autót, akkor az ügyfelet letiltják és soha többé nem kölcsönözhet.

A cég az autókat egy márkakereskedőtől szerzi be. A kifogástalanul működő autókat is rendszeresen szervizelik. Akkor, ha egy autót mozgásképtelennül hoznak be, azt okvetlenül szervizbe küldik. A lehasznált autókat átadják a márkakereskedőnek.

Definiálja az alábbi funkciókat ellátó processzeket:

Kérésre listázza ki azokat az autókat, amelyeket már legalább ötször vittek szervizbe és a következő kölcsönzés után ismét szervizelni kell !

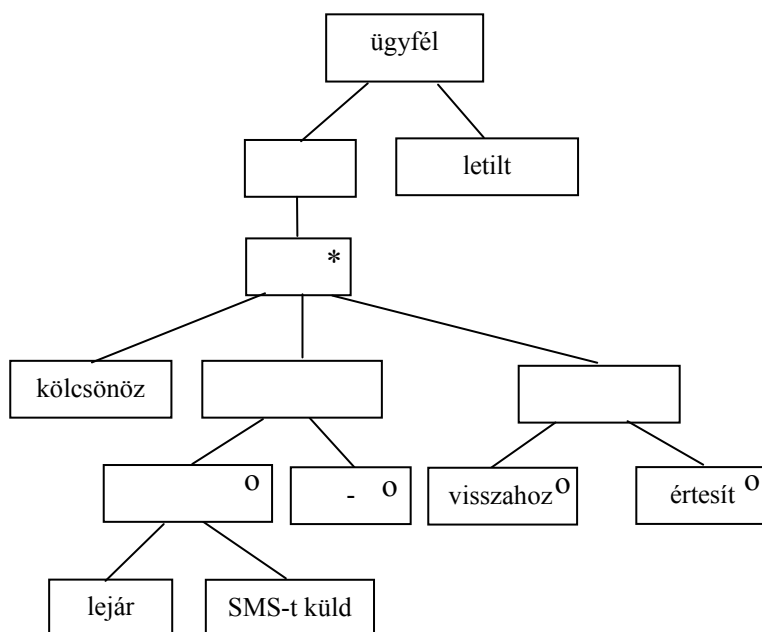
Kérésre listázza ki azokat az ügyfeleket, akiknek már küldtek SMS-t a késedelem miatt, de még mindig nem hozták vissza az autót !

Automatikusan adjon figyelmeztetést, ha egy ügyfél esetében harmadszor fejeződik be a kölcsönzés azzal, hogy az autó mozgásképtelenné válik !

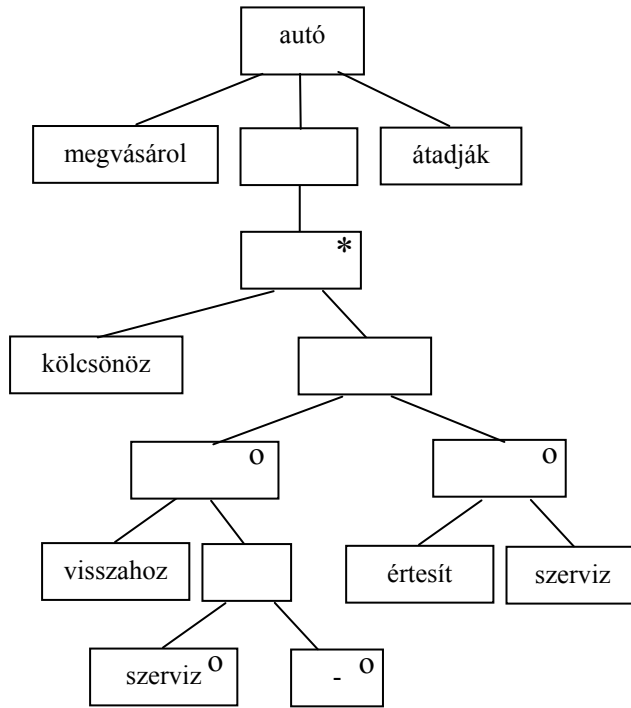
Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
 kapcsolja a modellhez a **funkció processzeket.**

- kölcsönöz (ügyfél, autó) 1 pont
- lejár az idő (ügyfél)
- SMS-t küld (ügyfél)
- visszahoz (ügyfél)
- értesít (ügyfél)
- letilt (ügyfél)
- megvásárol (autó)
- szervizre megy (autó)
- átadják (autó)

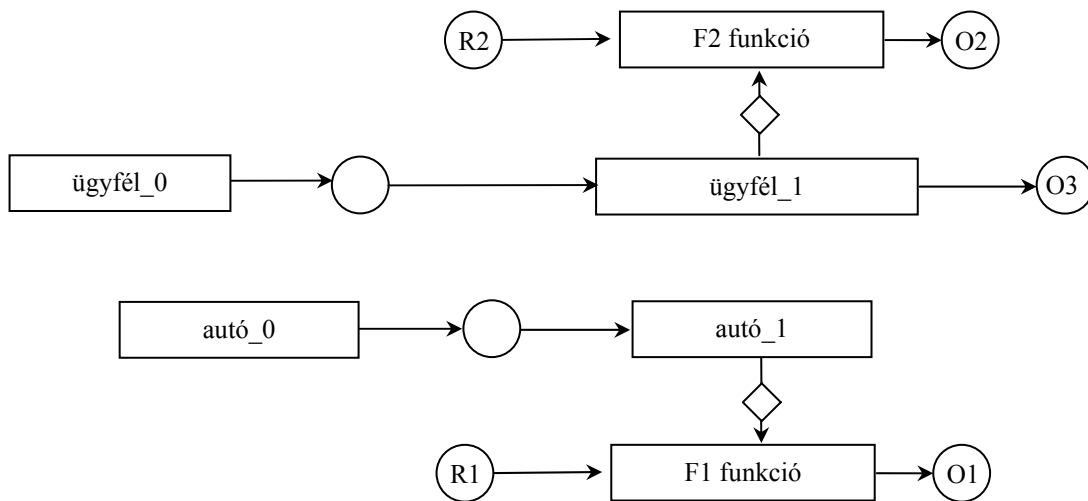
4 pont



4 pont



2 pont



7. Az elektronikus szerzetes (© Douglas Adams) a Secure Home and Industry Technologies legújabb készüléke.

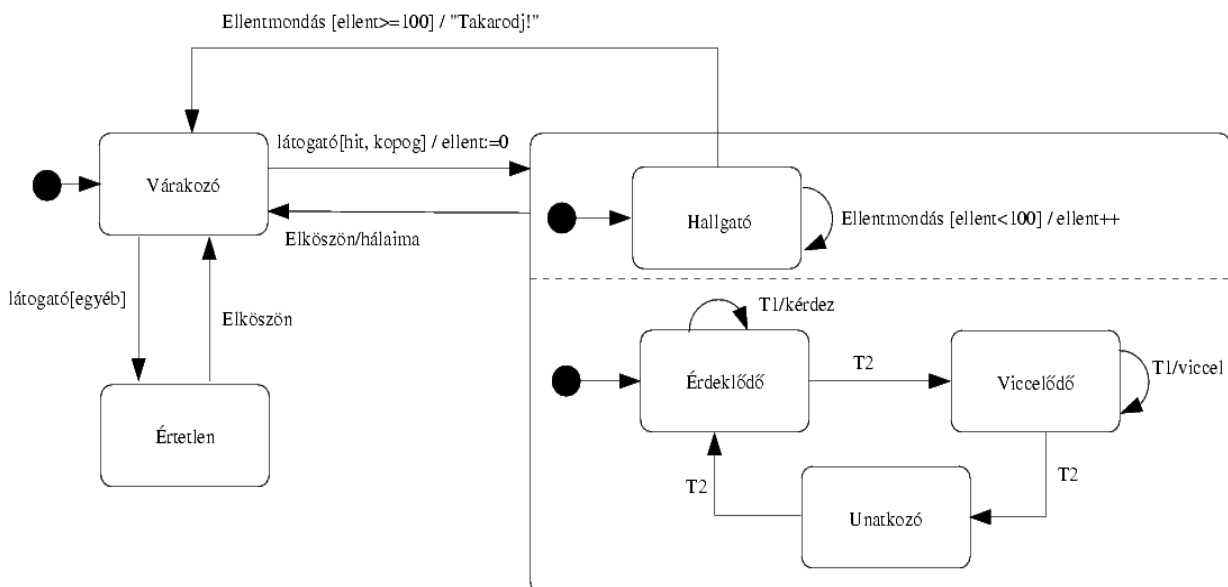
A szerzetes bekapcsolás után várakozó állapotba kerül. Hittérítő vagy kopogtatócédula-gyűjtő látogató érkezése esetén hallgató üzemmódra vált. Az üzemmód kezdetén nullázza az ellentmondásszámlálót, majd meghallgatja a mondandót. Ellentmondás meghallása esetén, ha a számláló értéke elérte a 100-at, akkor "takarodj!" felszólítással kitessekeli a látogatót és újra a várakozó állapotba kerül. Egyébként a számláló értéke eggyel nő.

Hallgatás közben, azzal párhuzamosan három állapotban lehet: unatkozik, ekkor unott arcot vág, érdeklődik, ekkor egy T1 órajel hatására kérdez, és viccelődik, ekkor T1 órajel hatására mond egy viccet. Az állapotok között T2 órajel hatására vált a fenti felsorolás szerinti soron következő állapotba (viccelés után unatkozik). Kezdetben mindig érdeklődik.

Ha a látogató elköszön, a szerzetes hálaimát rebeg, majd ismét várakozó állapotba kerül.

Ha nem a fenti típusú látogató jön, akkor értetlen állapotba kerül, ahol addig hallgat, amíg a látogató vagy el nem köszön, ekkor ismét várakozó állapotba kerül.

Rajzolja meg a szerzetes UML **állapot-diagramját** a fenti leírás alapján! (8 pont)



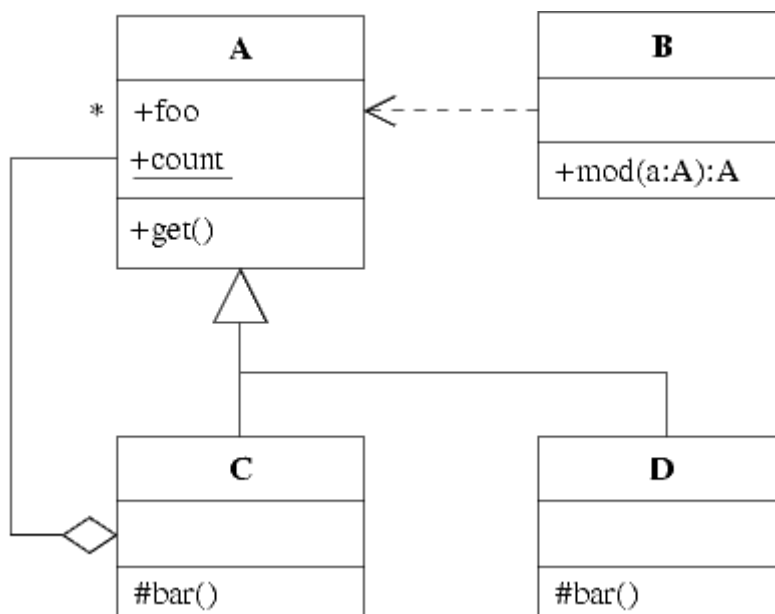
Kompozit állapot régiókkal – 1 pont

Hallgató alállapot – 1 pont

Párhuzamos alállapot – 2 pont

Kompozit állapoton kívül minden item 0,5 pont – összesen 4 pont

8. Az alábbi UML osztálydiagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (7 pont)



- A - csak az első tagmondat igaz
- B - csak a második tagmondat igaz
- C - mindkét tagmondat igaz, de a következtetés hamis
- D - mindkét tagmondat igaz és a következtetés is helyes
- E - egyik tagmondat sem igaz
- (F - az ki van zárva, mert nem tudom)

- [E] A osztályú objektum létrehozhat B osztályút, mert A függ B-től.
- [B] B nem látja A foo attribútumát, mert A-ban van publikus módszer.
- [E] C osztályú objektum nem tartalmazhat C osztályút, mert C bar () metódusa privát.
- [D] B osztályú objektum mod () metódusa kaphat C osztályú objektumot paraméterül, mert C A leszármazottja.
- [D] B az A count attribútumát A osztályú objektum létrehozása nélkül is módosíthatja, mert az attribútum statikus.
- [E] D nem tartalmazhat bar () metódust, mert az ábrán többszörös öröklődés szerepel.
- [B] C ismeri D-t, mert B ismeri az őseit.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2002. január 8. 10 óra

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A PÉTÉ HOTEL-ben található szobákat a számuk azonosítja; tudjuk mindegyik szobáról, hogy hány ágy van benne és a szoba pótágyazható-e. Egy személy egy éjszakáért fizetendő díja függ a szobától, a vendégtől (törzsvendég-e, gyerek, stb.), de a szálloda egyedi díjakat is alkalmaz. A szállodába érkező vendégek mindegyike kártyát kap, amely nyitja a szoba ajtaját, de minden egyéb szolgáltatás (étterem, szauna, stb.) is csak a kártyával vehető igénybe. A szálloda elhagyásakor történik a fizetés. Az igénybe vett szolgáltatásról tudjuk annak idejét, megnevezését (étterem, konditerem, stb.) és összegét. A szálloda a vendégek adatait (név, cím, telefon, email, foglalkozás) megőrzi, sőt azt is, hogy a vendég mikor, mennyi időt töltött a szállodában és mennyit fizetett alkalmanként.

SZOBA (szám, ágy, pótágyazhatóe)

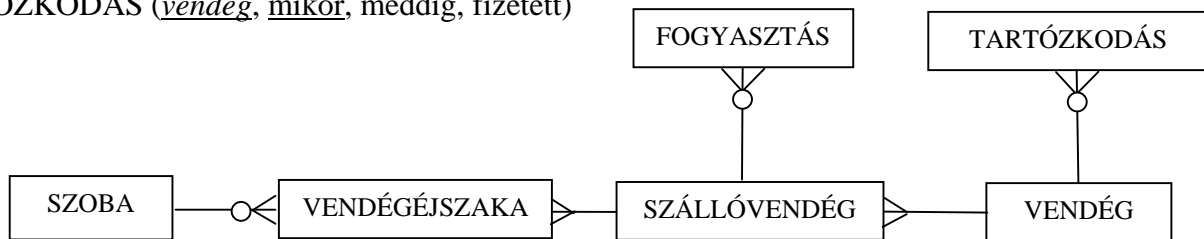
VENDÉGÉJSZAKA (szoba, szállóvendég, dátum, ár)

SZÁLLÓVENDÉG (vendég, kártyaszám)

FOGYASZTÁS (szállóvendég, idő, megnevezés, összeg)

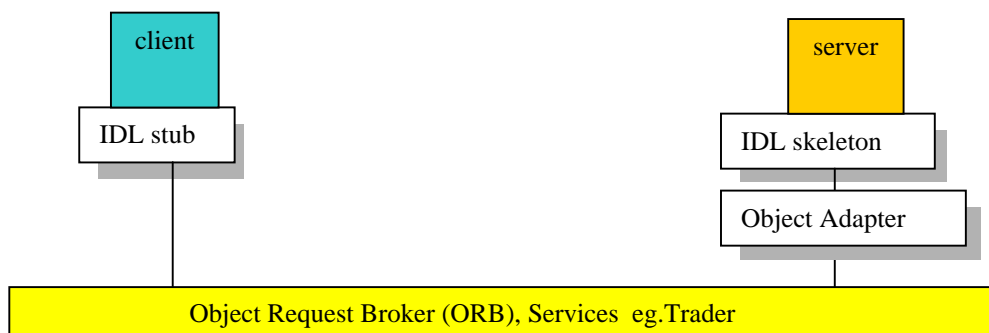
VENDÉG (név, cím, telefon, email, foglalkozás)

TARTÓZKODÁS (vendég, mikor, meddig, fizetett)



mindegyik entitás és mindegyik reláció 0.5 pontot ér. 3 pont fölött +0.5 pont.

2. Rajzolja fel a CORBA architektúrát. (3 pont)



Mindegyik elem 0.5 pont

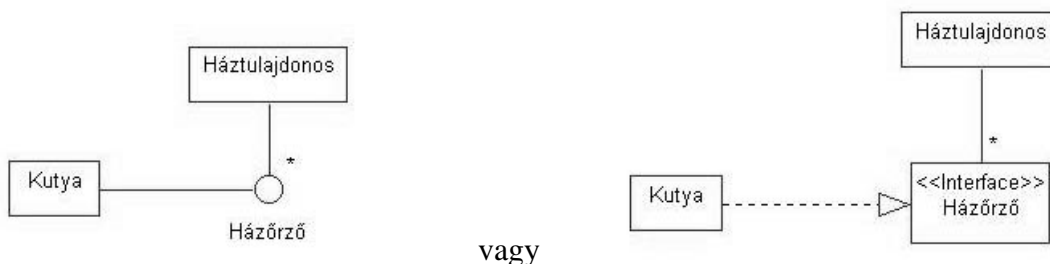
3. A tervezési elvek között bevezettük a kohézió és csatolás fogalmait. Mit jelentenek ? Milyen fő típusaik vannak ? (4 pont)

Kohézió: mennyire kell ismerni kívülről egy (al)rendszer belső szerkezetét. Mennyire fekete a doboz ?
 Típusai: funkcionális, szekvenciális, kommunikációs, procedurális, temporális, logikai, véletlen

Csatolás: egy (al)rendszer komponensei között kapcsolat erőssége.

Dimenziók: kapcsolat módja: paraméter, adat, vezérlés, tartalmi
 kapcsolat ideje: run-time, load, link, compile, source
 kapcsoló adatok mennyisége: kevés, sok

4. A Kutya osztály megvalósítja a Házőrző interfészt. A Háztulajdonos a kutyáit házőrzésre használja. Rajzoljon UML szerkezeti diagramot ! (3 pont)



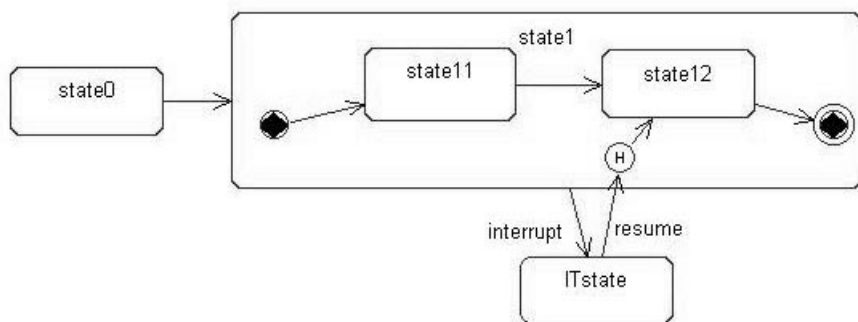
Házőrző ábrázolás 1 pont
 Házőrző és Kutya kapcsolat 1 pont
 Háztulajdonos Házőrző kapcsolat multiplicitással 1 pont

5. Mikor használjuk és hogyan jelöljük a history indicatorot ? (6 pont)

Hierarchikus állapotmodellben megőrizzük, hogy melyik alállapotból léptünk ki. A visszatéréskor a megőrzött alállapotba lépünk. Az indikátorból kivezető nyíl a predefinit visszatérő állapotba mutat. Az indikátor jele a körbe zárt H betű.

indicator jele – 2 pont
 visszatérés megőrzés – 2 pont
 predefinit eset – 2 pont

rajz nem kell, csak illusztráció.



6 Adjon **JSD** alapú megoldást az alábbi problémára ! (11 pont)

A *NAUTILUS búvárklub* amatőr könnyűbúvárok számára tanfolyamokat szervez. A jelentkezők közül csak az orvosi vizsgálaton megfelelőnek bizonyultak vehetnek részt a foglalkozásokon. Elméleti órákat tartanak, ahol a hallgatóság megismerkedik a merülés szabályaival és a felszerelés használatával, működésével. A sikeres elméleti vizsga feltétele, hogy a hallgatónak az órák 80 %-án részt kell venni. A sikertelen vizsga tetszőleges sokszor megismételhető. Aki az előírtnál kevesebb órán vett részt, annak újra kell kezdeni a tanfolyamot új jelentkezőként. A gyakorló merülésen az a hallgató vehet csak részt, aki sikeres elméleti vizsgát tett – bármelyik búvárklubban. Az oktató nevével és a merülés időpontjával meghirdetett gyakorló merülésekre a hallgatónak előzetesen be kell jelentkezni, mert egy oktató mellett maximum 5 hallgató merülhet. Minden hallgatónak legalább 5 sikeres gyakorló merülést kell teljesíteni ahhoz, hogy megszerezze az önálló merülésre jogosító bizonyítványt.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha valaki elsőre sikeres elméleti vizsgát tesz !

Azonnal jelezzen, ha valaki úgy kap önálló merülésre jogosító bizonyítványt, hogy nem volt sikertelen merülése !

Kérésre adja meg azokat, akik a még végre nem hajtott ötödik merülésükre más oktatóhoz jelentkeztek mint korábban !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
kapcsolja a modellhez a **funkció processzeket.**

jelentkezik (hallgató) eseménylista 2 pont

orvosin bukik (hallgató)

orvosin átmegy (hallgató)

órát látogat (hallgató)

vizsgán bukik (hallgató)

vizsgán átmegy (hallgató, merülő)

merülést hirdet (merülés)

merülésre jelentkezik (merülés)

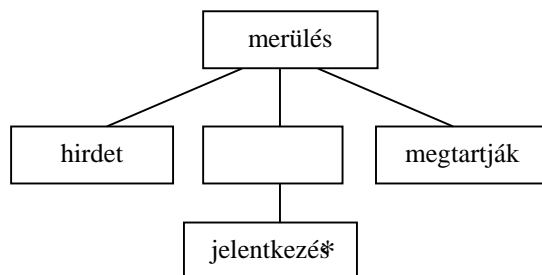
merülést megtartják (merülés)

sikeresen merül (merülő)

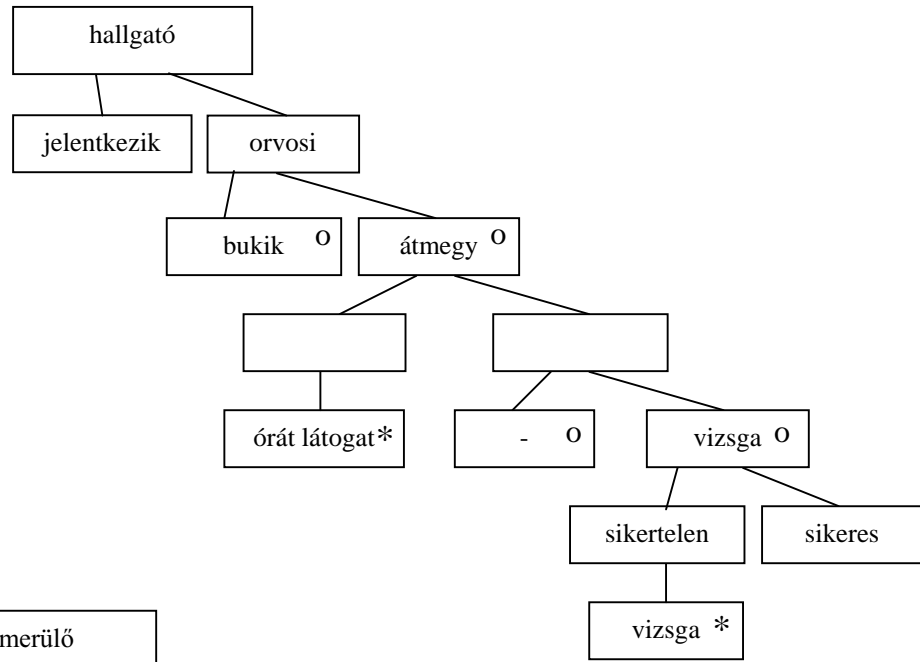
sikertelenül merül (merülő)

bizonyítványt kap (merülő)

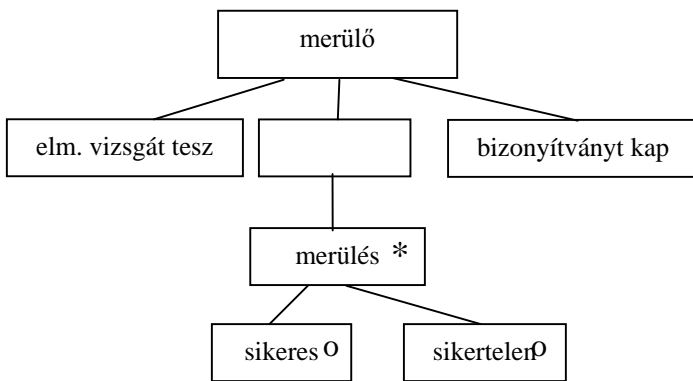
merülés entitás 1 pont



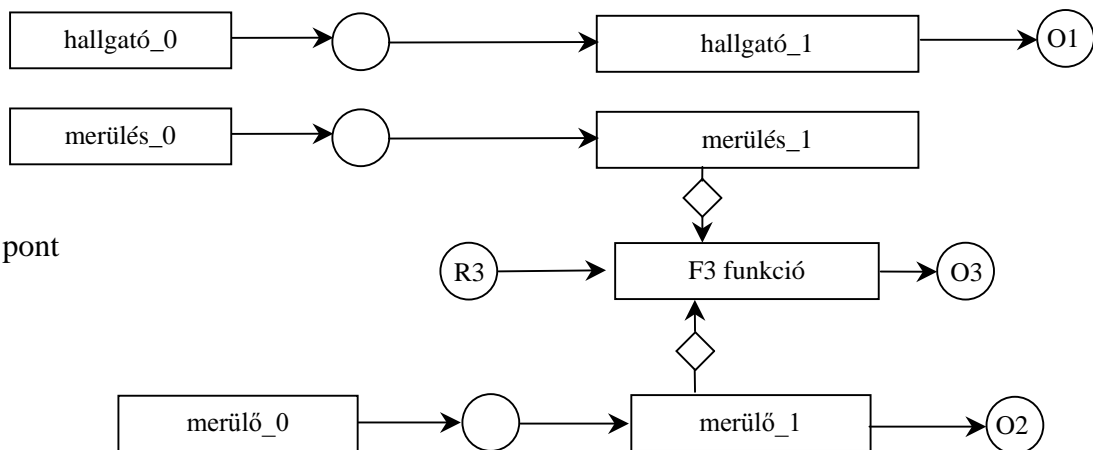
hallgató entitás 4 pont



merülő entitás 2 pont

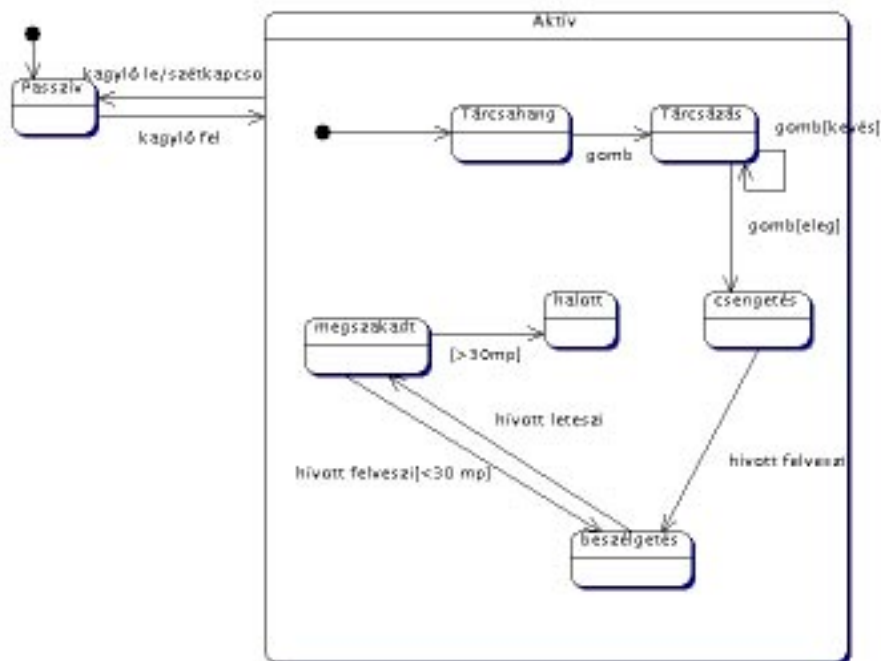


teljes SSD 2 pont



7. Rajzoljon **UML statechart**ot az alábbi feladat leírására ! (6 pont)

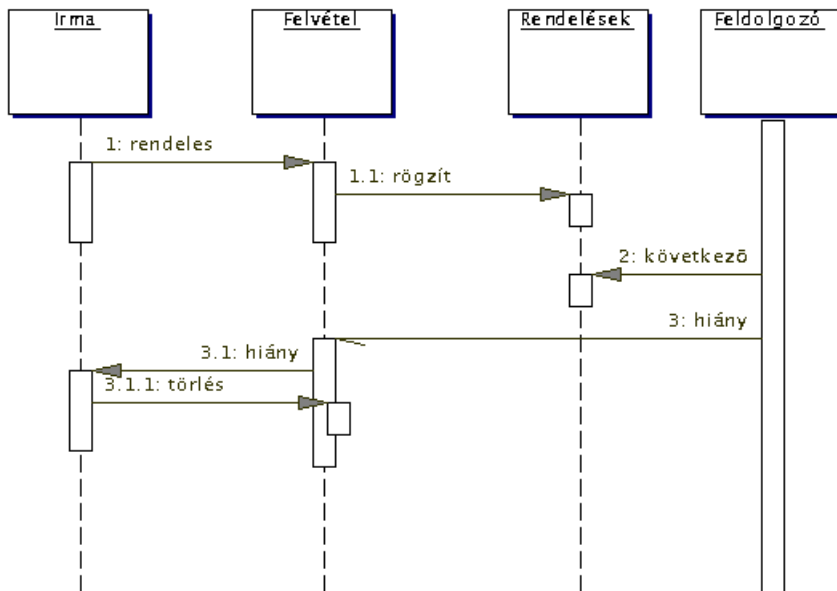
Egy telefon alapállapotban passzív. Ha megemeljük a kagylót, akkor aktív állapotba kerül. Aktív állapot először mindig tárcsahangot kapunk. Ha tárcsázunk, akkor a tárcsahang megszűnik. A tárcsázás végét a kellő mennyiségű számjegy jelzi. Ekkor a készülék kicseng, és ha hívott fél felveszi a kagylót, beszélhetünk. Ha a hívott fél beszélgetés közben leteszi, és 30 mp-en belül újra felveszi a kagylót, nem szakad meg a vonal. 30 mp után azonban bont a kapcsolat. Ha aktív állapotban akármikor letesszük a kagylót, a telefon passzív lesz.



2 fő állapot és skatulyázás 1 pont
minden él 0.5 pont

8. Rajzoljon az eseményekről **UML szekvencia-diagramot** ! (11 pont)

Irma szemöldökdúsító tablettákat szeretne. Felhívja a Kwelle rendelésvételt, ahol leadja rendelését. A felvétel a rendelést rögzíti. A feldolgozó osztály az előző nap rögzített rendeléseket szolgálja ki. Amikor Irma rendeléséhez érnek, kiderül, hogy a szemöldökdúsító tabletták elfogyottak, és ezt jelzik a Rendelésfelvételnek, amely felhívja Irmát, és megkérdezi, hogy mi legyen. Irma bánatában törli a rendelést.



minden objektum és üzenet 1 pont, törlés korrekt jelölése 1 pont.

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2002. január 8. 15 óra

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A KULTIPLUS moziban művészfilmeket (cím, rendező, gyártás éve) játszanak. Egy mozielőadásról nyilvántartják, hogy melyik filmet játszották és mennyi volt az összes bevétel. A teremben a sorok és a székek számozottak, a jegy ára a szék helyétől is függ. A jegy egy főt egy előadáson egy szék elfoglalására jogosít. A jegyen megtalálható még az eladás ideje és a pénztáros kódja. A jegyek vásárlásakor számla is készül, amely tartalmazza, hogy a vásárló hány jegyet vett és összesen mennyit fizetett. Telefonon lehet helyet foglalni. Az előjegyzésekről nyilvántartjuk, hogy meddig tartjuk fenn a helyet, melyik pénztáros vette fel a foglalást és mi az előjegyző neve (passwordje).

FILM (cím, rendező, gyártásiév)

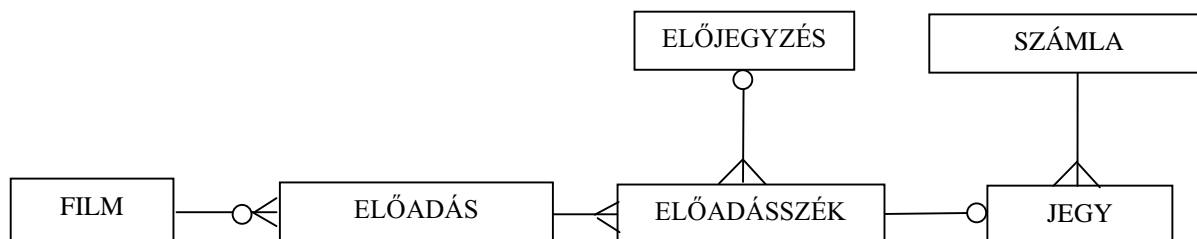
ELŐADÁS (*film*, dátumidő, összbevétel)

ELŐADÁSSZÉK (előadás, sor, ülés, állapot, előjegyzés)

ELŐJEGYZÉS (password, lejáratidő, pénztáros)

JEGY (előadásszék, eladásidő, számla, pénztáros)

SZÁMLA (sorszám, jegyekszáma, összeg)



mindegyik entitás és mindegyik reláció 0.5 pontot ér. 3 pont fölött +0.5 pont.

2. Sorolja fel a komponensek közötti kommunikáció módjait ! (3 pont)

- eljárás hívás,
- visszahívás (callback),
- távoli eljáráshívás (remote procedure call, rpc)
- közös memória alkalmazása
- FIFO, pipe, socket
- adattáblák, strukturált adattáblák (blackboards, structured blackboards)
- tranzakciós szolgáltatások (pl. DBMS)
- kód migráció (pl. Applet)

(minden jó válasz 0.5 pont, max 3 pont)

3. Jellemezze a Blackboard (tábla) szoftver architektúrát ! Adja meg a komponenseket, az architektúra alkalmazásának előnyeit és hátrányait !(5 pont)

Komponensek:

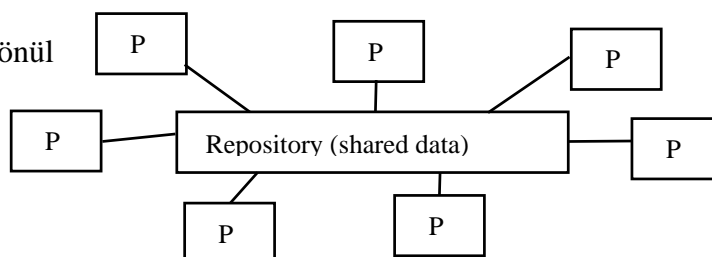
- szeparált, független programok, alrendszerek, amelyek közös memórián keresztül működnek együtt
- központi adattár (adatbázis)

Előnyök:

- az adatok, a vezérlés és a feldolgozás jól elkülönül
- kisebb adathibák tűrése és korrekciója

Hátrányok:

- bonyolult tesztelni
- alacsony hatékonyság
- nagy fejlesztési erőfeszítést igényel

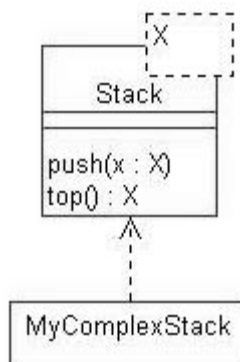


Komponensek 1-1 pont, 1 előny 1 pont, max 1 pont, 2 hátrány 1 pont, max 2 pont

4. Legyen egy *Stack* paraméteres osztályunk, amelybe a paraméternek (*X*) megfelelő elemek tehetők. Legyen a stacken egy *push* művelet a szokásos jelentéssel, és egy *top* művelet, ami visszaadja a stack tetején található értékeket. A *Stack* template felhasználásával elkészítjük a *MyComplexStack* osztályt. Rajzoljon **UML szerkezeti diagramot** ! (3 pont)

Stack template korrekt 2 pont

MyComplexStack Stack kapcsolat 1 pont



5. Adja meg a Rational Unified Process három fő jellemző tulajdonságát ! (6 pont)

- use-case vezérelt
- architektúra központú
- iteratív és inkrementális

mindegyik jellemző 2 pont

6 Adjon **JSD** alapú megoldást az alábbi problémára ! (10 pont)

A ROKFORT-i varázslóiskolába január 7.-én éjfélig lehet az aktuális szemeszterre jelentkezni. A jelentkezők eldönthetik, hogy igényelnek-e konzultációt, vagy östehetségnek tartják magukat. A konzultációt igénylők felkészítő feladatokat kérhetnek (disznófarok növesztés, békává változtatás, stb.), melyet otthon gyakorolhatnak, és a konzultáció vezetőjének bemutathatnak. Új feladatot csak akkor kapnak, ha az előzőt már bemutatták. A kis önbizalommal rendelkezők a konzultációs időszak alatt bármikor bejártott varázspálcát kérhetnek, amelyet önbizalmuk növekedésekor (de legkésőbb a konzultációs időszak végén) vissza kell adniuk. Ha valaki visszaadta a pálcát, újat már nem igényelhet. A konzultációs időszak egy eltűntető varázslat bemutatásával zárul. A szemeszter végén aki elég erősnek érzi magát, jelentkezhet a vizsgára, ahol a legöregebb varázslókból álló bizottság előtt legalább egy varázslást kell bemutatni. A vizsgázott varázslótanonc oklevelet kap.

Definiálja az alábbi funkciókat ellátó processzeket:

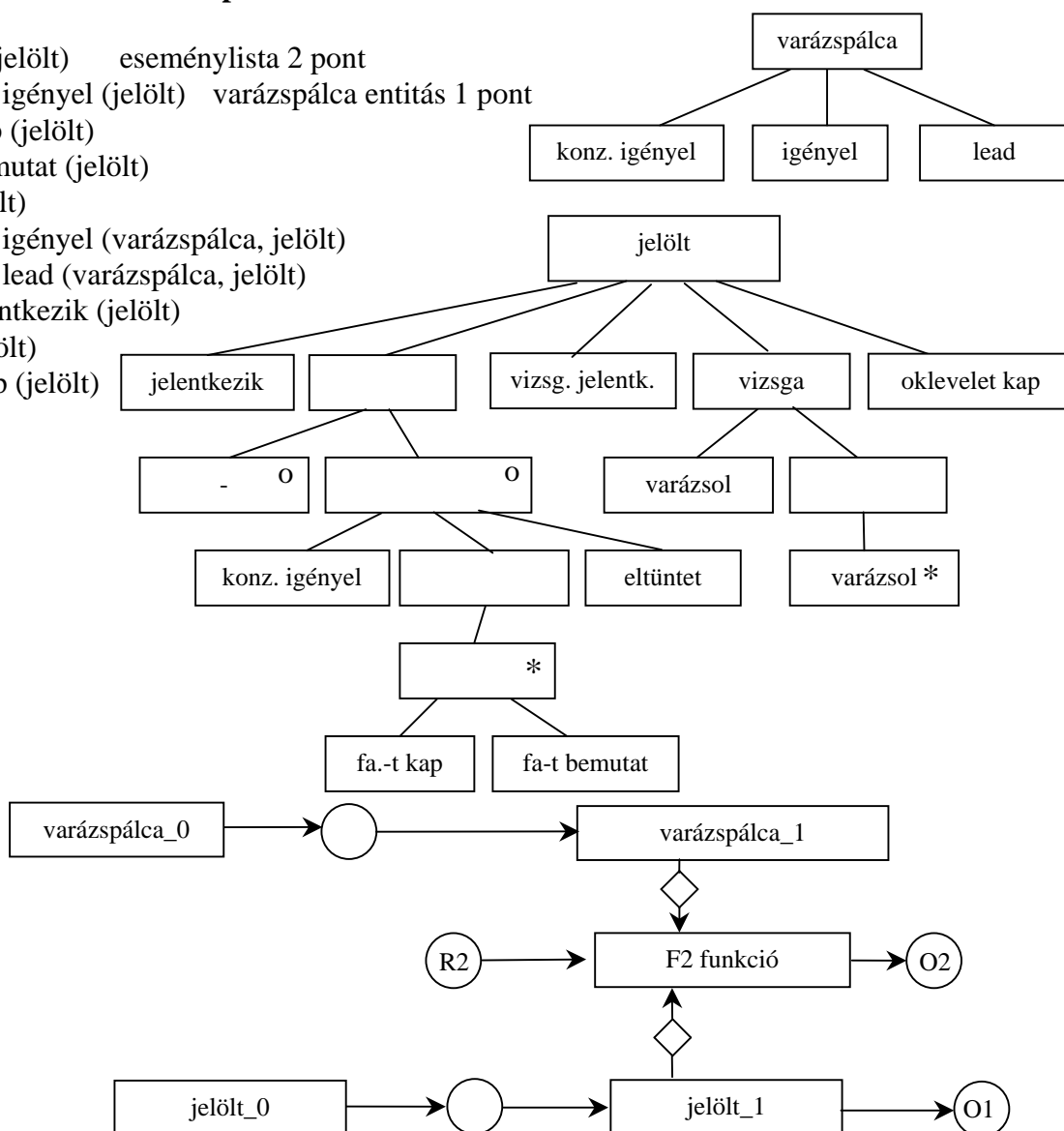
Jelezzon, ha valaki eltűntető varázslatot mutat be úgy, hogy négynél kevesebb felkészítő feladatot oldott !
Kérésre listázza ki, hogy milyen varázslato(ka)t mutattak be a vizsgán azok, akik a konzultációs időszakban bejártott varázspálcát használták !

Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modelt, a funkció processzeket.**

kapcsolja a modellhez a

- jelentkezik (jelölt) eseménylista 2 pont
- konzultációt igényel (jelölt) varázspálcá entitás 1 pont
- feladatot kap (jelölt)
- feladatot bemutat (jelölt)
- eltűntet (jelölt)
- varázspálcát igényel (varázspálcá, jelölt)
- varázspálcát lead (varázspálcá, jelölt)
- vizsgára jelentkezik (jelölt)
- varázsol (jelölt)
- oklevelet kap (jelölt)

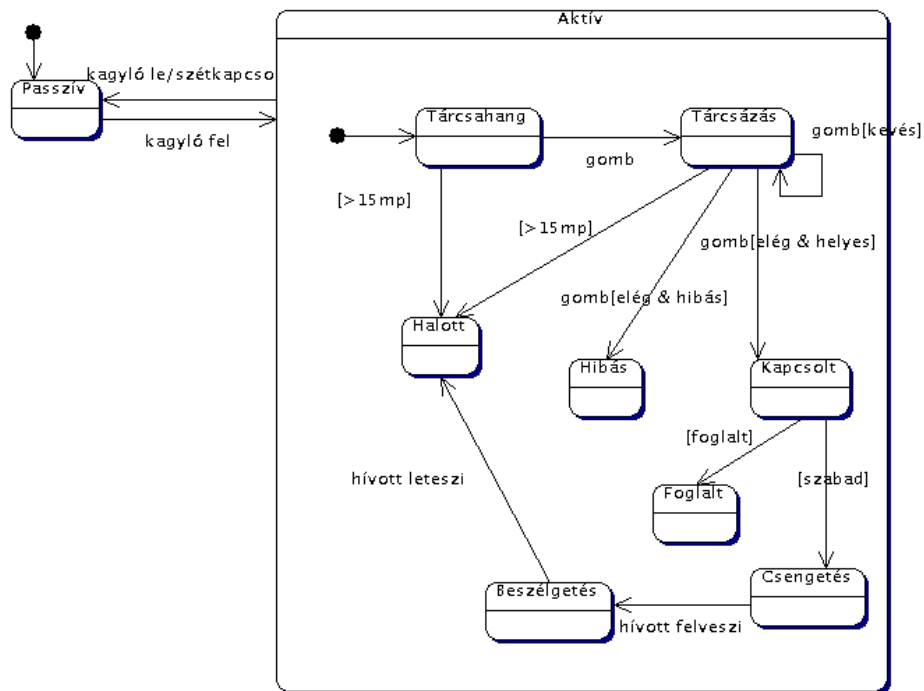
jelölt entitás 5 pont



teljes SSD 2 pont

7. Rajzoljon **UML statechartot** az alábbi feladat leírására ! (7 pont)

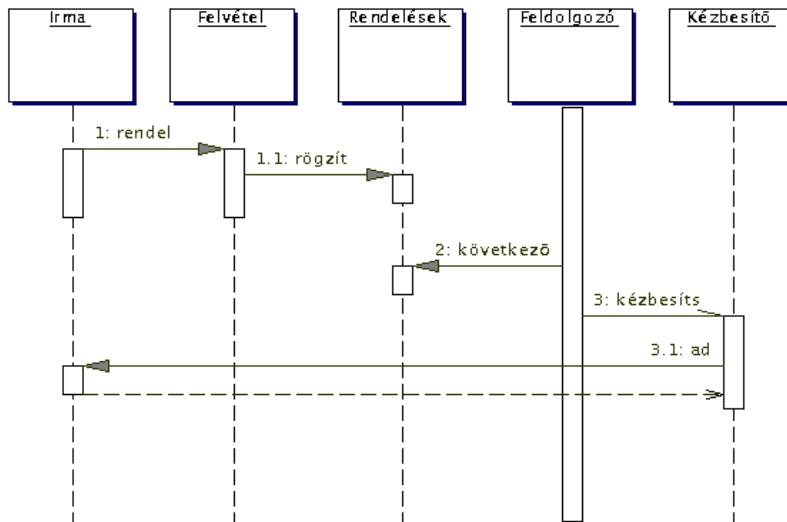
Egy telefon alapállapotban passzív. Ha megemeljük a kagylót, akkor aktív állapotba kerül. Az aktív állapotban először mindig tárcsahangot kapunk. Ha tárcsázunk, akkor a tárcsahang megszűnik. Ha két számjegy leütése között, vagy a kagyló felemelése és a tárcsázás elkezdése előtt több, mint 15 mp telik el, akkor a hívás felfüggesztődik, és foglalt jelzést kapunk. A tárcsázás végét a kellő mennyiségű számjegy jelzi. Hívhatunk hibás, illetve jó számot. Előző esetben a kagylóból a hibás számot jelző szignált halljuk. Utóbbi esetben a készülék kapcsol, és vagy foglalt jelzést ad, vagy kicseng. Ha kicseng, és a hívott fél felveszi a kagylót, beszélhetünk. Ha a hívott fél leteszi, akkor a hívás törlődik, és foglalt jelzést kapunk. Ha aktív állapotban akármikor letesszük a kagylót, a telefon passzív lesz.



2 fő állapot és skatulyázás 1 pont
minden él 0.5 pont

8. Rajzoljon az eseményekről **UML szekvencia-diagramot** ! (10 pont)

Irma szemöldökdúsító tablettákat szeretne. Felhívja a Kwelle Rendelésfelvételt, ahol leadja rendelését. A Felvétel a rendelést rögzíti. A Feldolgozó osztály az előző nap rögzített rendeléseket szolgálja ki. Amikor Irma rendeléséhez érnek, a Kézbesítőnek adják a tablettákat tartalmazó dobozt, és elküldik Irmához. A kézbesítő Irmának átadja a dobozt, viszonzásként a vételárat, és 2 Ft borralalót kap.



minden objektum és üzenet 1 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK

PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2000 december 19.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet ! (3 pont)

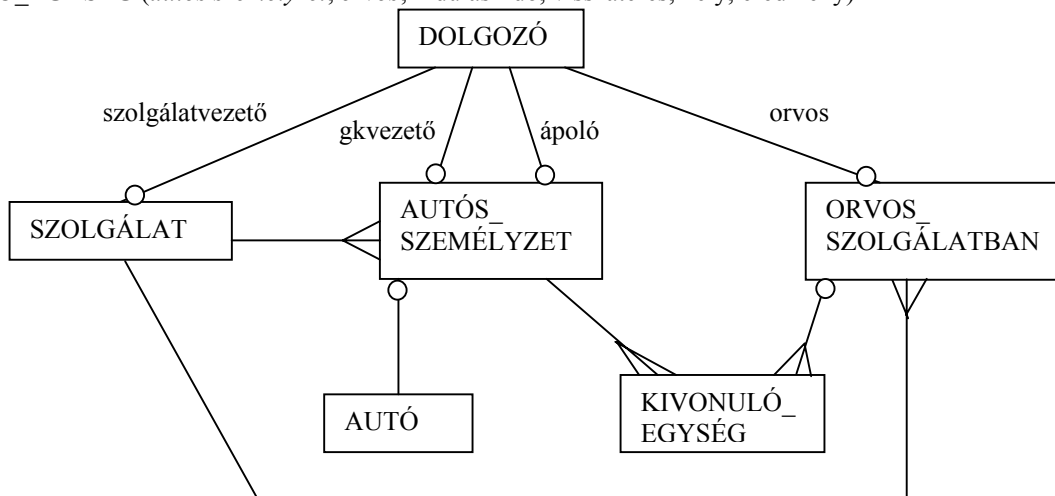
CRT()	új (üres) stringet hoz létre.
LGTH(s)	az s string karaktereinek számát adja.
ALL(s)	hamis, ha az s string üres vagy benne nem mindegyik karakter egyforma.
ADD(s,x)	az s string végére rakja az x karaktert.
HEAD(s)	az s string első karakterét mutatja meg.

LGTH(CRT()) = 0	-0.5 pont, ha hiányzik
LGTH(ADD(s, c)) = LGTH(s) + 1	0.5 pont
ALL(CRT()) = false	0.5 pont
ALL(ADD(s, c)) = LGTH(s) == 0 ALL(s) && HEAD(s) == c	1.5 pont
HEAD(CRT()) = undefined	-0.5 pont, ha hiányzik
HEAD(ADD(s, c)) == if (LGTH(s) == 0) c else HEAD(s)	0.5 pont

2. Készítsen entitás-relációs diagramot az alábbi problémára. (7 pont)

A Halljakend utcai mentőállomás dolgozóit a nevükkel, szakképzettségükkel és a szolgálati idővel jellemezzük. A mentőautókról nyilvántartjuk a rendszámot, a motorszámot és a korát. Az állomáson 12 óras szolgálatba szervezve folyik a mentés. Egy szolgálatot a szolgálatvezető irányít. A szolgálatra vezényelt autónál - amelynek személyzetét egy gépkocsivezető és egy ápoló alkotja - rögzítjük a szolgálatba lépés pillanatában esedékes kilométeróra állást. A mentőorvosokat ugyancsak a szolgálati beosztás vezényli. Valamennyi szolgálatba lépő dolgozónak megméri és rögzíti a vérnyomását. A balesethez kivonuló egységet az autó és személyzete alkotja, de ez kiegészülhet - a szolgálatvezető esetenkénti döntése alapján - a mentőorvossal. Rögzítjük a kivonulás és a visszatérés idejét, a kivonulás helyét, és annak eredményét.

SZOLGÁLAT (dátum, napszak, *dolgozó(szolgálatvezető)*, *szolg-vez_vérnyomás*)
 AUTÓS_SZEMÉLYZET (*dolgozó(gkvez)*, *gkvez_vérnyomás*, *dolgozó(ápoló)*, *ápoló_vérnyomás*, *szolgálat*, *autó*, kilométer)
 DOLGOZÓ (név, szakképzettség, szolgálati idő)
 AUTÓ (rendszám, motorszám, kor)
 ORVOS_SZOLGÁLATBAN (*dolgozó(orvos)*, *orvos_vérnyomás*, *szolgálat*)
 KIVONULÓ_EGYSÉG (*autós személyzet*, *orvos*, indulási idő, visszatérés, hely, eredmény)

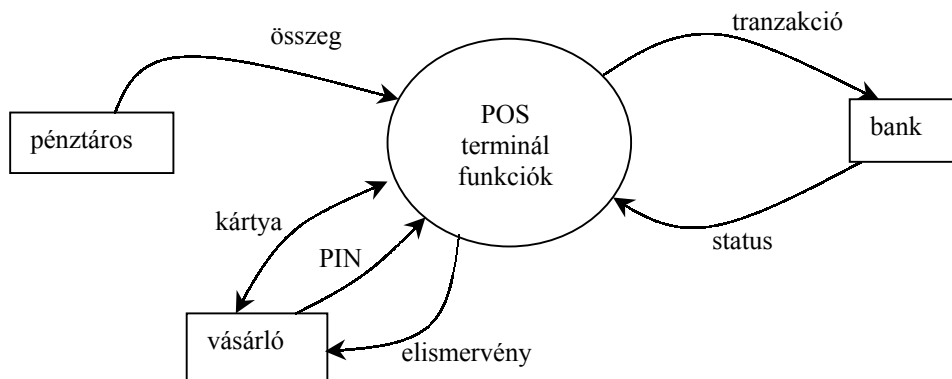


Pontozás: mindegyik entitás és reláció 0.5 pontot ér, max. 7 pont (15 elem van !!!)

3. Készítse el egy POS terminál adatfolyamábráját ! Adja meg a context diagramot és annak 0 szintű felbontását ! (6 pont)

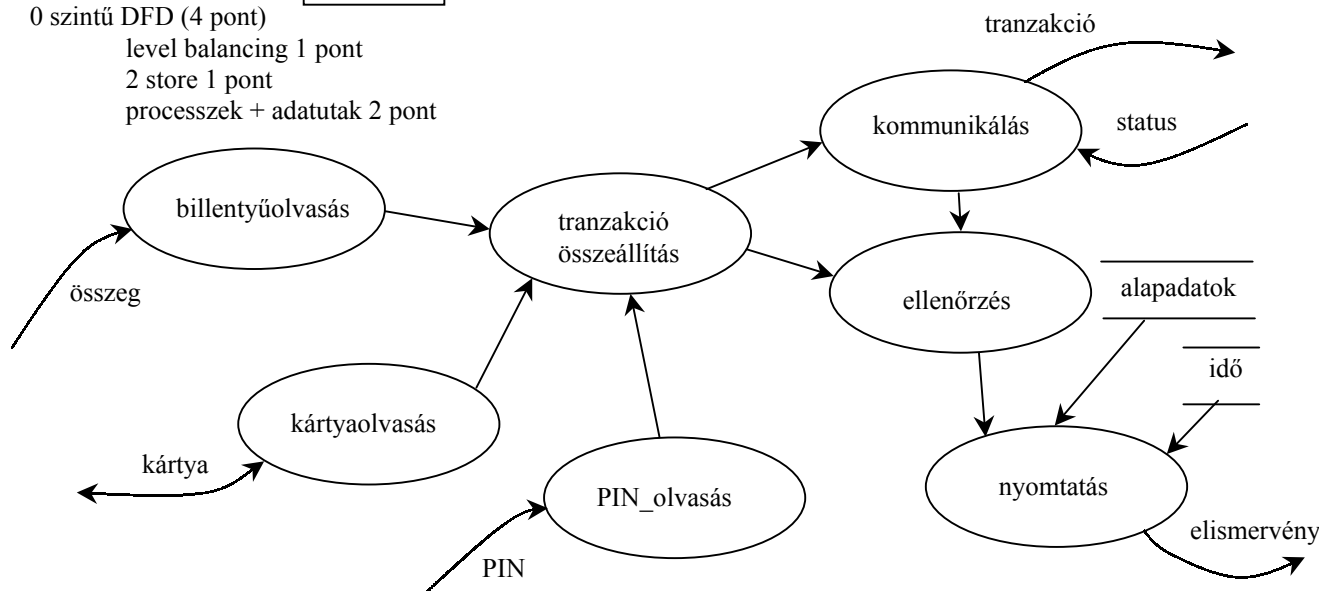
A POS (Point of Sale) terminált a bankkártya elfogadó helyeken használják. A pénztáros a billentyűzeten keresztül megadja a vásárlás összegét (a kártya terhelését), majd a kártyát az olvasón végighúzva leolvassa a kártya információkat. Bizonyos esetekben a vásárlónak a külön billentyűzeten meg kell adni a PIN kódját. A terminál telefonvonalon keresztül rákapcsolódik a bank számítógépére, ahonnan a tranzakciót elfogadják vagy elutasítják. Végezetül a terminál elismervényt nyomtat, amin szerepel a vásárlás helye, ideje, a kártya száma, a vásárlás összege, a terminál azonosítója, a banki jóváhagyás kódja.

Context diagram (2 pont) (3 terminátor 1 pont, 6 adat 1 pont)



0 szintű DFD (4 pont)

level balancing 1 pont
2 store 1 pont
processzek + adatutak 2 pont



4. Adjon JSD alapú megoldást az alábbi problémára ! (7 pont)

A Nyúlzavarógépgyár igazgatótanácsa (8 tag + elnök) a főnyúlzavarói állás betöltésére pályázatot ír ki. A kiírás szerint a jelentkezőnek fényképes, legalább 5 oldalas szakmai önéletrajzt kell benyújtani. A beérkezett pályázatokat szintaktikailag ellenőrzik (van-e fénykép és megvan-e az öt oldal), majd a megfelelők mindegyikét az igazgatótanács három tagja véleményezi. Egy igazgatósági tag akkor kap újabb pályázatot véleményezni, ha az előzőleg kapott anyagról már a véleményt elkészítette. Számítani lehet rá, hogy minden vélemény (előbb-utóbb) elkészül. Az összes vélemény elkészülte után összeül a teljes igazgatótanács és kiválasztja a 3 legjobb ajánlatot, akiknek benyújtóival személyesen is elbeszélgetnek. A beszélgetés alapján döntenek.

Definiálja az alábbi funkciókat ellátó processzeket:

Kérésre adja meg azon pályázókat, akiknek anyaga pontosan két igazgatósági tagnál van éppen bírálaton. !
Kérésre adja meg azokat az igazgatósági tagokat, akiknél éppen nincs bírálendő pályázat !

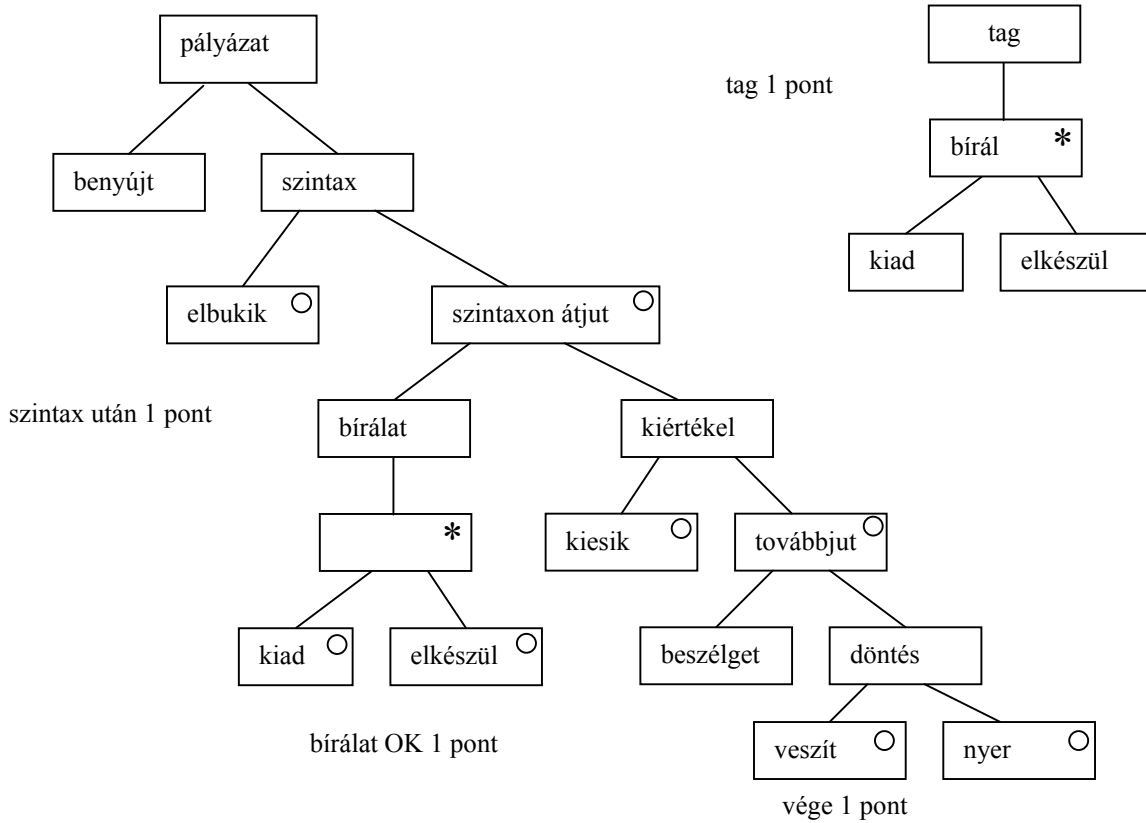
Készítse el az

**entitás/akció leírást, az entitások attribútumaival,
az entitás struktúra diagramot,
az initial modellt,
funkció processzeket.**

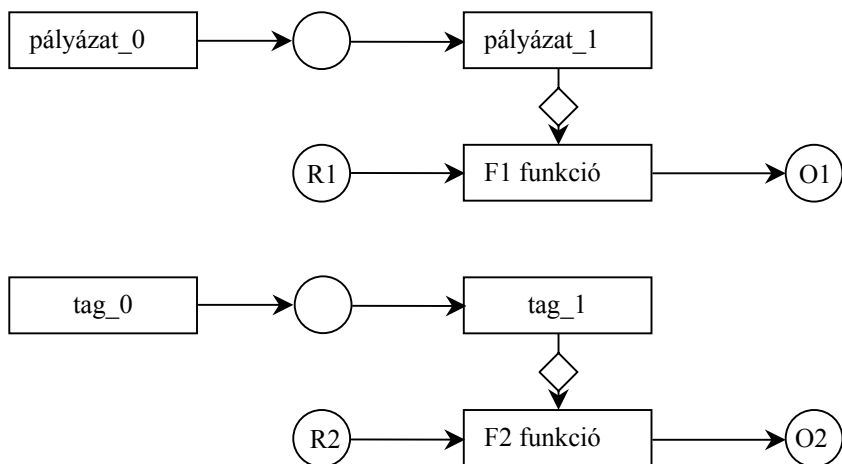
kapcsolja a modellhez a

- benyújt (pályázat) (pályázó, időpont)
- szintax ellenőrzés (pályázat) (eredmény)
- véleményezésre megy (pályázat, tag) (időpont)
- véleményezés kész (pályázat, tag) (eredmény, időpont)
- véleményt kiértékel (pályázat) (eredmény)
- beszélget (pályázat) (pályázó, időpont, tagok)
- döntés (pályázat) (eredmény)

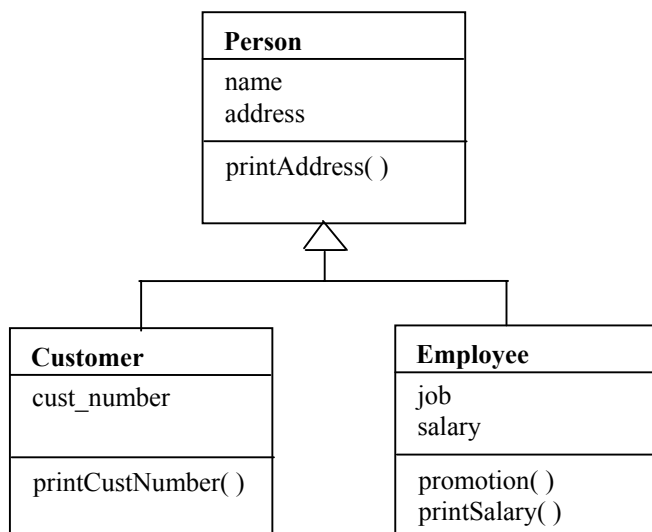
entitás/akció leírás 2 pont



SSD 1 pont



5. Legyenek a következő osztályaink.



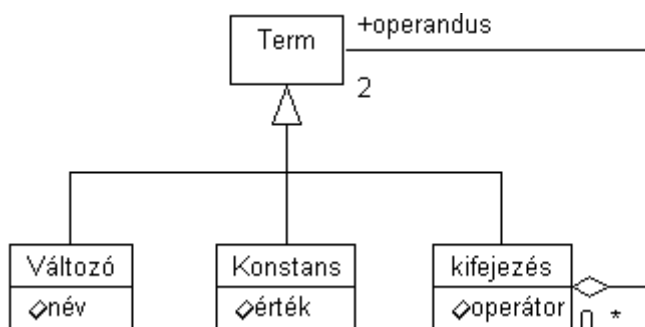
Adja meg az igaz állításokat (2 pont):

- | | |
|---|---|
| b) Employee helyettesíthető Personnal | p) Person helyettesíthető Employeevel |
| g) Person helyettesíthető Customerrel | d) Employee helyettesíthető Customerrel |
| t) Customer helyettesíthető Employeevel | v) Customer helyettesíthető Personnal |

igaz g és p

pontozás találat 1 pont, mellé -1 pont

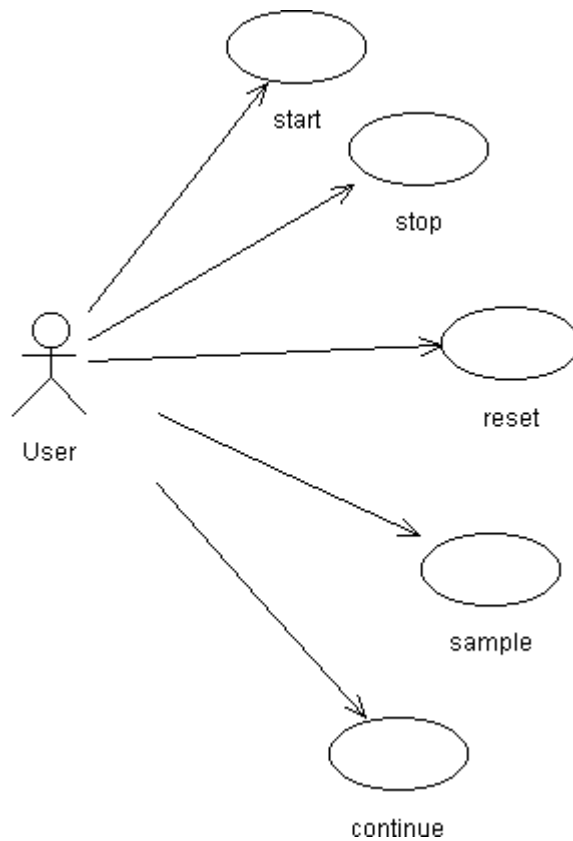
6. Egy matematikai term lehet változó, konstans vagy kifejezés. A változót megadhatjuk a nevével, a konstanst az értékével. Egy kifejezés pontosan két operandust és egy bináris operátort tartalmaz. Operandus lehet bármely term. Rajzoljon osztálydiagramot (5 pont)!



öröklés 2 pont, rekurzió 3 pont (ha multiplicitás is jó)

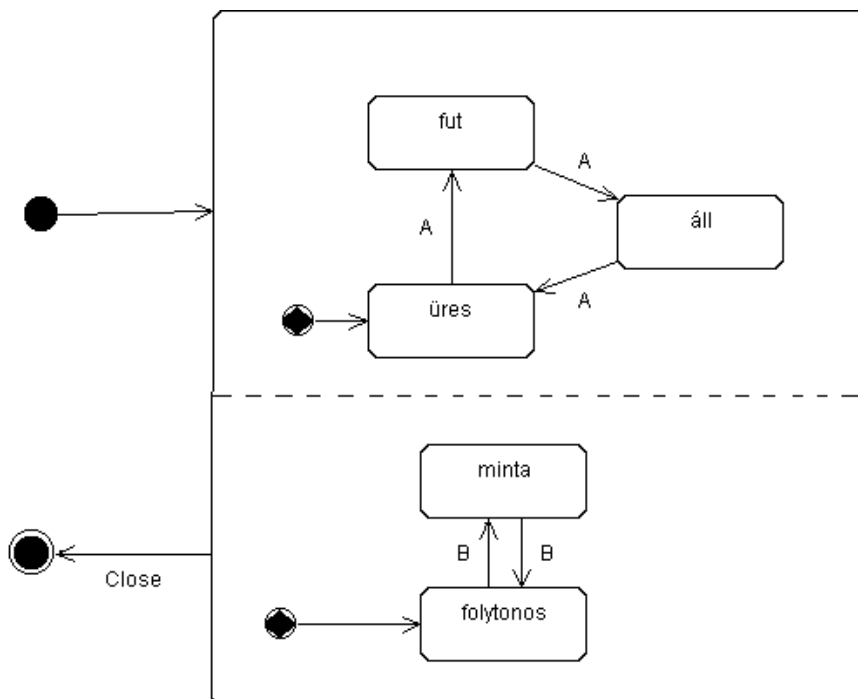
7. Egy stopperórának egy időkijelzője és két gombja (A és B gomb) van. Az A gombot megnyomva indítható az időmérés. A futó idő ugyancsak az A gomb megnyomásával állítható meg. Az A gomb ismételt megnyomásával az idő 0-ra állítható. A B gombbal a kijelzés vezérelhető. Alapállapotban a kijelzőn a futó idő látható. A B gomb megnyomásakor a futó időből mintát veszünk és azt jelenítjük meg, miközben az időmérés folytatódik. Így lehet részidőt mérni. A B gomb következő megnyomásakor visszaáll az alaphelyzet és a futó idő látható.

Rajzoljon use-case diagramot az óra kezelésére. (2 pont)



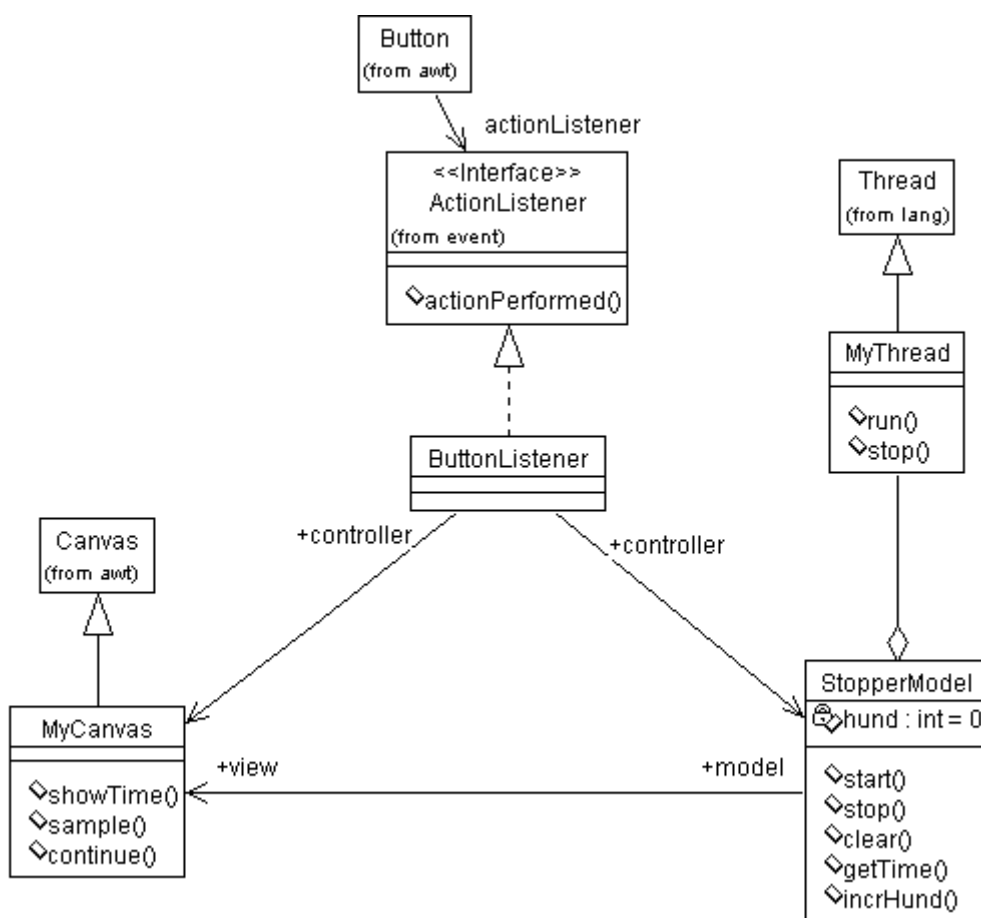
összesen 2 pont

Rajzolja fel a stopperóra viselkedését leíró UML statechart-ot. (5 pont)



A régiók külön-külön 2 –2 pont. Konkurencia felismerése 1 pont. Összekeverve max. 4 pont.

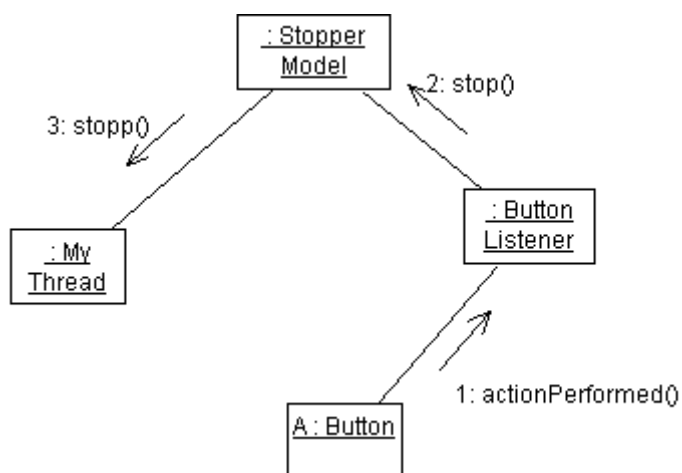
Rajzolja fel a Java awt-re épülő alkalmazás osztálydiagramját. Tüntesse fel az operációkat. (8 pont)



Model – view – controller felismerése és ábrázolása (metódusok nélkül) 1.5 –1.5 – 1.5 pont

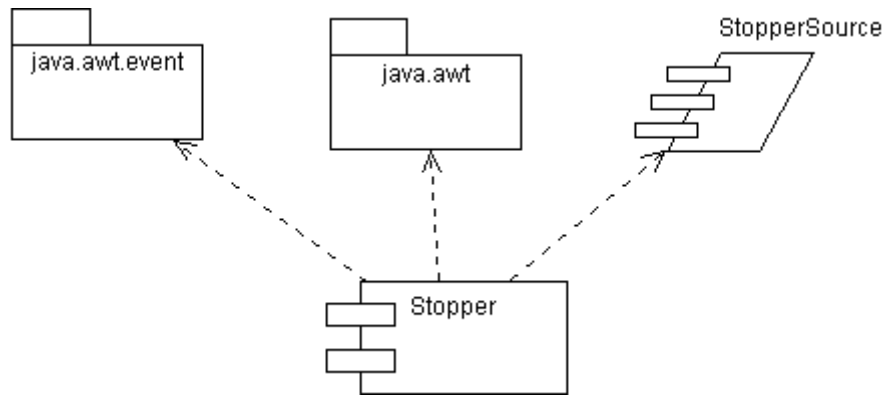
Metódusok modelnél és viewnál 2 – 1.5 pont

Rajzolja fel a kollaborációs diagramot arra az esetre, amikor az A gomb megnyomásával leállítja a futó órát. (3 pont)



minden objektum és üzenet 0,5 pont

Rajzolja fel a komponens diagramot. (2 pont)



minden jó elem (komponens és kapcsolat) 0.5 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

AAAAAAAAAAAAAAAAAAAA

2001. január 9.

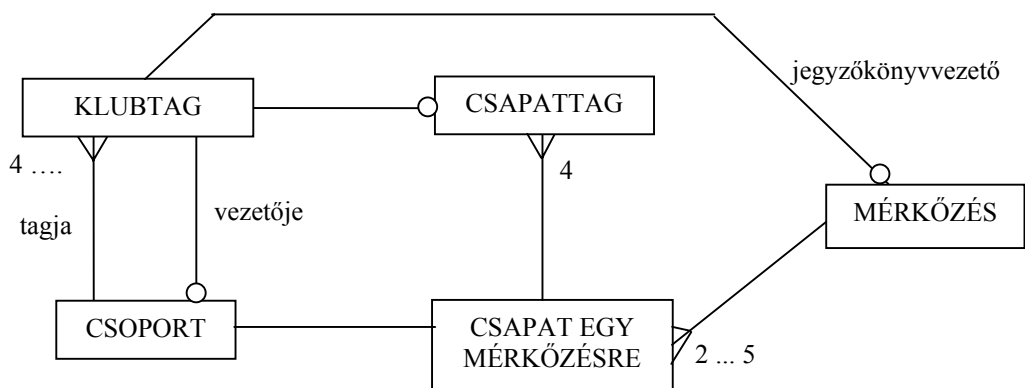
1. Algebrai axiómák segítségével specifikálja az egész számoknak az alábbi műveletekkel jellemzett halmazát !
(4 pont) Az axiómákban alkalmazhat egy $\text{odd}(x) \rightarrow \text{boolean}$ függvényt, amely igaz, ha az x egész páratlan.

NEW() új (üres) halmazt hoz létre.
LGTH(h) a **h** halmaz elemeinek számát adja.
EVEN(h) a **h** halmaznak azon részhalmaza, amely csak a páros elemeket tartalmazza.
ADD(h,i) a **h** halmazhoz hozzáveszi az **i** egészet.
ISIN(h, i) igaz, ha **i** a **h** halmaz eleme.

LGTH(NEW()) = 0	- 0.5 pont, ha hiányzik
ISIN(NEW(),x) = false	- 0.5 pont, ha hiányzik
EVEN(NEW()) = NEW()	- 0.5 pont, ha hiányzik
LGTH(ADD(h, x)) = if (ISIN(h,x)) LGTH(h) else LGTH(h) + 1	1 pont
ISIN(ADD(h, x), y) = (x==y) ISIN(h, y)	1 pont
EVEN(ADD(h, x)) = if (odd(x)) EVEN(h) else ADD(EVEN(h), x)	2 pont

2. Készítsen entitás-relációs diagramot az alábbi problémára !(8 pont)
 A bilgécfalvi tekeklub tagjai (a tagokat névvel, címmel, korról azonosítjuk) szakmák szerinti csoportokat alkotnak (pékek, asztalosok, informatikusok stb.). Minden csoportnak van legalább négy tagja és pontosan egy vezetője. A csoportok egymással versenyeznek. A csoportokat az egy mérkőzésre eső átlagos leütött bábuk (fa) alapján rangsorolják. Holtversenyben figyelembe veszik a csoport által játszott mérkőzések számát. A csoport egy mérkőzésre a saját tagjaiból választ ki egy négytagú csapatot. Egy mérkőzésen minimum 2, maximum 5 csapat küzd egymással. Nyilvántartjuk a mérkőzés idejét, a jegyzőkönyvvezetőt (aki egy a mérkőzésen részt nem vevő csoport tagja), a nézőszámot, és természetesen a küzdő csapatokat az elért helyezésekkel és a tagonként leütött fák számával.

KLUBTAG (név, cím, kor, csoport)
CSOPORT (foglalkozás, összesfa, mérkőzésszám, klubtag(vezető))
MÉRKŐZÉS (idő, klubtag(jegyzőkönyvvezető), nézőszám)
CSAPAT EGY MÉRKŐZÉSRE (csoport, mérkőzés, csapatfa, helyezés)
CSAPATTAG (klubtag, csapat egy mérkőzésre, egyénifa)

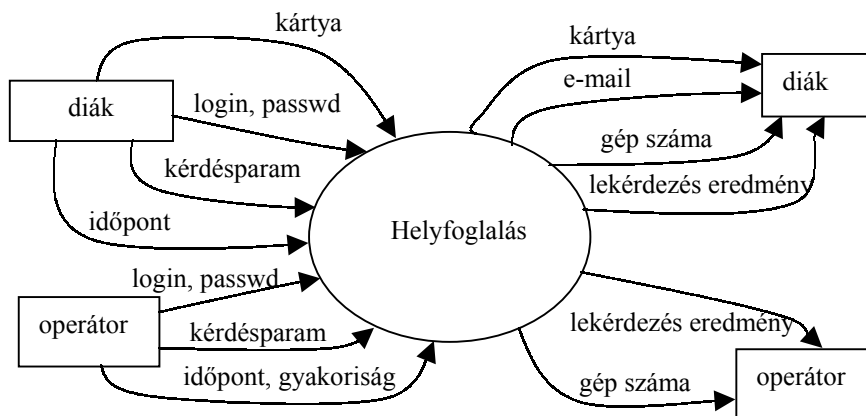


Pontozás: mindegyik entitás 1 pontot, mindegyik reláció 0.5 pontot ér, max. 8 pont (8.5 pont van benne !!!)

3. Készítse el egy számítógépes labor helyfoglaló rendszerének adatfolyamábráját ! Adja meg a context diagramot és annak 0 szintű felbontását ! (7 pont)

A helyfoglaló rendszer segítségével a hallgatók gépidőt (kvóta = 3 óra/hét) foglalhatnak maguknak. A rendszer használatához szükséges a hallgató kártyája, amit az olvasó leolvas. A bejelentkezőnek az azonosításhoz még meg kell adni a login nevét és a passwordját. A hallgató különböző lekérdezéseket (az általa foglalt idők, szabad gépidő) kezdeményezhet. Ha van még kvótája, annak terhére foglalhat. A sikeres foglalás eredménye a lefoglalt gép száma. Lehetséges a foglalás törlése is. A rendszert kezelő operátoroknak nem kell kártya, őket a login név és a password azonosítja. Ők bárki nevében tudnak foglalni (akár a kvótán felül is) és törölni. Az operátorok feladata az egész labor lefoglalása a tanórák idejére. Ehhez meg kell adni az óra időpontját és a gyakoriságát (pl. hetente, 6 héten keresztül). Ha az operátori foglalások vagy törlések következtében már bejegyzett hallgatói foglaltság elvész, akkor a rendszer e-mailt küld az érintett hallgatónak. Az operátorok is tudnak lekérdezéseket indítani.

Context diagram (2 pont) (2 terminátor 1 pont, összes adat 1 pont, 2 hiba 0.5 pont, 4 hiba 0 pont)

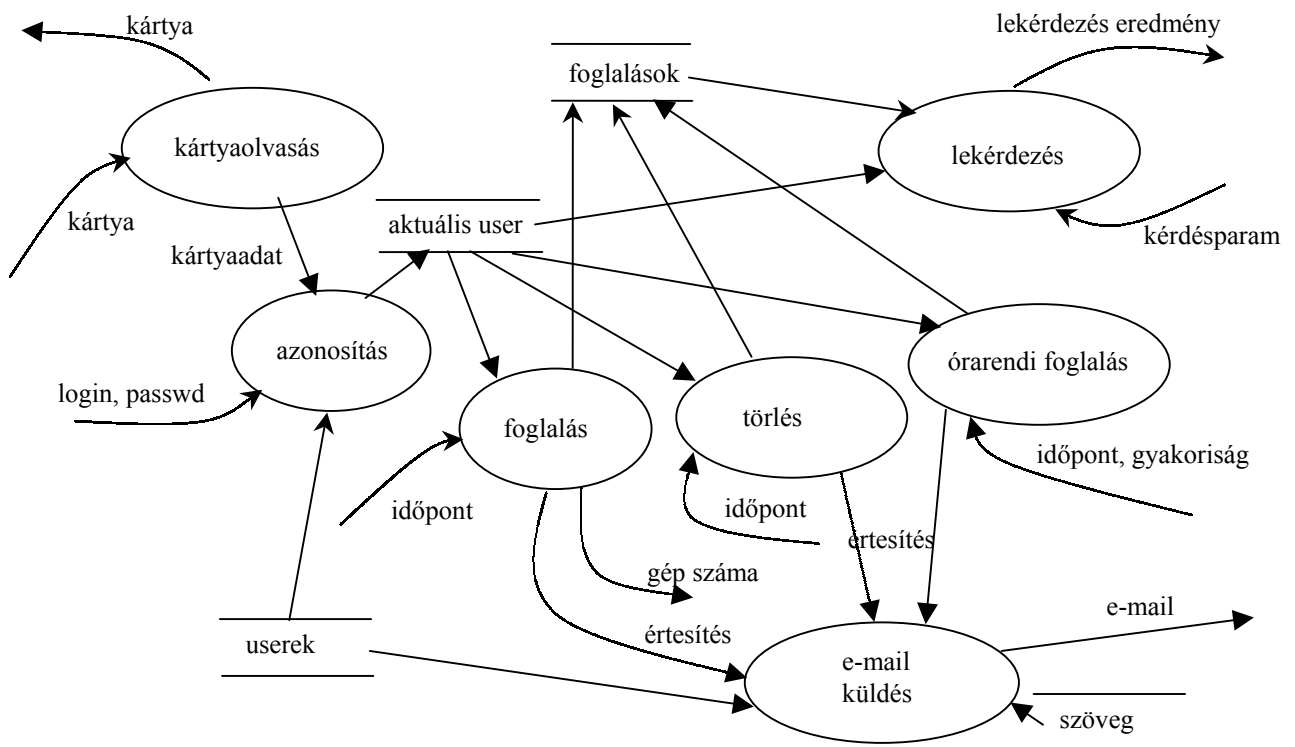


0 szintű DFD (max. 5 pont)

level balancing 1 pont (2 hiba 0.5 pont, 4 hiba 0 pont)

minden store 0.5 pont

processzek 1 pont + adatutak 1 pont



4. Adjon JSD alapú megoldást az alábbi problémára ! (8 pont)

A Lámpahivatal zöldlámpás autók vezetésére jogosító képzést végez. A képzésre felvettek (minden jelentkezőt felvesznek) azonnal megkapják a tanfolyam elméleti anyagát. A zöldlámpa használatára vonatkozó elméleti vizsga letétele után legalább 8 alkalommal kell a forgalomban felügyelet mellett vezetési gyakorlatot végezni, majd ezt követi a forgalmi vizsga. A jogosítvány megszerzéséhez ugyancsak szükséges a szakápolói vizsga, amelyet a hallgató a felvételt követően bármikor letehet. Zöldlámpás jogosítványt az kap, aki a forgalmi és ápolói vizsgát egyaránt letette. Valamennyi sikertelen vizsga korlátlan számban ismételhető. Az ismételt forgalmi vizsgának előfeltétele, hogy legalább 1 alkalommal vezetési gyakorlatot kell végezni.

Definiálja az alábbi funkciókat ellátó processzeket:

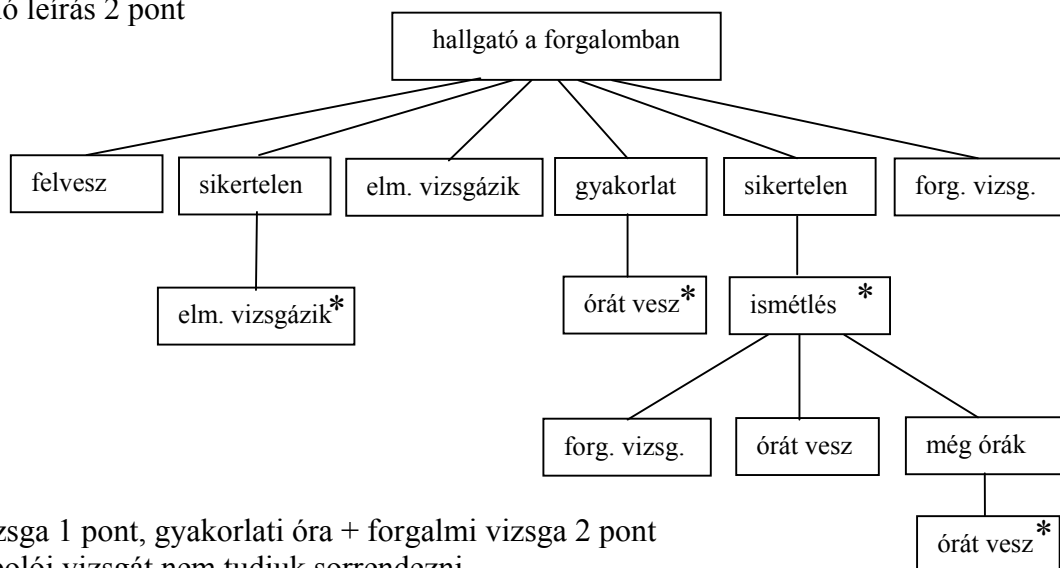
Kérésre adja meg azokat a hallgatókat, akik csak harmadik kísérletre tudták letenni az elméleti vizsgát és kevesebb mint 6 óra forgalmi gyakorlatuk van még !

Kérésre sorolja fel azokat a hallgatókat, akik letették az ápolói vizsgát, az utolsó forgalmi vizsgájuk sikertelen volt, de még nem végezték el a következő vizsgához szükséges minimum 1 órás vezetési gyakorlatot !

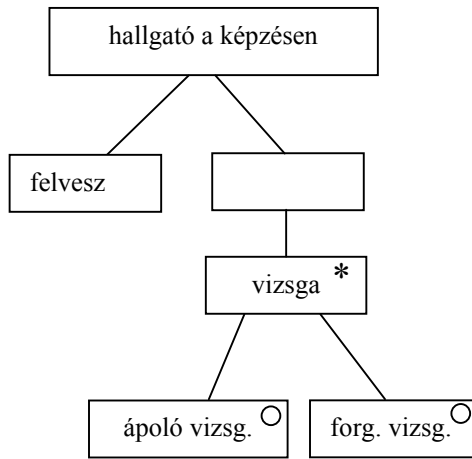
Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modellt, kapcsolja a modellhez a funkció processzeket.**

- felvesz (hallgató) (név)
- elméleti vizsgát tesz (hallgató) (eredmény, időpont)
- forgalmi órát vesz (hallgató) (időpont)
- forgalmi vizsgát tesz (hallgató) (eredmény, időpont)
- ápolói vizsgát tesz (hallgató) (eredmény, időpont)

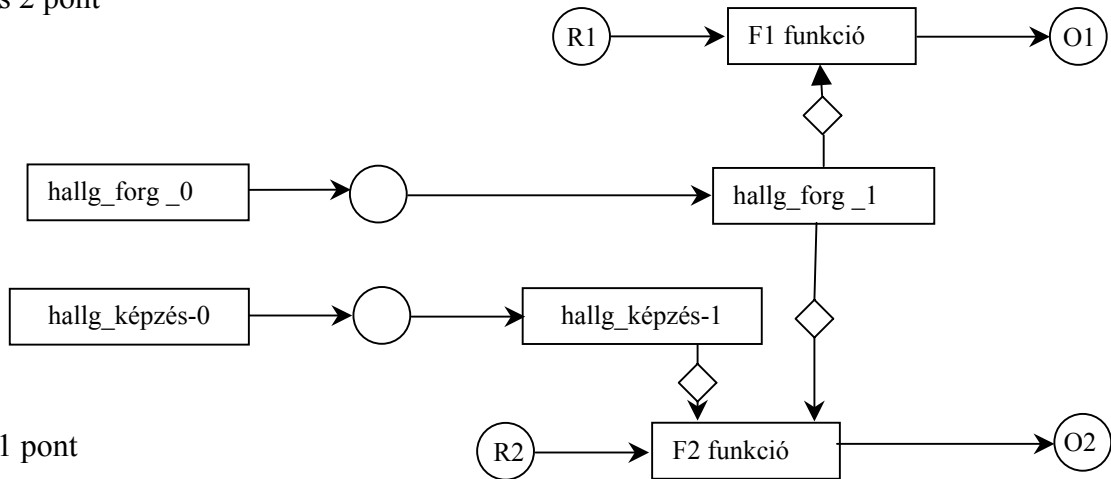
entitás/akció leírás 2 pont



elméleti vizsga 1 pont, gyakorlati óra + forgalmi vizsga 2 pont
gond: az ápolói vizsgát nem tudjuk sorrendezni.



ez az entitás 2 pont



teljes SSD 1 pont

5. Mit jelent az objektum változókon bevezetett **tipizálás** és **kötés** fogalma? Milyen összefüggés van közöttük? (4 pont)

tipizálás: milyen objektumok tehetők a változóba.

(statikus – fordításkor, dinamikus – futáskor definit) (1 pont)

kötés: kihez kapcsolódnak a műveletek (statikus – a változóhoz, dinamikus – az értékhez) (1 pont)
kapcsolat (2 pont):

	<i>stat. tipizálás</i>	<i>din. tipizálás</i>
<i>stat. kötés</i>	nem OO	értelmetlen
<i>din. kötés</i>	normál OO (Java, C++)	Smalltalk, Excel cella

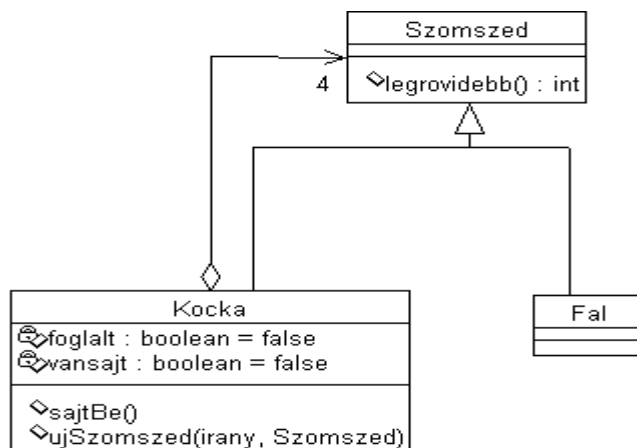
6. Rajzoljon kollaborációs diagramot, arra az esetre, amikor egy user a Rendőr osztályú objektumok egy kollekciónak egy konkrét rendőrt előléptet! (5 pont)

multiobject 2 pont

példány 1 pont

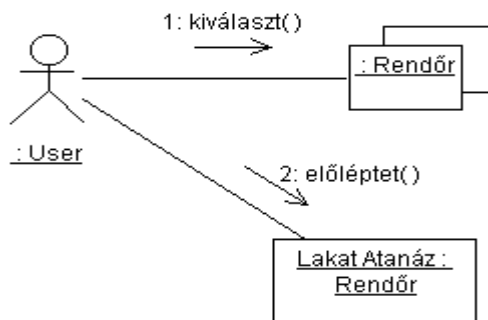
user 1 pont

két üzenet 1 pont



7. Egy labirintus olyan kockákból áll, amelyeknek négy szomszédja van. A szomszéd vagy fal, vagy egy másik kocka. Egy kockához hozzárendelhetjük a szomszédait. A kockába tehetünk sajtot, és minden kockától megkérdezhetjük, hogy milyen hosszú (hány kockából áll) a legrövidebb út a sajtig. Készítsen UML osztálydiagramot és azon adja meg az osztályok metódusainak szignatúráit és az attribútumokat is ! (8 pont)

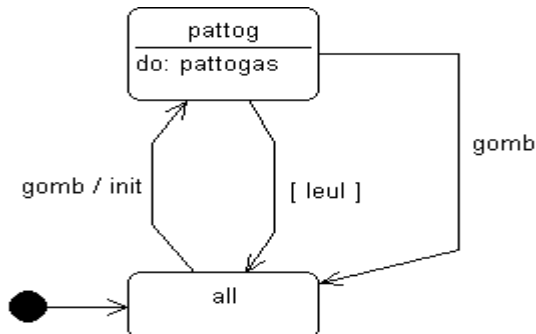
- osztályok 1 - 1pont
- öröklés 1 pont
- aggregáció 1 pont
- 3 metódus 2 pont
- 2 metódus 1.5 pont
- 1 metódus 1 pont
- attribútumok 1 pont



8. A leejtett labda pattogását utánzó program vezérlésére egy gomb szolgál. A gombot megnyomva a labdát az induló helyzetből elengedjük, mire a labda leesik és energiát veszítve a földről visszapattan. A pattogást addig folytatja, amíg az energiája el nem fogy (leül). Ha a labda leült, a gombot megnyomva a pattogás az induló helyzetből ismét elkezdhető. Ha a gombot pattogás közben nyomjuk meg, a labda mozgása megáll és csak a gomb ismételt megnyomásával indítható el az induló helyzetből.

Rajzolja meg a labda viselkedését leíró UML state-chartot ! (6 pont)

- minden állapot és él 1 pont
- do 1 pont



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

BBBBBBBBBBBBBBBBBBBB

2001. január 9.

1. Algebrai axiómák segítségével specifikálja az egész számoknak az alábbi műveletekkel jellemzett halmazát !
(4 pont) Az axiómákban alkalmazhat egy $even(x) \rightarrow boolean$ függvényt, amely igaz, ha az x egész páros.

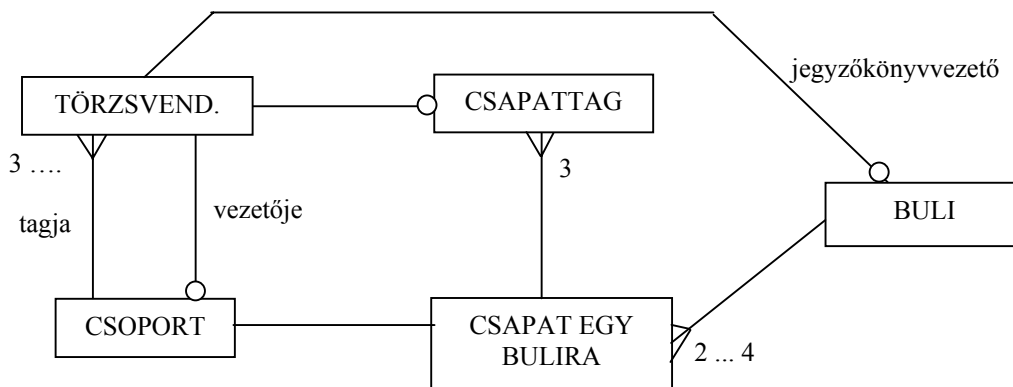
- CRT()** új (üres) halmazt hoz létre.
- ADD(s,q)** az **s** halmazhoz hozzáveszi az **q** egészet.
- LGTH(s)** az **s** halmaz elemeinek számát adja.
- IN(s, i)** igaz, ha **q** az **s** halmaz eleme.
- DELODD(s)** az **s** halmaznak a páratlan elemek kitörlése után maradó részhalmaza.

- LGTH(CRT()) = 0 -0.5 pont, ha hiányzik
- IN(CRT(),x) = false -0.5 pont, ha hiányzik
- DELODD(CRT()) = CRT() -0.5 pont, ha hiányzik
- LGTH(ADD(s, x)) = if (IN(s,x)) LGTH(s) else LGTH(s) + 1 1 pont
- IN(ADD(s, x), y) = (x==y) || IN(s, y) 1 pont
- DELODD(ADD(s, x)) = if (even(x)) ADD(DELODD(s), x) else DELODD(s) 2 pont

2. Készítsen entitás-relációs diagramot az alábbi problémára ! (8 pont)

A bilgécfalvi kocsmá törzsvendégei (a vendégeket névvel, címmel, korról azonosítjuk) szakmák szerinti csoportokat alkotnak (fuvarosok, állattartók, kertészek stb.). Minden csoportnak van legalább három tagja és pontosan egy vezetője. A csoportok egymással sörívásban versenyeznek, időről időre sörbulikát tartanak. A csoportokat az egy bulira eső átlagosan elfogyasztott korsó sörök száma alapján rangsorolják. Holtversenyben figyelembe veszik, hogy a csoport hány bulin vett részt. A csoport egy bulira a saját tagjaiból választ ki egy háromtagú csapatot. Egy bulin minimum 2, maximum 4 csapat mér össze erejét. Nyilvántartjuk a buli idejét, a jegyzőkönyvvezetőt (aki egy a bulin részt nem vevő csoport tagja, nem ihat !), a sör fajtáját, és természetesen a küzdő csapatokat az elért helyezésekkel és a tagonként elfogyasztott korsók számával.

- TÖRZSVENDÉG (név, cím, kor, csoport)
- CSOPORT (foglalkozás, összeskorsó, buliszám, törzsvendég(vezető))
- BULI (idő, törzsvendég(jegyzőkönyvvezető), sörfajta)
- CSAPAT EGY BULIRA (csoport, buli, csapatkorsó, helyezés)
- CSAPATTAG (törzsvendég, csapat egy bulira, egyénikorsó)

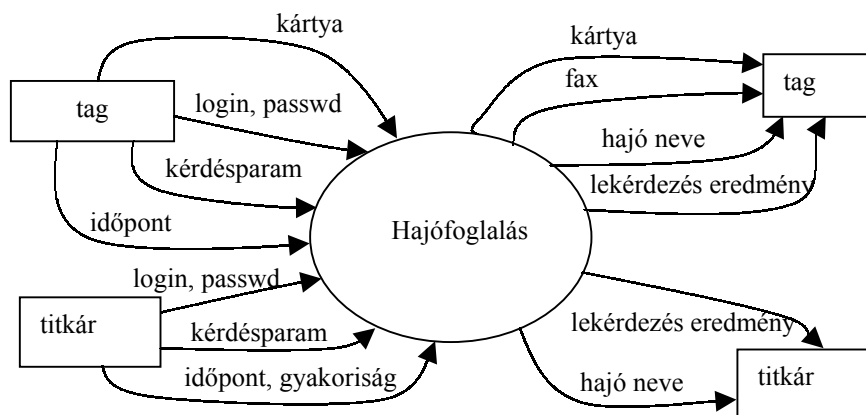


Pontozás: mindegyik entitás 1 pontot, mindegyik reláció 0.5 pontot ér, max. 8 pont (8.5 pont van benne !!!)

3. Készítse el az egyetemi vitorlássegylet hajófoglaló rendszerének adatfolyamábráját ! Adja meg a context diagramot és annak 0 szintű felbontását ! (7 pont)

A hajófoglaló rendszer segítségével a tagok vitorlást (kvóta = havi 7 nap) foglalhatnak maguknak. A rendszer használatához szükséges a tag azonosító kártyája, amit az olvasó leolvas. A bejelentkezőnek az azonosításhoz még meg kell adni a login nevét és a passwordjét. A tag különböző lekérdezéseket (az általa foglalt vitorlások, mikorra van szabad vitorlás) kezdeményezhet. Ha van még kvótája, annak terhére foglalhat. A sikeres foglalás eredménye a hajó neve. Lehetséges a foglalás törlése is. Az egylet titkárainak nem kell kártya, őket a login név és a password azonosítja. Ők bárki nevében tudnak foglalni (akár a kvótán felül is) és törölni. A titkárok feladata az egész hajópark lefoglalása versenyek idejére. Ehhez meg kell adni a verseny időpontját és a gyakoriságát (pl. 6 héten keresztül minden hétfőn). Ha a titkári foglalások vagy törlések következtében már bejegyzett tagi foglaltság elvész, akkor a rendszer faxot küld az érintett tagoknak. A titkárok is tudnak lekérdezéseket indítani.

Context diagram (2 pont) (2 terminátor 1 pont, összes adat 1 pont, 2 hiba 0.5 pont, 4 hiba 0 pont)

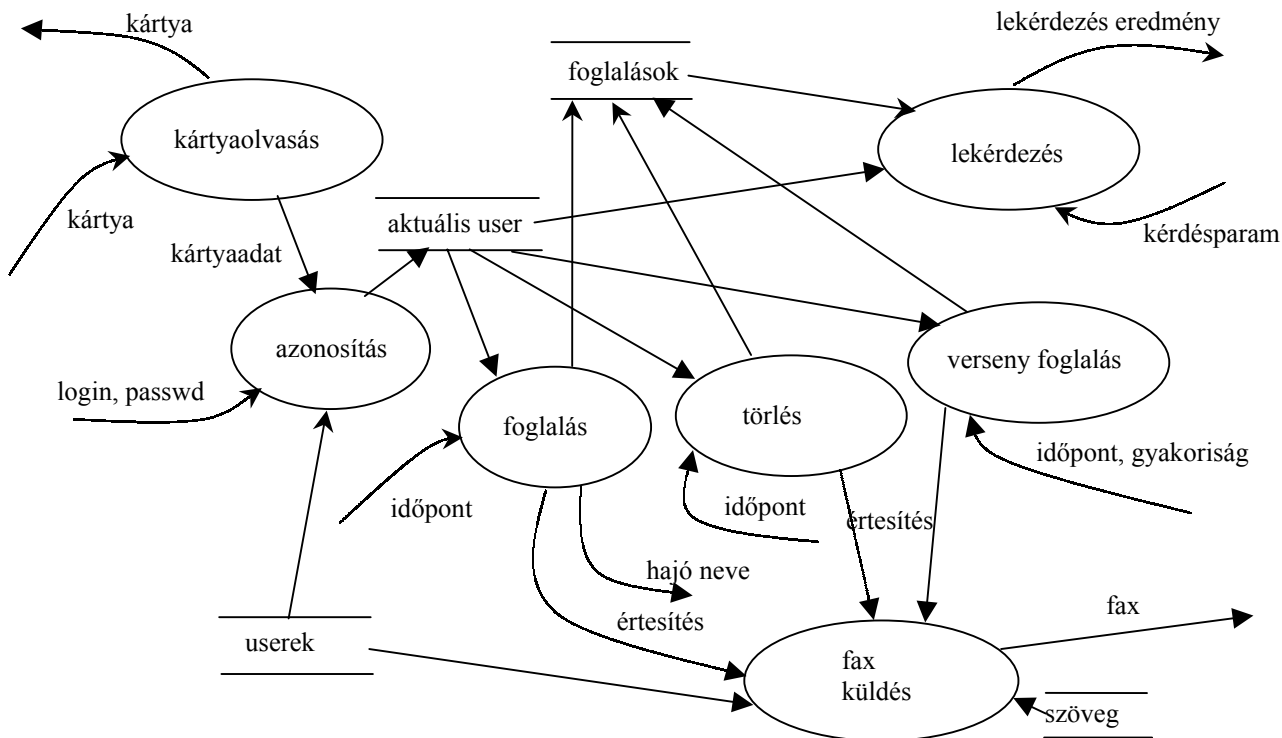


0 szintű DFD (max. 5 pont)

level balancing 1 pont (2 hiba 0.5 pont, 4 hiba 0 pont)

minden store 0.5 pont

processzek 1 pont + adatutak 1 pont



4. Adjon JSD alapú megoldást az alábbi problémára ! (8 pont)

A Hamiskártyás Öregfiúk Köre (HÖK) továbbképzést szervez amatőr kártyásoknak. Mindenkit örömmel fogadnak (nem kell bizonyítani az alapfokú hamiskártyás ismereteket), és a jelentkezés elfogadásaként azonnal átadják a jelentkezőnek a hamiskártyások 12 pontját tartalmazó illusztrált kézikönyvet. Az elméleti vizsga (módszerek és eljárások gyanútlan kártyapartnernek átverésére) letétele után legalább 10 alkalommal profi hamiskártyásokkal gyakorló partikat kell játszani (a profikat a HÖK biztosítja), majd ezt követi a szakma nagymenőivel lebonyolított vizsgaparti. A végbizonyítvány megszerzéséhez ugyancsak szükséges egy erőnléti vizsga (ha lebukunk, gyorsan kell futni), amelyet a továbbképzés ideje alatt bármikor le lehet tenni. Oklevelet az kap, aki a vizsgapartin megfelelt, és az erőnléti vizsgát is letette. Minden sikertelen vizsga korlátlan számban megismételhető. Sikertelen vizsgaparti ismétlésének feltétele, hogy legalább egy alkalommal gyakorló partin kell részt venni.

Definiálja az alábbi funkciókat ellátó processzeket:

Kérésre adja meg azokat a résztvevőket, akik csak a harmadik kísérletre tudták letenni az elméleti vizsgát, és eddig kevesebb, mint 5 gyakorló partit játszottak !

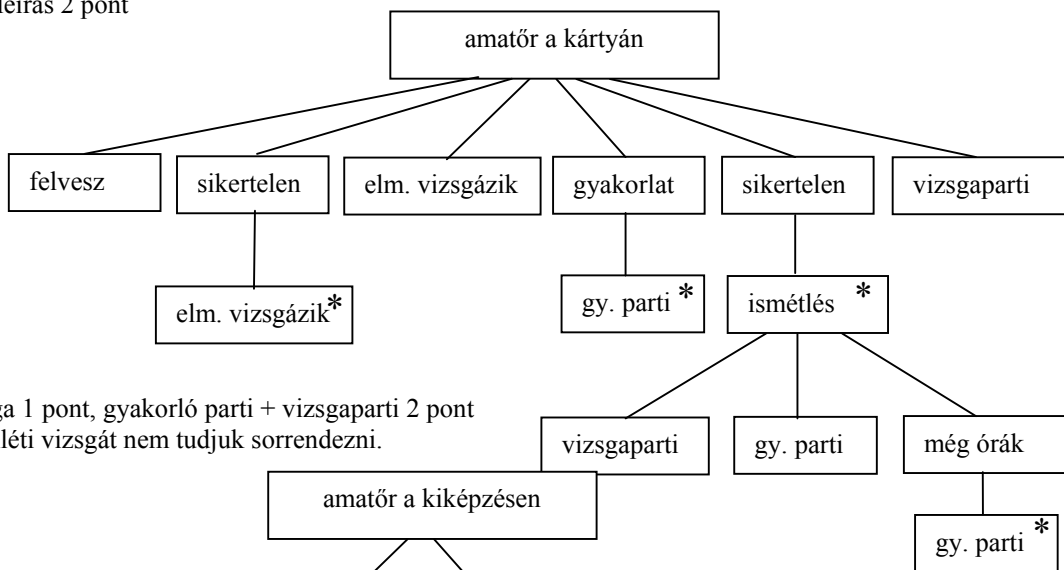
Kérésre sorolja fel azokat a résztvevőket, akik letették az erőnléti vizsgát, a vizsgapartijuk nem sikerült, de még nem vettek részt a következő vizsgapartihoz szükséges gyakorló partin !

Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modelt, a funkció processzeket.**

kapcsolja a modellhez a

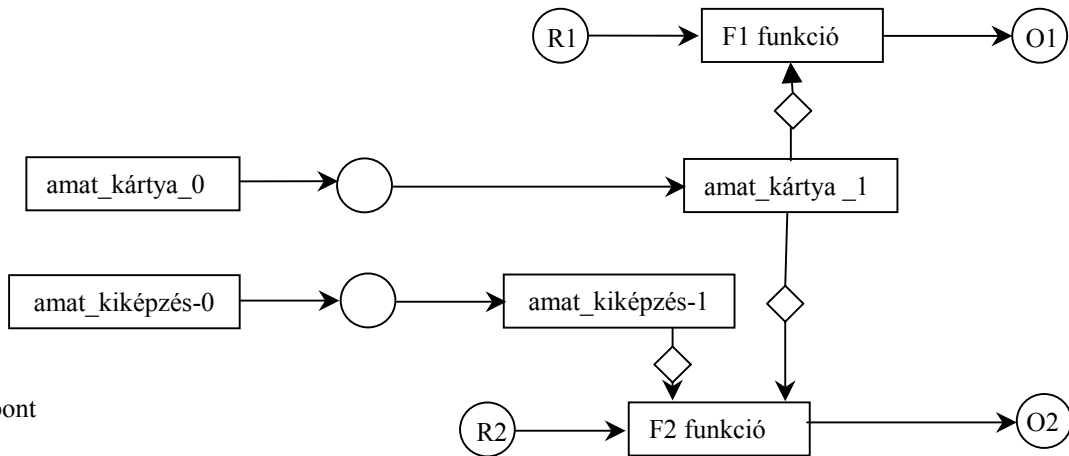
- felvesz (amatőr) (név)
- elméleti vizsgát tesz (amatőr) (eredmény, időpont)
- gyakorló partit játszik (amatőr) (időpont)
- vizsgapartit játszik (amatőr) (eredmény, időpont)
- erőnléti vizsga tesz (amatőr) (eredmény, időpont)

entitás/akció leírás 2 pont



elméleti vizsga 1 pont, gyakorló parti + vizsgaparti 2 pont
gond: az erőnléti vizsgát nem tudjuk sorrendezni.

ez az entitás 2 pont



teljes SSD 1 pont

5. Definiálja az interfész fogalmát ! Mi a kapcsolat az interfész és az osztály között ? Egy leszármazott osztály interfésze milyen kapcsolatban van az ősztyály interfészével ? Hogyan jelöli az UML az interfészt ? (3 pont)

interfész : szignatúrák megnevezett halmaza 0.5 pont

interfész - osztály: egy osztály megvalósít (implementál) egy vagy több interfészt 0.5 pont

az ősztyály által megvalósított interfész a leszármazott osztály interfészének részhalmaza. 1 pont

UML jelölés: darabja 0.5 pont



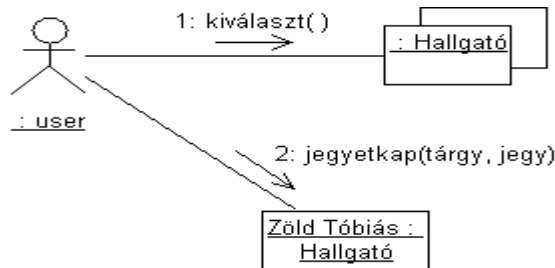
6. Rajzoljon kollaborációs diagramot, arra az esetre, amikor egy user a Hallgató osztályú objektumok egy kollekciónjából egy konkrét hallgatónak osztályzatot ad ! (5 pont)

multiobject 2 pont

példány 1 pont

user 1 pont

két üzenet 1 pont



7. Egy fájlrendszerben fastruktúrában könyvtárak és fájlok vannak. A könyvtárakban további könyvtárak és fájlok lehetnek. Egy könyvtárba fel lehet venni újabb könyvtárakat és fájlokat. A könyvtárakat ki lehet listázni (mind rekurzívan, mind csak egyszeres mélységben), a fájlaktól meg lehet kérdezni a típusukat. A könyvtárak meg tudják mondani, hogy összesen (minden alkönyvtárat beleértve) hány fájl található bennük.

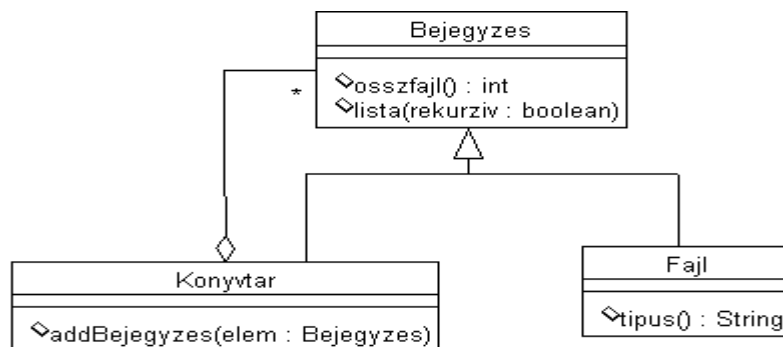
Készítsen osztálydiagramot és tüntesse fel a metódusok szignatúráit ! (7 pont)

osztályok 1 - 1pont

öröklés 1 pont

aggregáció 1 pont

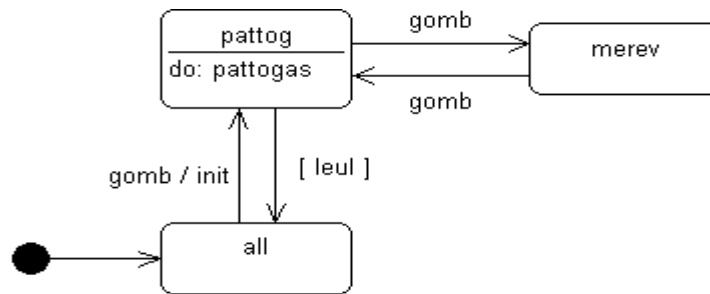
egy metódus 0.5 pont



8. A leejtett labda pattogását utánzó program vezérlésére egy gomb szolgál. A gombot megnyomva a labdát az induló helyzetből elengedjük, mire a labda leesik és energiát veszve a földről visszapattan. A pattogást addig folytatja, amíg az energiája el nem fogy (leül). Ha a labda leült, a gombot megnyomva a pattogás az induló helyzetből ismét elkezdhető. Ha a gombot pattogás közben nyomjuk meg, a labda mozgása átmenetileg megáll, majd a gomb ismételt megnyomásával a mozgás folytatható.

Rajzolja meg a labda viselkedését leíró UML state-chartot ! (8 pont)

minden állapot és él 1 pont
do 1 pont



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2001. január 23.

1. Algebrai axiómák segítségével specifikálja az egész számoknak az alábbi műveletekkel jellemzett kollekciónak ! (3 pont)

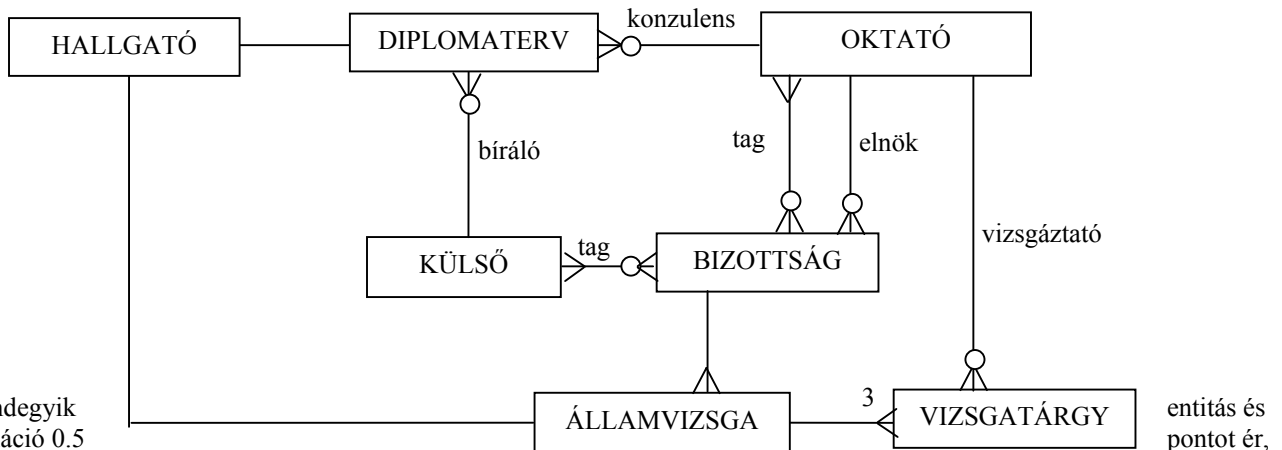
NEW()	új (üres) kollekción hoz létre.
ADD(b,i)	a b kollekciónhoz hozzáveszi az i egészet.
IN(b,i)	megadja, hogy a b kollekciónak hány darab i értékű eleme van.
DEL(b,i)	a b kollekciónból egyetlen i értékű elem törlésével kapott kollekción adja vissza.
ERA(b,i)	a b kollekciónból az összes i értékű elem törlésével kapott kollekción adja vissza.

IN(NEW(), x) = 0	- 0.5 pont, ha hiányzik
DEL(NEW(), x) = NEW()	- 0.5 pont, ha hiányzik
ERA(NEW(), x) = NEW()	- 0.5 pont, ha hiányzik
IN(ADD(b, x), y) = if (x==y) IN(b, y) + 1 else IN(b, y)	1 pont
DEL(ADD(b, x), y) = if (x==y) b else ADD(DEL(b, y),x)	1 pont
ERA(ADD(b, x), y) = if (x==y) ERA(b, y) else ADD(ERA(b, y), x)	1 pont

2. Készítsen entitás-relációs diagramot az alábbi problémára !(8 pont)

Az egyetem végzős hallgatói (név, neptun-kód, cím, szigorlatátlag) diplomaterveznek, majd államvizsgát tesznek. Egy hallgató egyetlen diplomatervet készít és feltételezhetjük, hogy csak egyetlen alkalommal államvizsgázik. A diplomatervnek mindig van egy oktató konzulense. Az elkészült diplomatervet egyetlen külsős szakember bírálja el. Az államvizsgára bizottságot állítanak össze. Egy bizottságnak mindig van egy docens vagy tanár elnöke, legalább egy oktató és egy külsős tagja, és egy adminisztrátor titkára. A bizottságot egy alkalomra (megadott időre és helyre) hozzák létre és a bizottság legkevesebb egy, de inkább 5-6 hallgatót államvizsgáztat egymás után. Az államvizsgán a diploma megvédésén kívül három tárgyból kell vizsgát tenni. Egy vizsgán egy hallgatót egy tárgyból egy oktató (általában a tárgy felelőse) vizsgáztat, de ugyanazon oktató több tárgyból is vizsgáztathat. Az államvizsga eredménye a szigorlati átlagból, a vizsgatárgyakból és a diplomatervre kapott jegyből áll elő.

HALLGATÓ (név, neptun-kód, cím, szigorlatátlag)
 DIPLOMATERV (cím, hallgató, oktató(konzulens), külsős(bíráló), eredmény)
 OKTATÓ (név, fokozat)
 KÜLSŐS (név, cím)
 BIZOTTSÁG (idő, hely, oktató(elnök), titkár, oktató(tag), külsős(tag))
 ÁLLAMVIZSGA (bizottság, hallgató, eredmény)
 VIZSGATÁRGY (államvizsga, tárgynév, oktató(vizsgáztató), eredmény)



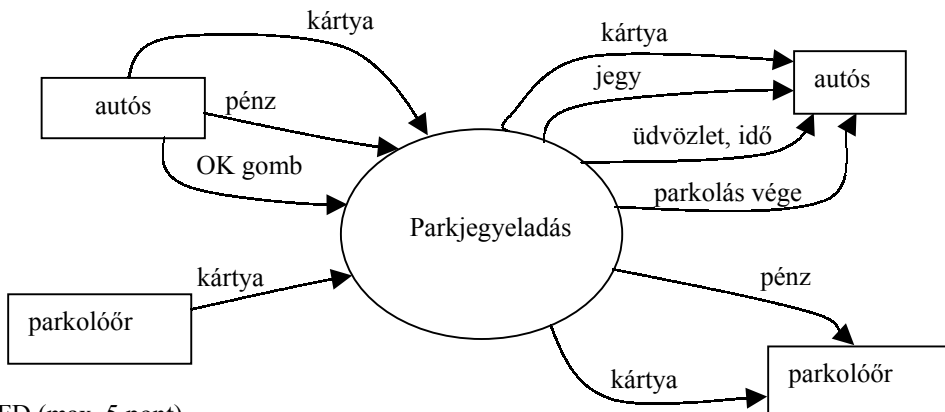
Pontozás: mindegyik mindegyik reláció 0.5 max. 8 pont (8.5 pont van benne !!!)

entitás és pontot ér,

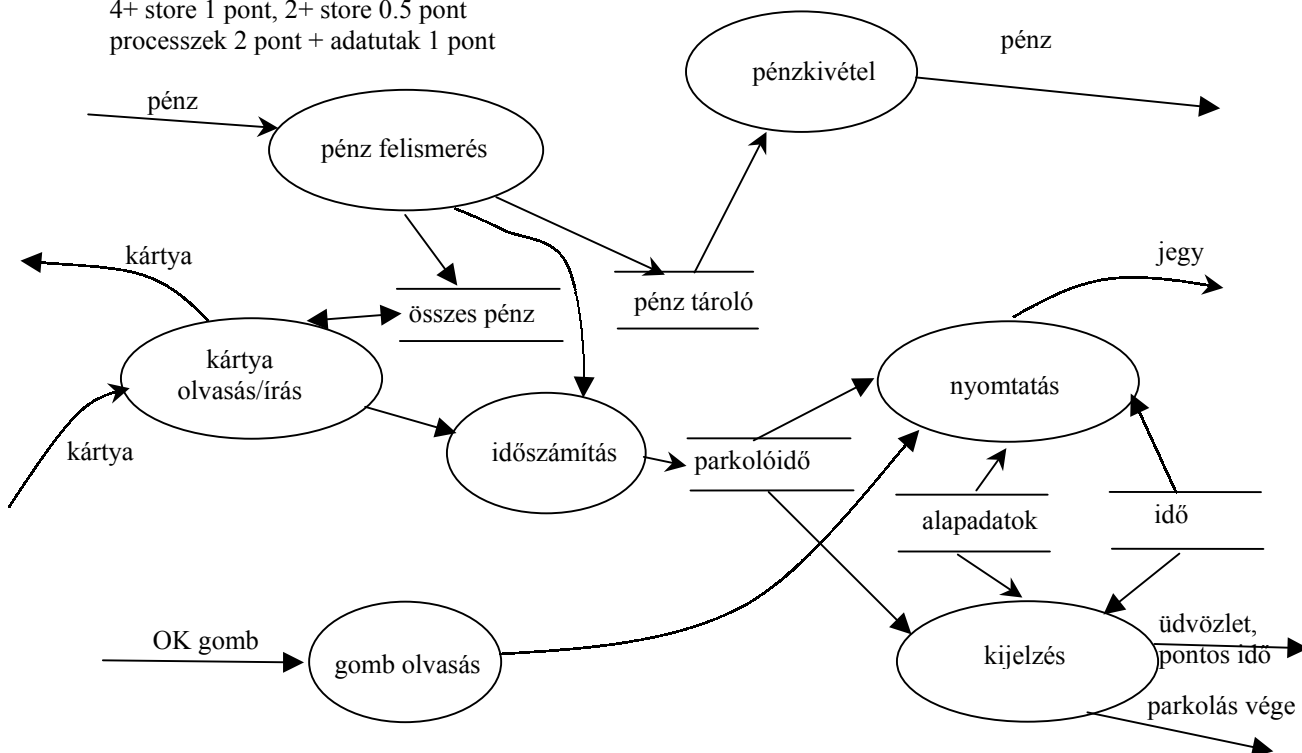
3. Készítse el egy parkolójegyet kiadó automata adatfolyamábráját ! Adja meg a context diagramot és annak 0 szintű felbontását ! (7 pont)

Az automatától pénzért parkolási idő vásárolható. Alaphelyzetben az automata kijelzőjén a pontos idő és üdvözlőszöveg látható. Pénz bedobásakor a kijelzőről leolvashatjuk, hogy a bedobott pénz ellenében meddig parkolhatunk. Az OK gombot megnyomva az automata parkolójegyet nyomtat. Ha a pénz bedobása előtt a parkolókárttyánkat az olvasóba helyezzük, akkor kedvezményes tarifáért vásárolhatunk parkoloidót, amíg a kártyán levő kedvezmény el nem fogy. A parkolóőr speciális kártyájára felíródik a legutóbbi ürítés óta beszedett pénz mennyisége és az automatában összegyűlt pénz kivehető.

Context diagram (2 pont) (2 terminátor 1 pont, összes adat 1 pont, 2 hiba 0.5 pont, 4 hiba 0 pont)



0 szintű DFD (max. 5 pont)
level balancing 1 pont (2 hiba 0.5 pont, 4 hiba 0 pont)
4+ store 1 pont, 2+ store 0.5 pont
processzek 2 pont + adatutak 1 pont



4. Adjon JSD alapú megoldást az alábbi problémára ! (8 pont)

A Bergengóciai Tengeri-herkentyű Művek a raktáron lévő tengeri-herkentyűitől (TeHe) szeretne megszabadulni, ezért bárki, aki elfogadja a tengeri-herkentyű birtoklás feltételeit, egy díszes birtoklevél kíséretében (melyen a sorozatszám mellett a herkentyű beceneve is szerepel) ingyen és bérmentve megkapja első TeHe-nét. Ahhoz, hogy a TeHe-n billentyűzni lehessen, először hozzáférhetővé kell tenni a billentyűket. Ezt úgy lehet megtenni, hogy a TeHe kurbliját egy adott ütemben (az ütem példányonként változó, és ezt az ütemet kell a tulajdonosnak eltalálni) meg kell tekerni. Tetszőleges sokszor lehet próbálkozni. A sikeres kurblijást a TeHe harangszóval jelzi és hozzáférhetővé teszi a billentyűzetét. Ettől kezdve a kurblijnak már nincs szerepe. A birtoklás feltételei között szerepel, hogy a TeHe-t időnként billentyűzet ellenőrzésre a szervizbe kell vinni. Az első szerviz legalább 15 billentyűzés után kérhető, de ez után két szerviz között már csak az az előírás, hogy legalább egy

billentyűzésre sor kerüljön. (Csalni nem lehet, mert a herkentyű logolja a történéseket) A herkentyű átvételét követően bármikor kérhető általános átvizsgálás, amely a herkentyű minden egységére kiterjed. Az átvizsgálások során észlelt hibákat a szerviz javítja, de azt a herkentyűt, amelyeket a szerviz a billentyűzet ellenőrzése során hibásnak talál, és már egy általános átvizsgálás alkalmával javításon esett át, leselejtezik.

Definiálja az alábbi funkciókat ellátó processzeket:

Kérésre adja meg azokat a herkentyűket, amelyek már 15-ször kurbliztak, de még nem használható a billentyűzetük.

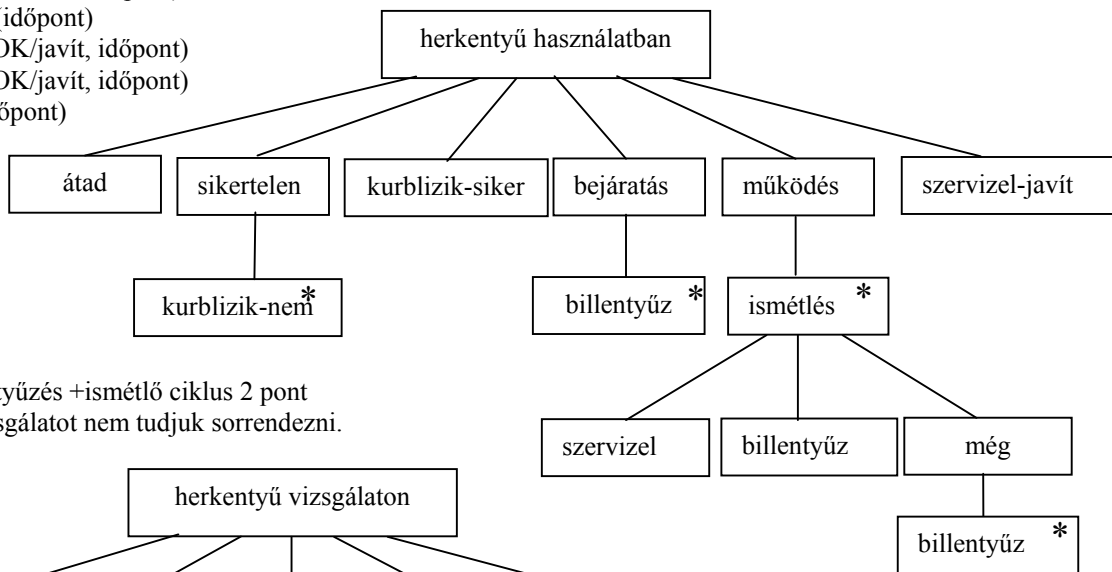
Kérésre adja meg azokat a herkentyűket, amelyek már 5 általános átvizsgáláson átesetek át, és a legutóbbi billentyűellenőrzés óta már háromszor billentyűztek rajtuk.

Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modellt,**

kapcsolja a modellhez a **funkció processzeket.**

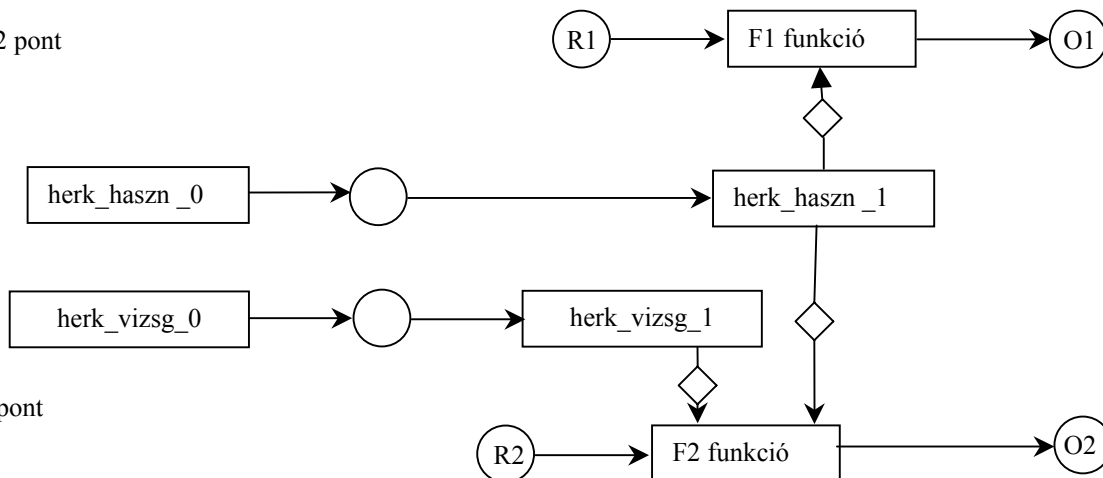
- átvész (herkentyű) (sorozatszám, becenév, tulaj, időpont)
- kurbliz (herkentyű) (siker/nem, időpont)
- billentyűz (herkentyű) (időpont)
- szervizel (herkentyű) (OK/javít, időpont)
- átvizsgál (herkentyű) (OK/javít, időpont)
- selejtez (herkentyű) (időpont)

entitás/akció leírás 2 pont



sikeres kurbli 1 pont, billentyűzés +ismétlő ciklus 2 pont
gond: a használatot és a vizsgálatot nem tudjuk sorrendezni.

ez az entitás 2 pont



teljes SSD 1 pont

5. Mit jelent az objektum változókon bevezetett **tipizálás** és **kötés** fogalma ? Milyen összefüggés van közöttük ? (3 pont)

tipizálás: milyen objektumok tehetők a változóba.

(statikus – fordításkor, dinamikus – futáskor defínit) (1 pont)

kötés: kihez kapcsolódnak a műveletek (statikus – a változóhoz, dinamikus – az értékhez) (1 pont)

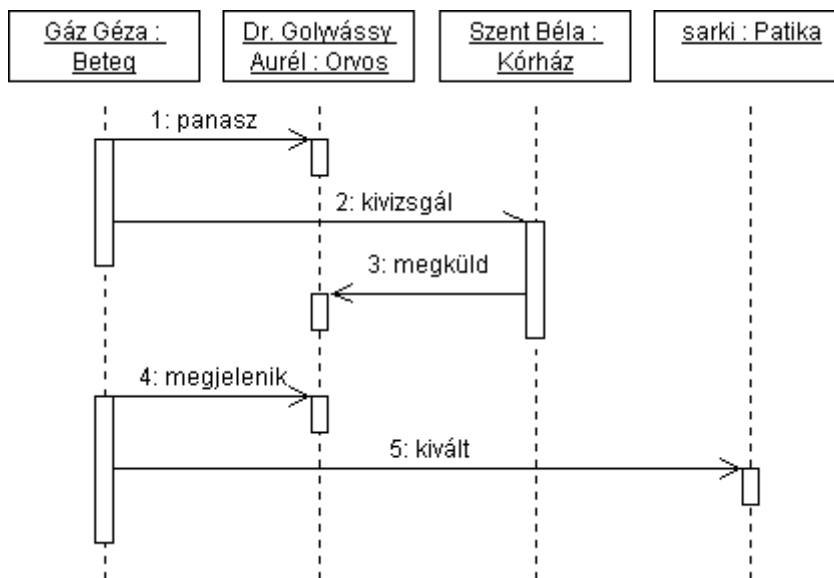
kapcsolat (1 pont):

	<i>stat. tipizálás</i>	<i>din. tipizálás</i>
<i>stat.kötés</i>	nem OO	értelmetlen
<i>din. kötés</i>	normál OO (Java, C++)	Smalltalk, Excel cella

6. Rajzoljon az eseményekről UML szekvencia-diagramot ! (5 pont)

Gáz Géza rosszul érzi magát, és elmegy a házi orvosához. Elpanaszolja neki, hogy már két hete lázas, és a jobb lábfeje a kétszeresére dagadt. Dr. Golyvássy Aurél, az orvos elküldi beutalóval Gáz urat kivizsgálásra a Szent Béla kórházba, ahol különböző vizsgálatokat ejtenek meg rajta. Az adatokat pár nap alatt kiértékelik és postán megküldik Dr. Golyvássynak. Amikor Gáz úr ismét megjelenik a rendelésen, a doktor receptre felír Gáz úrnak köptetőt, amelyet ő kivált a sarki gyógyszertárban.

objektumok 1 pont
 üzenet 0.5 pont
 fókusz 1 pont
 új szál 0.5 pont

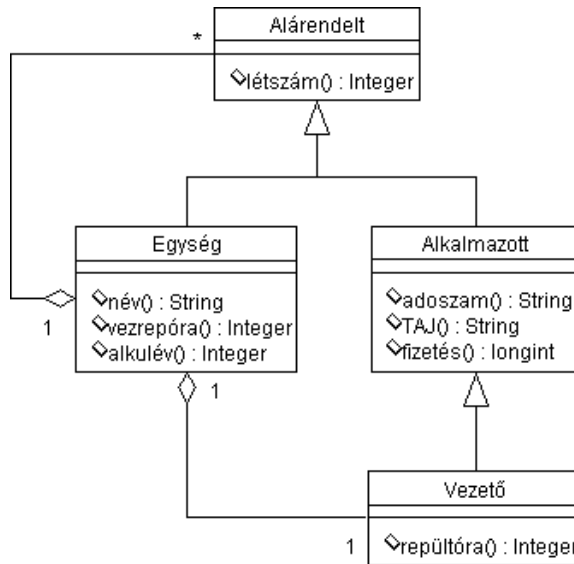


7. Készítsen UML osztálydiagramot és azon adja meg az osztályok metódusainak szignatúráit is ! (9 pont)

A Vidékelőremozdító Minisztérium felépítése sok rétegű hierarchiát alkot. A minisztérium államtitkárságokból, az államtitkárság divíziókból, a divíziók főosztályokból, a főosztályok ... stb. állnak. Minden egységnek van egy vezetője, és számos tisztviselője.

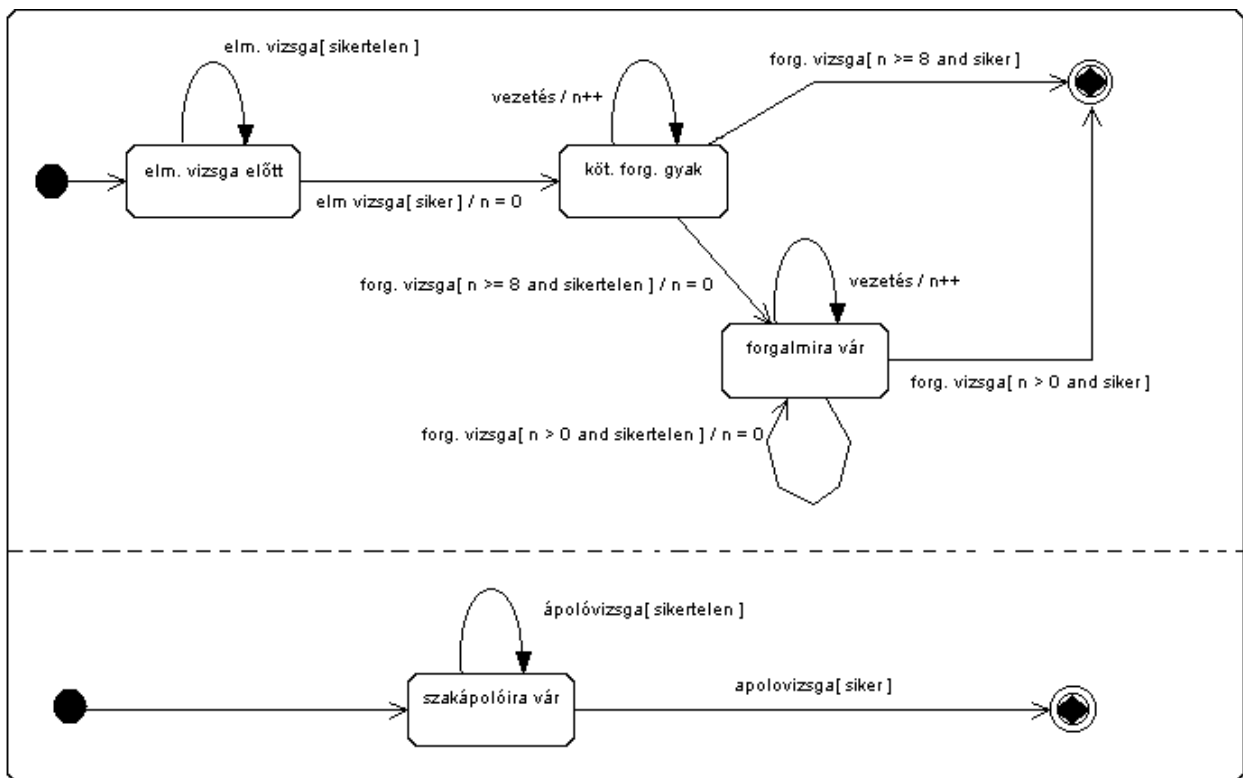
Az egységtől lekérdezhető az elnevezése, alakulásának éve, a vezető repült óráinak száma valamint az egységhez tartozó összes alkalmazotti (vezető + tisztviselői) létszám. Az alkalmazottaktól lekérdezhető az adószámuk, a TAJ-számuk és a fizetésük.

osztályok 0.5 pont
 öröklés 1 pont
 aggregáció 1 pont
 létszám a helyén 1 pont
 repült óra 1 pont
 többi metódus 1 pont



8. Rajzolja meg a hallgató viselkedését leíró UML state-chartot ! (7 pont) Az állapot-átmeneteket pontosan definiálja.

A zöldlámpás autók vezetésére jogosító képzésben résztvevő hallgatónak a zöldlámpa használatára vonatkozó elméleti vizsga letétele után legalább 8 alkalommal kell a forgalomban felügyelet mellett vezetési gyakorlatot végezni, majd ezt követi a forgalmi vizsga. A jogosítvány megszerzéséhez ugyancsak szükséges a szakápolói vizsga, amelyet a hallgató a felvételt követően bármikor letethet. Zöldlámpás jogosítványt akkor kap, ha a forgalmi és ápolói vizsgát egyaránt letette. Valamennyi sikertelen vizsga korlátlan számban ismételhető. Az ismételt forgalmi vizsgának előfeltétele, hogy legalább 1 alkalommal vezetési gyakorlatot kell végezni.



ápolói szál 2 pont
 minden állapot és minden él 0.5 pont (max 5 pont) 5.5. van benne.

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK

PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2001. május 24.

1. Készítsen entitás-relációs diagramot az alábbi problémára ! (8 pont)

Egy konferenciára (neve, helye, ideje) több cikket (címe és hossza) nyújtanak be. A cikkeket személyek (név, e-mail cím) írják. Egy cikknek egy vagy több szerzője lehet, az egyes szerzőket szerzői hányad illeti meg. Ugyanaz a személy több cikket is benyújthat. Egy cikket csak egy konferenciára adnak be. A programbizottság a beérkezett cikkeket bírálja. A bizottság tagjai személyek, akik a bizottsági munkáért díjazást kapnak. Minden cikket legalább hárman bírálják. A bírálat eredménye lehet: elfogadás, feltételes elfogadás, feltételes elvetés vagy elvetés. Bíráló lehet a bizottság által felkért személy, aki a bírálat elkészítéséért díjazást kap. A programbizottság a cikkeket elfogadásáról szavazással dönt. Szavazategyenlőség esetén a bizottság elnökének szavazata dönt. A cikkeket a bizottság szekcióba sorolja. Egy konferencián több szekció van, minden szekciónak van titkára. Egy szekcióba több cikket sorolnak, az elfogadott cikk csak egy szekcióba kerülhet. Egy személy több bizottságnak is lehet tagja, illetve elnöke.

CIKK (cím, konferencia, szekció, hossz)

SZERZŐ (személy, cikk, hányad)

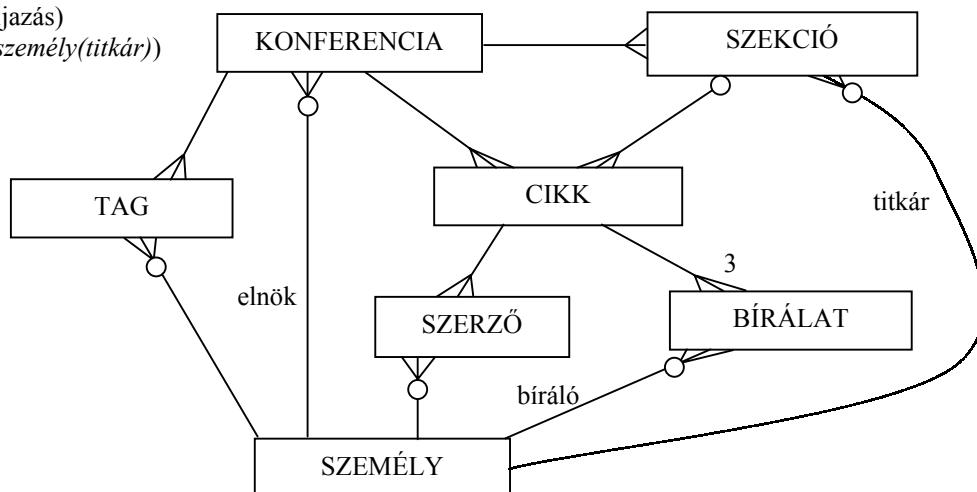
SZEMÉLY (név, e-mail cím)

BÍRÁLAT (cikk, személy(bíráló), eredmény, díjazás)

KONFERENCIA (név, idő, hely, személy(elnök))

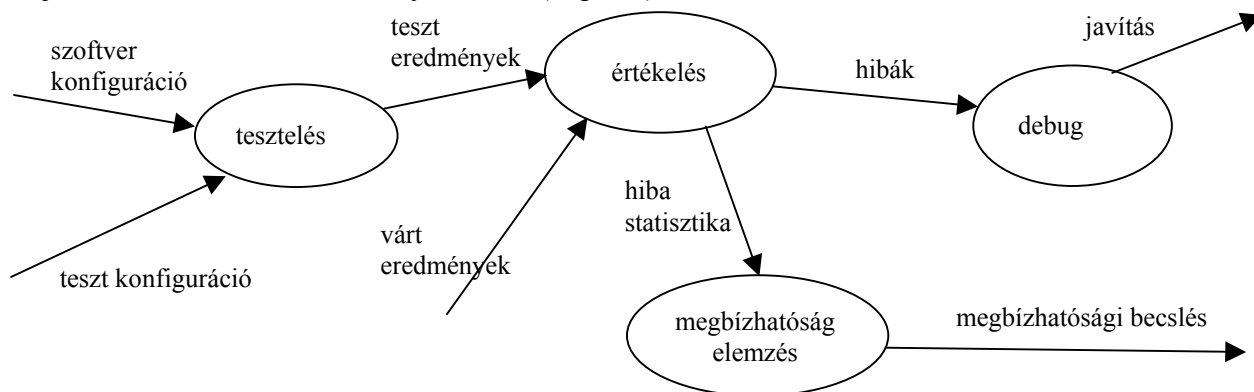
TAG (konferencia, személy, díjazás)

SZEKCIÓ (név, konferencia, személy(titkár))



Pontozás: mindegyik entitás és mindegyik reláció 0.5 pontot ér, max. 8 pont (9 pont van benne !!!)

2. Definiálja a tesztelés információs folyamatát ! (7 pont)



Pontozás: mindegyik processz 1 pontot, mindegyik adatút 0.5 pontot ér, max. 7 pont (8 pont van benne !!!)

3. Adjon JSD alapú megoldást az alábbi problémára ! (8 pont)

Egy hetilap főszerkesztőjének feladata a rendelkezésre álló cikkekből a szám összeállítása. Egy cikk elkészítésére a főszerkesztő ad megbízást a szerkesztőnek a határidő és a felhasználható költségek megadásával. A szerkesztő (szerző) anyagot gyűjt, majd írja a cikket. Ha elkészült, akkor az anyagot átadja egy lektornak, aki a megjegyzéseivel ellátott anyagot visszaadja a szerkesztőnek. Ha javít a szerkesztő a cikken, akkor azt újra lektoráltatni kell. A főszerkesztő visszaadhatja a szerzőnek az anyagot átdolgozásra, aki az átdolgozás után ismét lektoráltat a fentiek szerint. A főszerkesztő teszőleges sokszor adhatja vissza a cikket átdolgozásra. Az átvett cikket a főszerkesztő vagy beszerkeszti valamelyik számba, vagy kidobja. Egy szerkesztő egyszerre több cikken is dolgozhat, egy cikknek egy, kivételes esetben két szerzője lehet. A szerkesztők nagyon szeretnek dolgozni, de időnként szabadságra mennek.

Definiálja az alábbi funkciókat ellátó processzeket:

Kérésre adja meg azon cikkek szerzőit, amely cikkeket a főszerkesztő egyszer már átdolgozásra visszaadott és most éppen lektorálás alatt vannak.

Kérésre adja meg azon szerkesztőket, akik most szabadságon vannak és nincs leadandó cikkük.

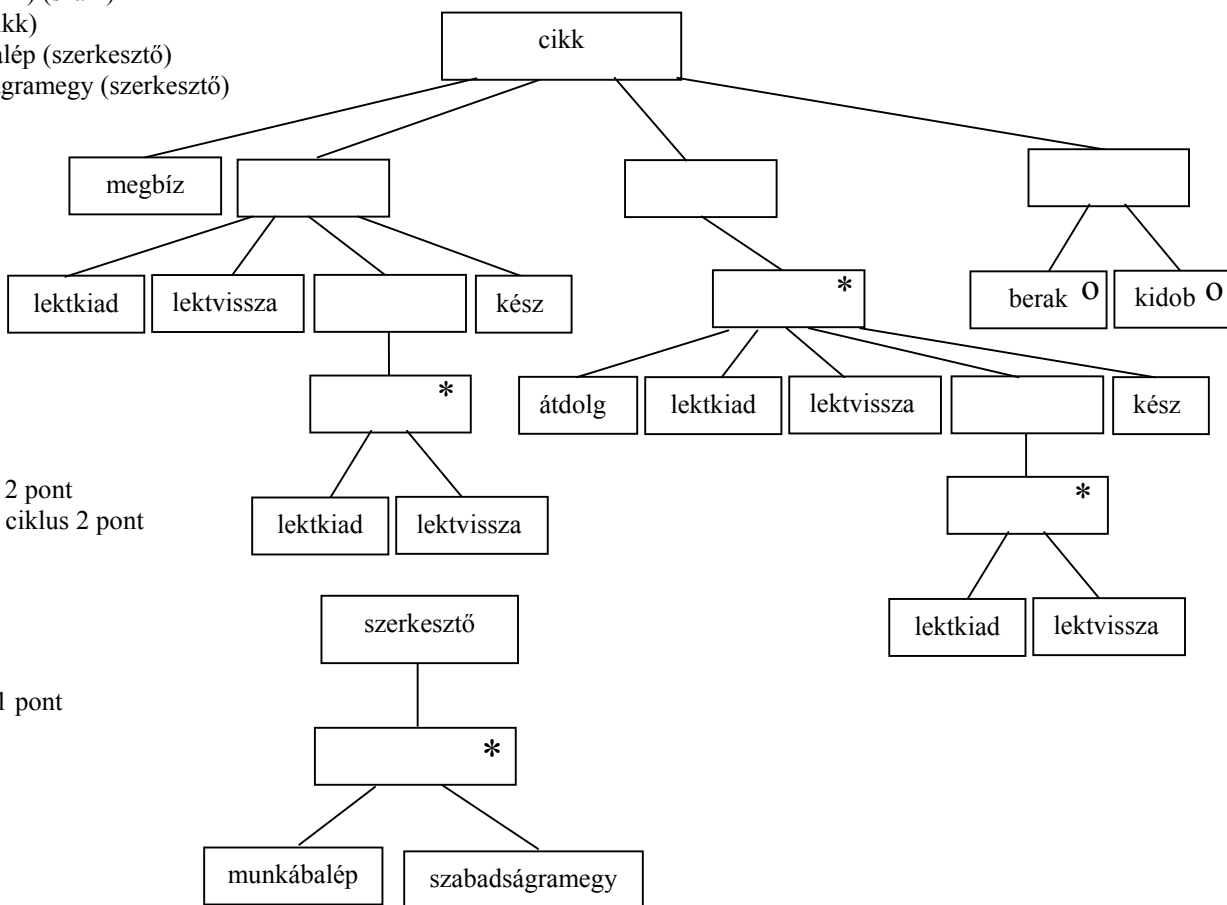
Készítse el az

**entitás/akció leírást, az entitások attribútumaival,
az entitás struktúra diagramot,
az initial modelt,**

kapcsolja a modellhez a **funkció processzeket.**

- megbíz (cikk) (téma, határidő, költség, szerkesztő1, szerkesztő2)
- letkiad (cikk) (lektor)
- lektvissza (cikk) (vélemény)
- kész (cikk) ()
- átdolg (cikk) ()
- berak (cikk) (szám)
- kidob (cikk)
- munkábalép (szerkesztő)
- szabadságramegy (szerkesztő)

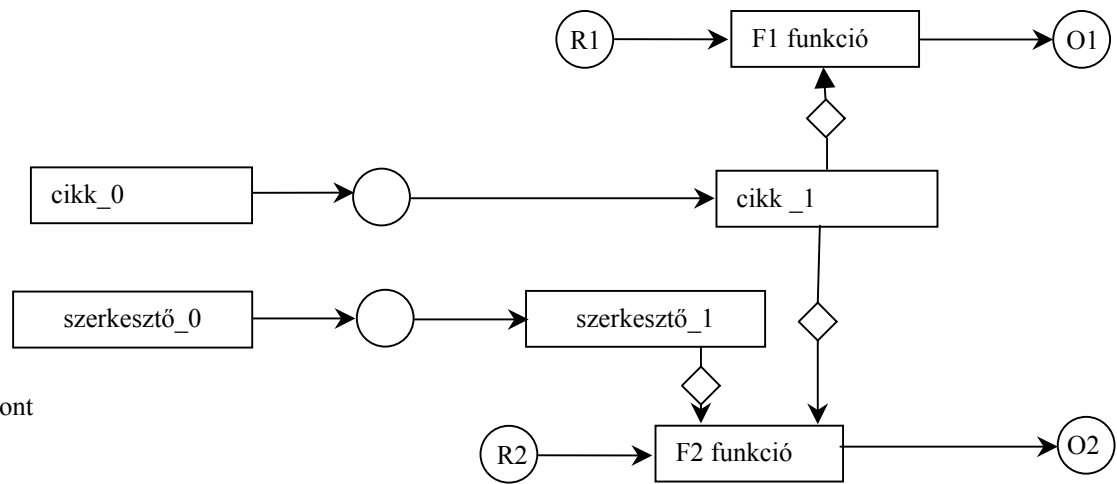
entitás/akció leírás 2 pont



lektori ciklus 2 pont

főszerkesztői ciklus 2 pont

ez az entitás 1 pont



teljes SSD 1 pont

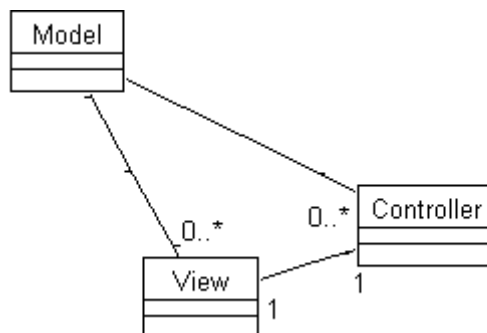
4. Adja meg a program hosszára vonatkozó Halstead-féle mérőszámot és a programtérfogat definícióját ! (3 pont)

ζ_1 = különböző operátorok száma ζ_2 = különböző operandusok száma $\zeta = \zeta_1 + \zeta_2$ szótár
 N_1 = összes operátorok száma N_2 = összes operandusok száma $N = N_1 + N_2$ program hossza
 $N = \zeta_1 * \lambda \log_2 \zeta_1 + \zeta_2 * \lambda \log_2 \zeta_2$
 $\zeta = N * \lambda \log_2 \zeta$

5. Ismertesse az MVC modellt ! Rajzolja le a classdiagramot és definiálja az osztályok felelősségeit. (4 pont)

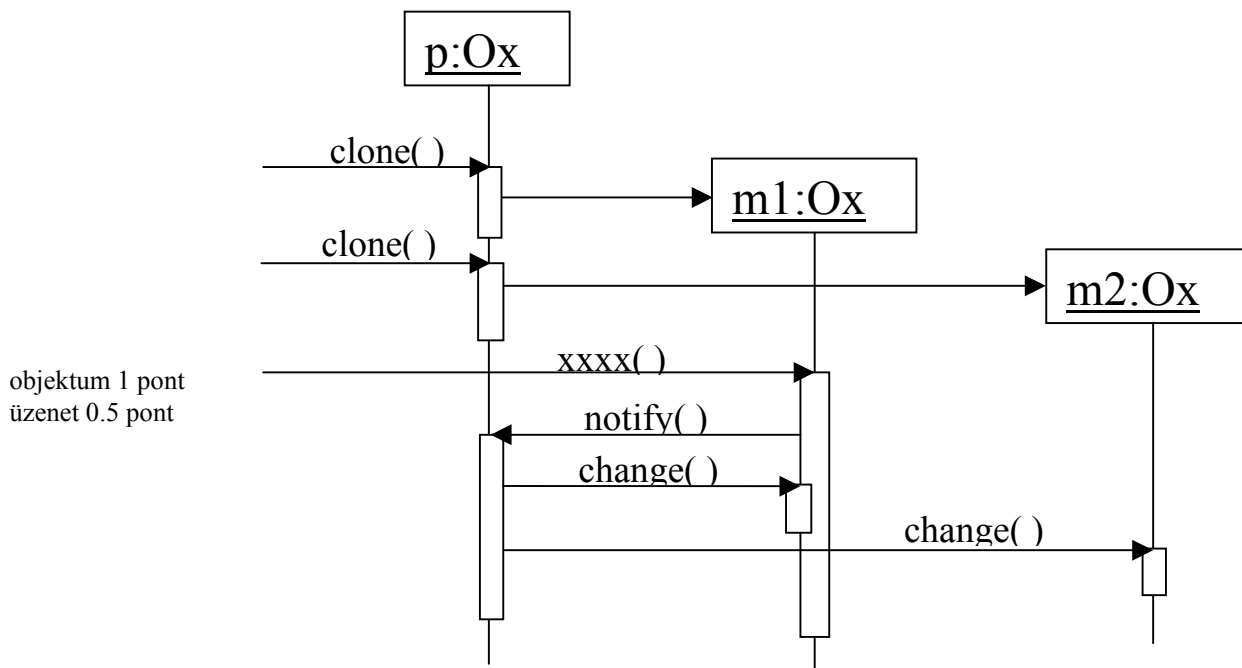
M – model – a business logic
 V – megjelenés
 C – controller – a V és M vezérlése

minden class 1 pont
 a 3 asszociáció 1 pont

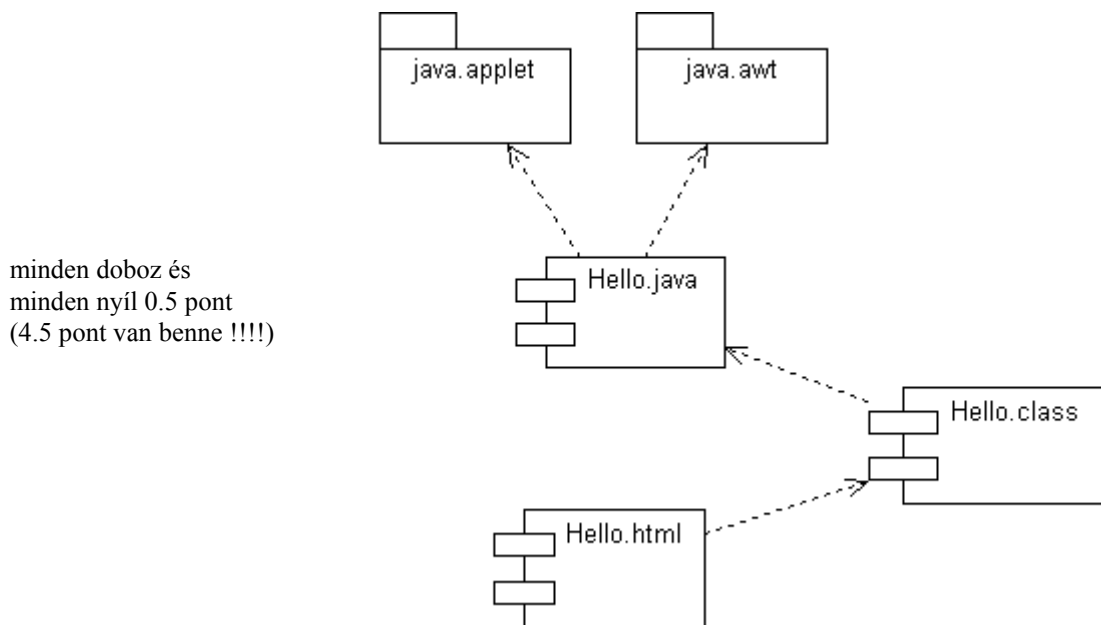


6. Legyen egy tetszőleges O osztály. Ennek egy olyan Ox változatát kell elkészíteni, amelyik rendelkezik O valamennyi tulajdonságával + egy p példánya a clone() metódusának meghívásakor készít magáról egy m másolatot, amely folyamatosan követi p állapotának minden változását. p-ről több másolat is készíthető, de m már nem másolható tovább. Egy m másolaton egy kliens részéről kezdeményezett olyan műveletet, amely m állapotának megváltozását okozná, m továbbítja p-nek, aki a műveletet saját magán hajtja végre, hogy valamennyi másolata kövesse a változtatást.

Rajzoljon UML szekvencia diagramot arra az esetre, amikor p-ről egy kliens készít két másolatot (m1, m2), majd az m1-nek meghívja egy olyan metódusát (xxxx()), amely m1-nek módosulását okozná. (7 pont)



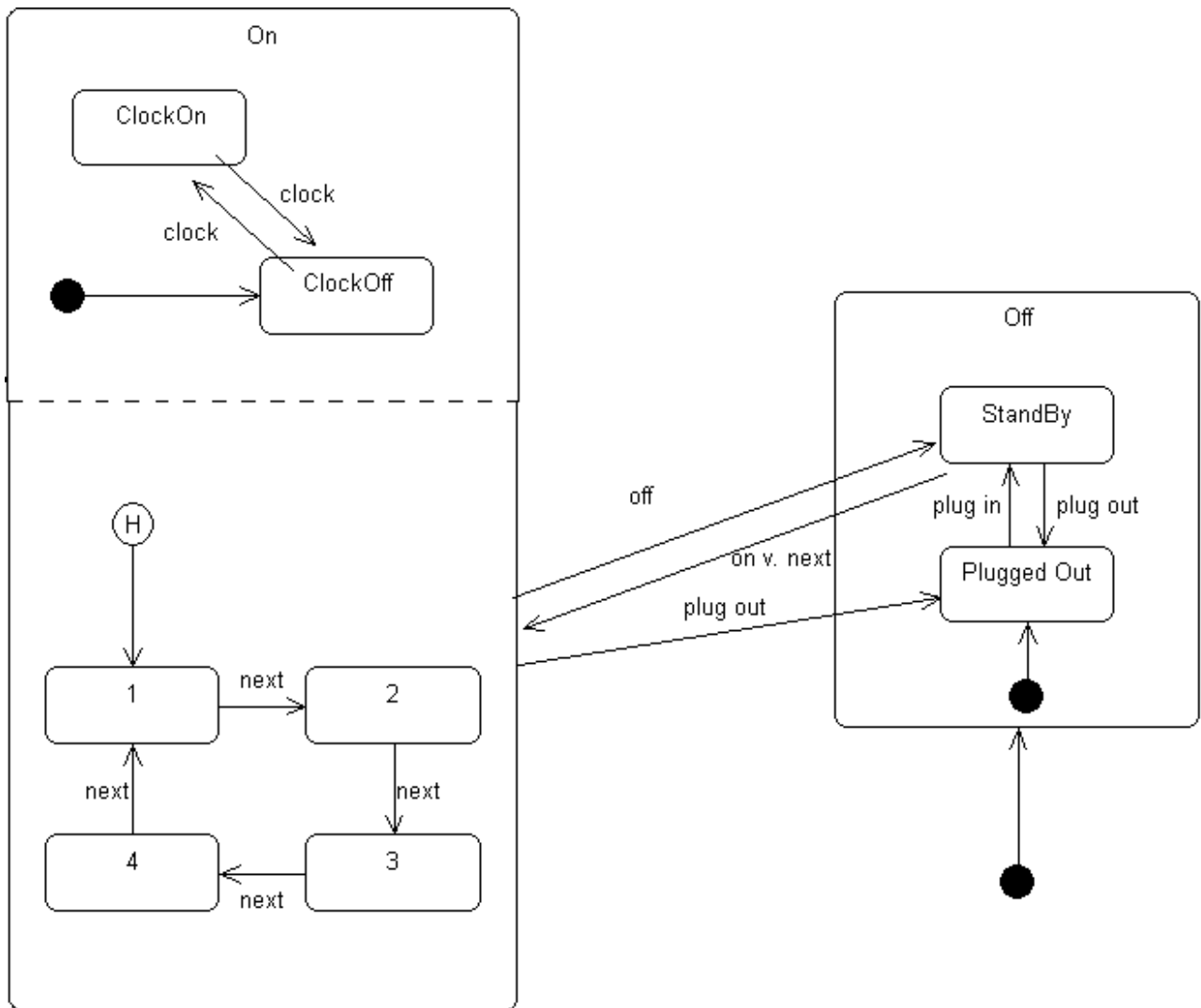
7. Rajzolja fel az UML komponens diagramját egy html oldalból és egy java appletből álló programnak, amelyben az applet kiírja a "Hello World !" feliratot (4 pont)



8. Rajzolja meg a HályTek® televíziókészülék viselkedését leíró UML state-chartot ! (9 pont)

A kicsomagolt televíziókészülék csatlakozóját a dugaszoló aljzatba kell helyezni, aminek hatására a készülék feszültség alá kerül, és készenléti állapotban vár. A készüléket a távirányítón található ON vagy NEXT gombokkal lehet bekapcsolni. A kikapcsoláshoz a távirányító OFF gombját kell megnyomni, ennek hatására a készülék ismét készenléti állapotba kerül. Ismételt bekapcsoláskor a legutoljára beállított csatorna lesz látható. A csatornák között a NEXT gombbal tudunk váltani, a gomb megnyomásakor mindig a soron következő csatorna lesz látható. A készüléknek 4 különböző csatornája van, a negyedik után a NEXT gomb az első csatornára kapcsol. A készülék tápkábelét bármikor ki lehet húzni, de ekkor a készülék nem fog működni ! A feszültség alá helyezés utáni első bekapcsoláskor az első csatorna lesz kiválasztva.

A készülék képes az aktuális idő megjelenítésére. Az óra azonban bekapcsoláskor nem látszik. Ha a képernyőn látni kívánjuk az órát, akkor a CLOCK gombot kell megnyomnunk. Az órát a CLOCK gomb ismételt megnyomásával tudjuk eltüntetni.



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2001. június 5.

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A LetepAir légitársaság repülőjáratokat üzemeltet. A járat jellemzője a járatszám, az induló és célállomás. Egy adott napon a konkrét járatra vonatkozóan nyilvántartjuk az utasok számát és a késést. Egy napon nem indul többször ugyanazon számú járat. A járatához a repülőt az aktuális utaslétszámot figyelembe véve rendelik hozzá. A repülőket leírja a lajstromjel (LA-DSE), a gép repült óráinak száma és a gép típusa. A gép típusát definiálja a típusazonosító (B767), a hossz és a maximális utaslétszám. A társaságnak különböző típusú gépei vannak, de a típus jellemző adatait az általa nem üzemeltetett gépekre vonatkozóan is nyilvántartja. Minden repülőgépet két pilóta (első és másod) vezet. A pilótákat jellemzi a nevük és az összes repült órák száma. A pilóták csak azokat a típusokat vezethetik, amelyekre típusengedélyt szereztek. A típusengedély egyik jellemző adata a megszerzés dátuma.

JÁRAT (szám, indulás, cél)

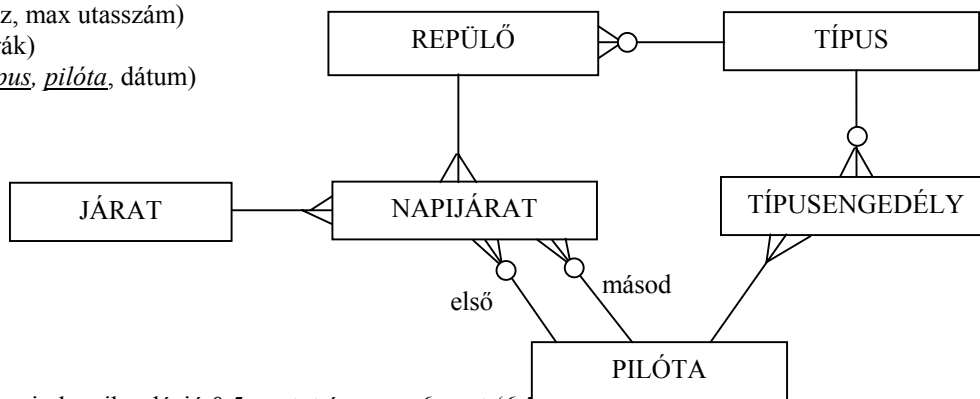
NAPIJÁRAT (járat, dátum, utasok száma, késés, *repülő*, *pilóta(első)*, *pilóta(másod)*)

REPÜLŐ (lajstromjel, repült óra, *típus*)

TÍPUS (azonosító, hossz, max utasszám)

PILÓTA (név, repült órák)

TÍPUSENGEDÉLY (*típus*, *pilóta*, dátum)



mindegyik entitás és mindegyik reláció 0.5 pontot ér, max. 6 pont (6.5 pont van benne !!!)

2. Nevezze meg sorrendben a **CMM szintjeit** ! (4 pont)

minden találat 0.5, minden hiba -0.5 pont, jó sorrend 1.5 pont

kezdetleges, ismétlődő, definiált, irányított, optimalizált

3. Mi a különbség a **walkthrough** és az **inspection** között ? (2 pont)

walkthrough
mindig alkalm.
beszámoló

inspection
definit szintaxis
előzetes anyag

4. Nevezze meg a **CASE rendszerek** fontosabb komponenseit ! (4 pont)

minden találat 0.5, minden hiba -0.5 pont, max. 4 (4.5 pont van benne !!!)

Repository
Project management support
Structured diagramming tools
Document generation facilities
Skeleton code generator
Query language facilities
Form creation tools
Design analysis and checking tools
Export/import facilities

5 Adjon **JSD** alapú megoldást az alábbi problémára ! (9 pont)

A Super Business School-ban nemzetközi könyvelői tanfolyamokat indítanak, ahová a tetemes tandíjat megfizető jelentkezőket veszik fel. A szorgalmi időszakban hetente két alkalommal tartanak előadást. Amikor az előadások befejeződnek vizsgát kell tenni. A sikertelen vizsga tetszőleges sokszor ismételhető, de minden ismételő vizsga előtt kötelező részt venni egy fizetős konzultáción. Az elméleti tanfolyam mellett a hallgatók szaknyelvi képzést is kapnak nyelvi laborban. Aki felkészültnek érzi magát, az bármikor megkísérelheti a nyelvvizsga letételét. A nyelvvizsga is sokszor ismételhető. Az ismétléshez csak a pótvizsga díját kell befizetni. A labort a hallgató mindaddig igénybe veheti, amíg a nyelvvizsgát le nem tette. A tanfolyam elvégzéséről csak az kap bizonyítványt, aki mind az elméletből, mind a nyelvből sikerrel vizsgázott.

Definiálja az alábbi funkciókat ellátó processzeket:

Automatikusan állítsa ki a bizonyítványt azoknak, akik megfeleltek a feltételeknek.

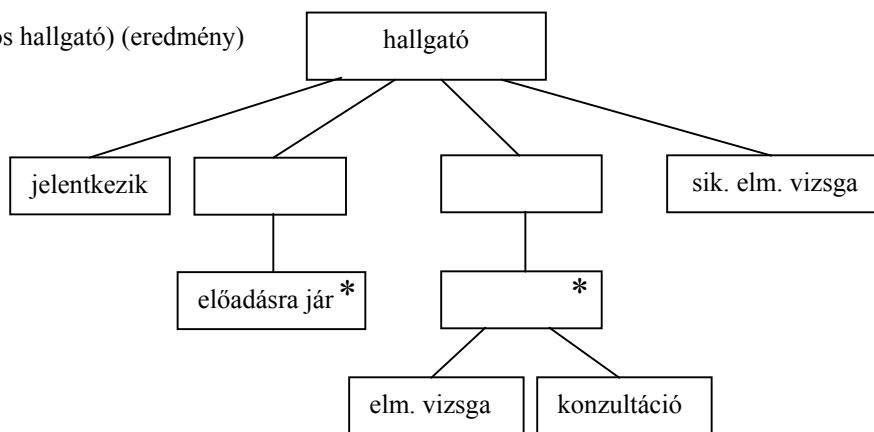
Kérésre adja meg azokat, akik többször jártak nyelvi laborra, mint előadásra.

Kérésre adja meg azokat, akiknek sem az elméleti sem a nyelvi vizsga nem sikerült elsőre.

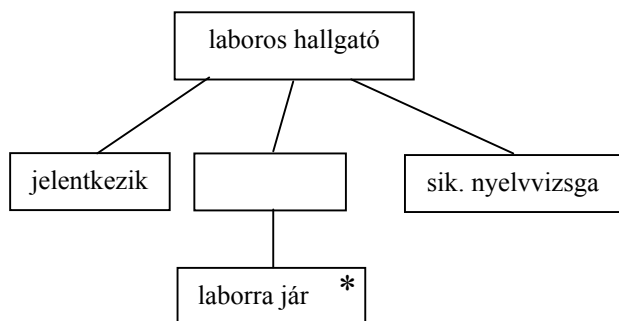
Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modellt, a funkció processzeket.** kapcsolja a modellhez a

- jelentkezik (hallgató, laboros hallgató, nyelvvizsgáló) (alapadatok)
- előadásra jár (hallgató) (idő)
- elméleti vizsga (hallgató) (eredmény)
- konzultáció (hallgató) (idő)
- laborra jár (laboros hallgató) (idő)
- nyelvvizsga (nyelvvizsgáló, laboros hallgató) (eredmény)
- vizsgadíjat fizet (nyelvvizsgáló)

entitás/akció leírás 2 pont

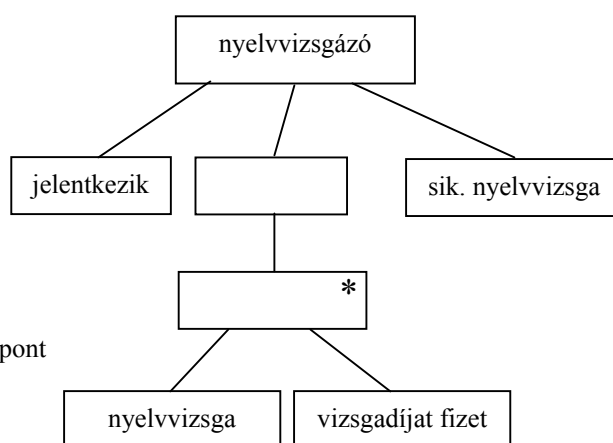


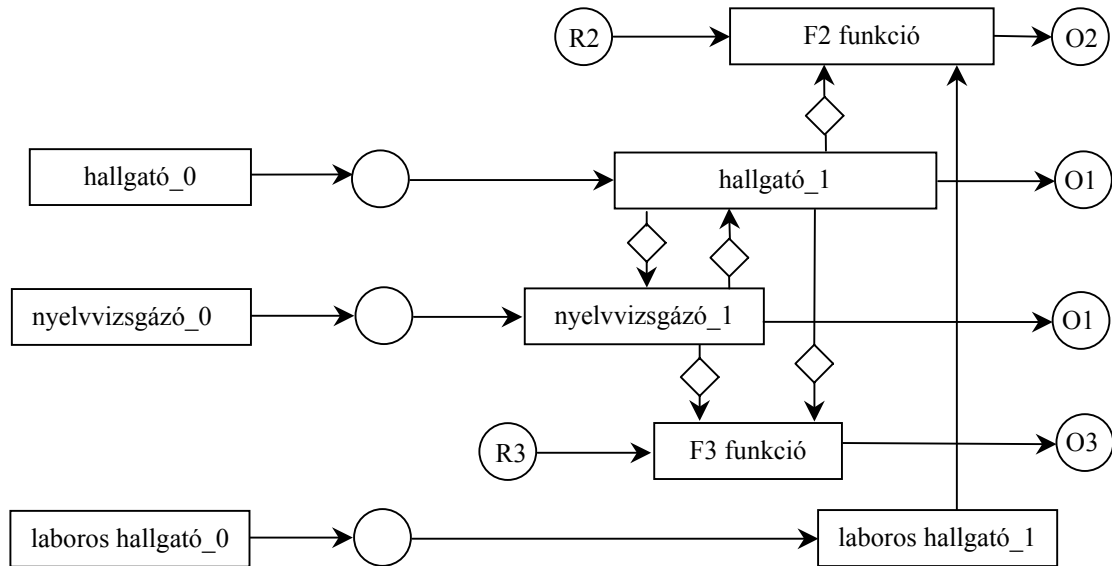
hallgató ELH 2 pont



laboros hallgató ELH 1 pont

nyelvvizsgáló ELH 2 pont

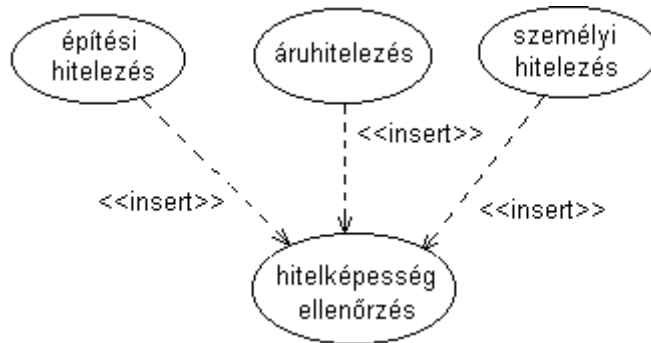




teljes SSD 2 pont, funkciók nélkül csak 0.5 pont

6. Egy bank áruvásárlási, építési és személyi hitelt ad az ügyfeleknek. A hitelezések lebonyolítása különbözik, de mindegyik tevékenységben közös és fontos a kölcsönt igénylő hitelképességének ellenőrzése. Rajzolja meg az **UML használati eseteket** ! (2 pont)

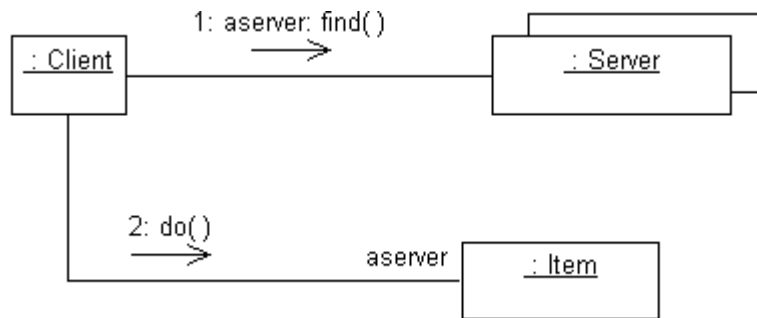
4 use case = 1 pont
insert = 1 pont



Elírás a megoldásban:
nem <<insert>> hanem <<include>> !!!!

7. Mi a **multioject** ? Hogyan jelöljük és milyen diagramon fordul elő ? (2 pont)

jele 1 pont
kollaboráció 1 pont



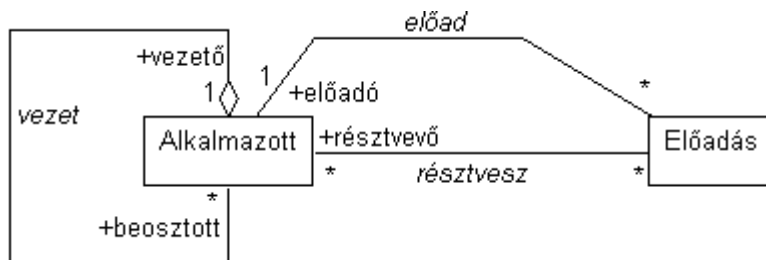
8. Egy cég az alkalmazottai részére továbbképzés céljából szakmai előadásokat szervez, amelyeket a cég egy-egy hozzáértő munkatársa tart. Az egyik előadás előadója lehet a másíknak hallgatója. Egy előadáson csak azok vesznek részt, akiket erre a vezetőjük kijelölt. Tanfolyamra küldhetik a vezetőket is.

Rajzoljon **osztálydiagramot** a szerepek és a multiplicitások feltüntetésével. (5 pont)

Rajzoljon **kollaborációs diagramot** arra az esetre, ha egy vezetőt megkérdeznék arról, hogy egy adott beosztottját kik tanították az előadásokon. (6 pont)

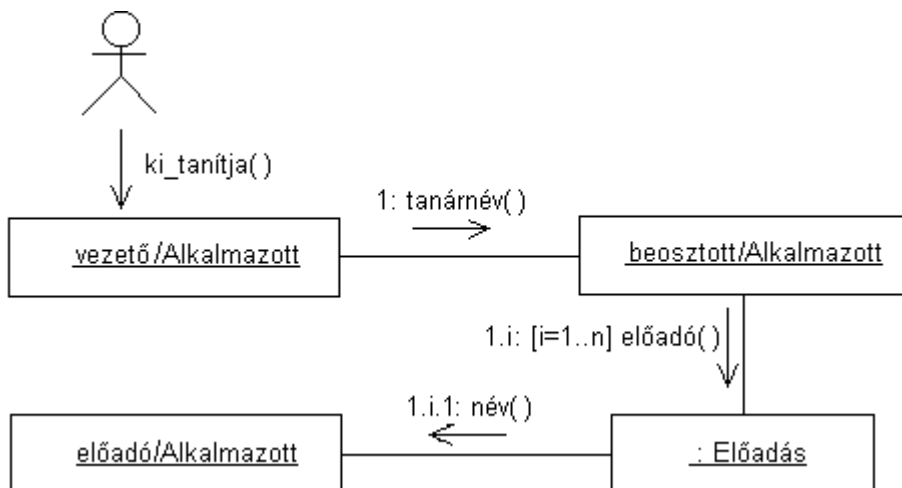
osztálydiagram (5 pont)

5.5 pont van benne
 osztályok, asszociációk 0.5-0.5 pont
 4 szerep 1 pont
 6 multiplicitás 2 pont



kollaborációs diagram (6 pont)

obj. szerepekkel 1-1 pont
 üzenetek 0.5 pont



9. Legyen egy O objektumunk, amelynek állapotát időnként menteni kell (egy Memento objektumba, annak setState() metódusát meghívva), hogy az később visszaállítható legyen (a Mementótól a getState()-tel lekérve). A Mementókat egy X objektum kezeli, amely utasítja O-t egy mentésre (CreateMemento()), vagy az átadott Memento szerinti állapot visszaállítására (SetMemento(Memento)).

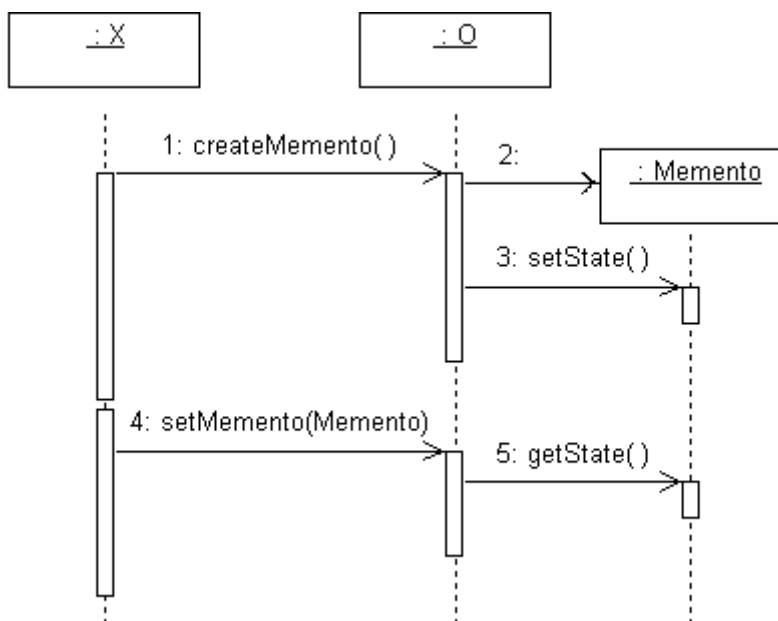
Rajzoljon **szekvencia diagramot** egy állapot kimentésére és visszaállítására ! (4 pont)

Rajzolja le az **osztálydiagramot**, az operációk feltüntetésével ! (3 pont)

Hogyan biztosítja, hogy a Mementókat őrző X az elmentett állapotot ne tudja megváltoztatni ? Minden jó megoldás 3 pont.

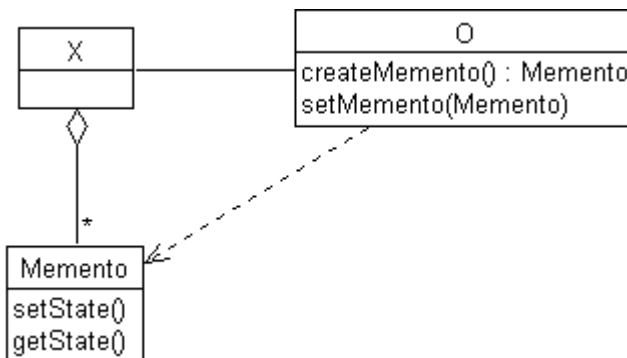
Szekvencia diagram (4 pont)

minden obj és üzenet 0.5 pont



Classdiagram (3 pont)

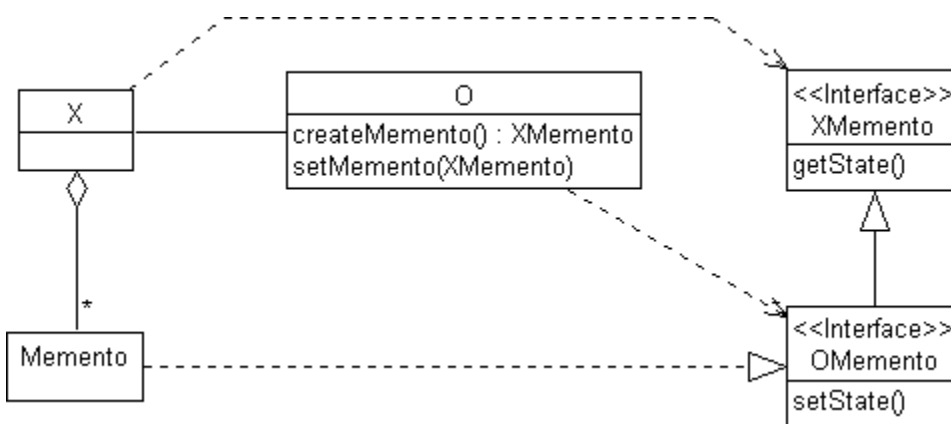
minden class és assoc 0.5 pont



Szingleton (mindegyik 3 pont)

A) Mementóban van egy boolean változó, amelyet a konstruktor false-ra állít. A setState() belépéskor megnézi, hogy ez a változó false-e. Ha igen, engedi a továbbiak végrehajtását és a változót true-ba viszi.

B) Legyen XMemento interfész, amelyből származtatjuk az OMemento interfészt úgy, hogy az OMemento interfésznek van setState() operációja.



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2001. június 26.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet ! A string karakterei előről 1-gyel kezdődően számozottak. Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveletet. (6 pont)

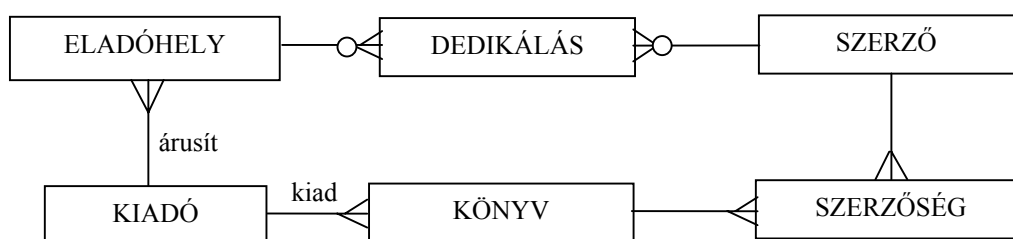
- NEW()** új (üres) stringet hoz létre.
ADD(s,x) az **s** string végére rakja az **x** karaktert.
LGTH(s) az **s** string karaktereinek számát (hosszát) adja.
DEL(s,i) az **s** string **i**-ik karakterét törli és a stringet tömöríti. Ha **i** nagyobb mint a string hossza, akkor a művelet hatástalan.
VAL(s,i) eredményül adja az **s** string **i**-ik karakterét. Ha **i** nagyobb mint a string hossza, akkor az eredmény értelmetlen (nem definiált).

LGTH(NEW()) = 0	- 0.5 pont, ha hiányzik
LGTH(ADD(s, x)) = LGTH(s) + 1	- 0.5 pont, ha hiányzik
IN(NEW(), i) = nem definiált	- 0.5 pont, ha hiányzik
DEL(NEW(), i) = NEW()	- 0.5 pont, ha hiányzik
VAL(ADD(s, x), i) = if (i < LGTH(s) + 1) VAL(s, i)	1 pont
if (i == LGTH(s) + 1) x	1 pont
if (i > LGTH(s) + 1) nem definiált	1 pont
DEL(ADD(s, x), i) = if (i < LGTH(s) + 1) ADD(DEL(s, i),x)	1 pont
if (i == LGTH(s) + 1) s	1 pont
if (i > LGTH(s) + 1) ADD(s, x)	1 pont

2. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A könyvhétre minden kiadó (neve, címe azonosítja) több új könyvet jelentetett meg. A könyvre jellemző a címe, az ára és a példányszáma; ugyanazt a könyvet nem adta ki több kiadó. Egy könyvnek több szerzője (név, cím, életkor) is lehet, egy szerző több könyvnek is lehet a (társ)szerzője. Nyilvántartjuk azt is, hogy egy szerző egy könyv írásában való közreműködésért mennyi honoráriumot kapott. A kiadók az új könyveket csak a saját eladóhelyeiken (bolt vagy sátor) árulták. A szerzők az eladóhelyeken dedikáltak. Egy eladóhelyen egy időben mindig csak egyetlen szerző dedikált. Előfordult olyan, hogy ugyanazon a napon ugyanazon a helyen egy szerző két különböző időben is dedikált, de voltak olyan eladóhelyek ahol egyáltalán nem volt dedikálás, és szép számmal voltak írók, akik nem dedikáltak egyszer sem. A dedikálások jellemzője a megjelent olvasók száma. Az eladóhelyet jellemzi még, hogy mennyi bevételt, forgalmat produkált a könyvhéten.

- KIADÓ (neve, címe)
 KÖNYV (cím, kiadó, ár, példányszám)
 SZERZŐSÉG (könyv, szerző, honorárium)
 SZERZŐ (név, cím, kor)
 DEDIKÁLÁS (szerző, eladóhely, dátum, idő, megjelentek száma)
 ELADÓHELY (cím, kiadó, fix/ideigl, forgalom)



mindegyik entitás és mindegyik reláció 0.5 pontot ér, max. 6 pont.

3. Mik a karbantartás költségeit befolyásoló technikai tényezők ? (4 pont)

- objektumok, modulok függetlensége
- programozási nyelv
- programozási stílus
- verifikáció és validáció minősége
- dokumentáció minősége

mindegyik találat 1 pontot ér, az 5. találat bonus.

4. Mi a CASE rendszerek "export/import facilities" alrendszerének feladata ? (3 pont)

A modellnek vagy egyes részeinek a mozgatása különböző modellek/CASE rendszerek között.
 Kulcskérdése az export-import formátum (egységessége)

5 Adjon **JSD** alapú megoldást az alábbi problémára ! (10 pont)

Bergengócia királya szeretné férjhez adni bitang ronda, ámde dúsgazdag leányát. A királylány kezének elnyerésére a király közbeszerzési eljárást írt ki lovagok számára. A lovagok jegyajándék küldésével jelzik részvételi szándékukat. A király megmustrálja a jegyajándékokat. Amelyik ajándékot nem tartja megfelelőnek, azt visszaküldi és az ajándékot küldő lovagot számúzi Szibériába. A megfelelő ajándékot küldő lovagok díszes kardot kapnak a királytól, és részt vesznek a királylány kezéért folyó lovagi tornán. A tornát körmérkőzéses rendszerben lebonyolított párviadatok alkotják. Egy párviadal mindig az egyik fél győzelmével ér véget. Minden párviadal után mindkét harcos lovagot a királyi kirurgusok megvizsgálják, és ha a lovag sérüléseit súlyosnak találják, akkor a lovagot visszaléptetik a további küzdelmektől. Valahányszor a lovag veszít egy párviadalon, annyiszor jogot szerez arra, hogy a torna tartama alatt bármikor elküldhet egy baglyot, hogy az a Burkus király kertjéből lopjon egy aranydiót. Mivel az aranydiót termő fát ördögfiókák őrzik erősen, a baglyok gyakorta aranydió nélkül térnek vissza, esetenként vissza sem jönnek. A körmérkőzés eredményeinek és a beérkezett aranydiók száma alapján a győztes lovag meghatározható.

Definiálja az alábbi funkciókat ellátó processzeket:

Párviadal végével illetve aranydió érkezésével azonnal készítsen rangsort a pillanatnyi állásról !

Kérésre adja meg a számúzott és a visszaléptetett lovagok listáját !

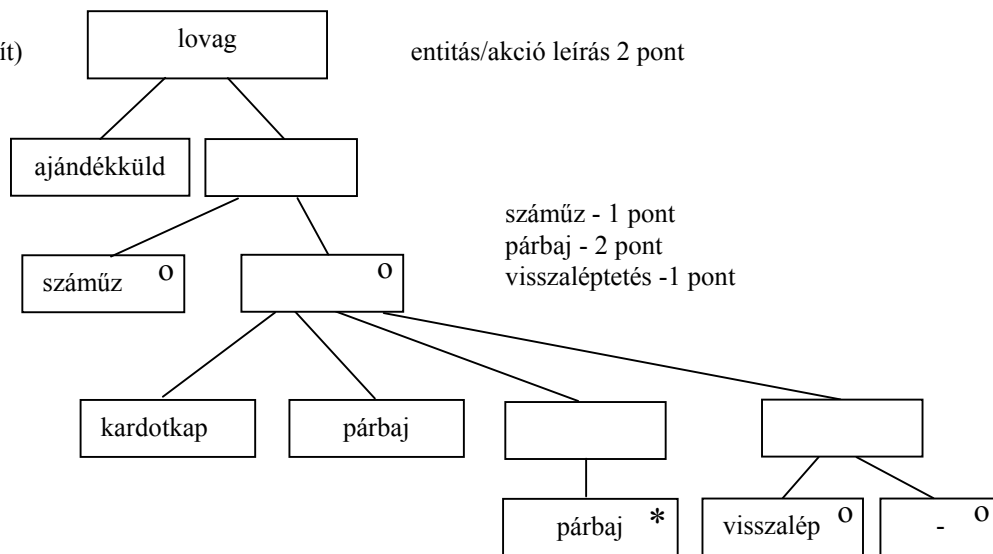
Kérésre adja meg, hogy egy loagnak hány baglya van még úton. !

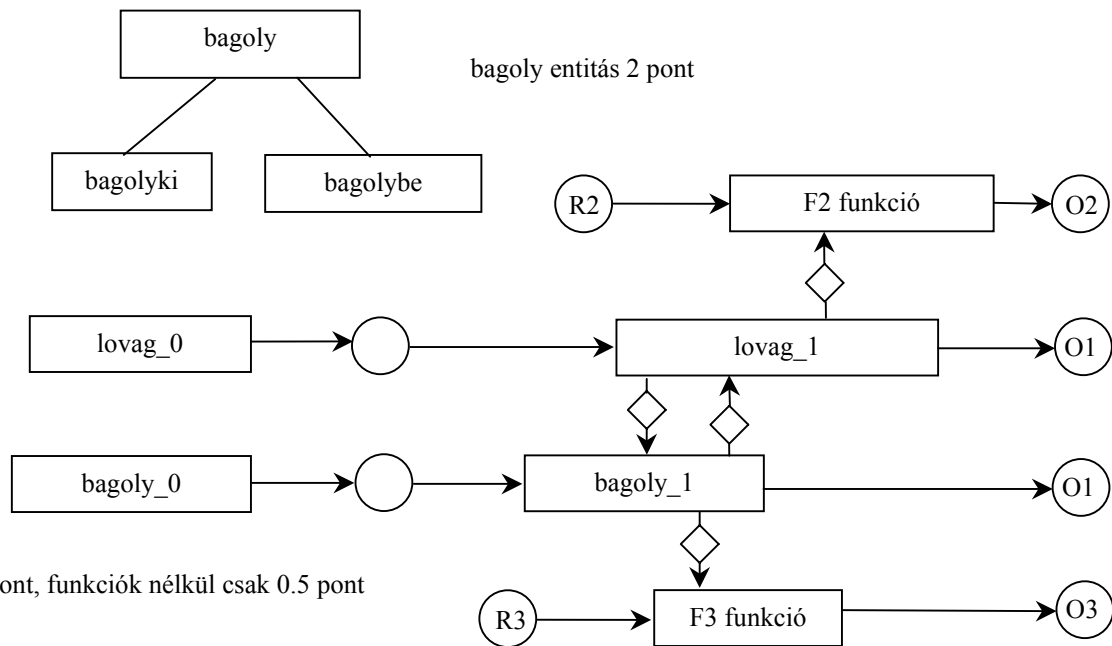
Készítse el az

**entitás/akció leírást, az entitások attribútumaival,
 az entitás struktúra diagramot,
 az initial modelt,
 funkció processzeket.**

kapcsolja a modellhez a

- ajándékküld (lovag)
- számúz (lovag)
- kardotkap (lovag)
- párbajozik (lovag) (nyer/veszít)
- visszalép (lovag)
- bagolyki (bagoly, lovag)
- bagolybe (bagoly, vandió)



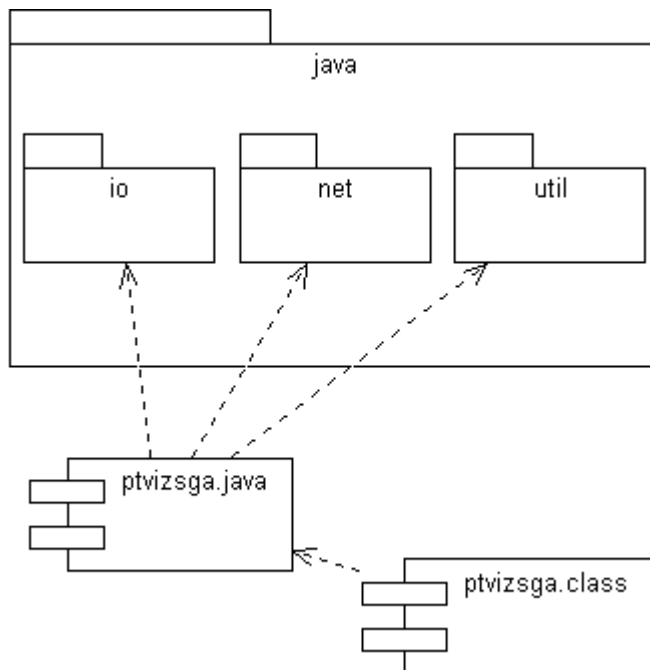


6. Legyen egy java alkalmazás, amelynek egy részlete az alábbiakban látható:

```
import java.io.*;
import java.net.*;
import java.util.*;
public class ptvizsga {
    public static void main(String args[]) {
        ...
    }
}
```

Rajzolja fel a program **UML komponens diagramját** ! (8 pont)

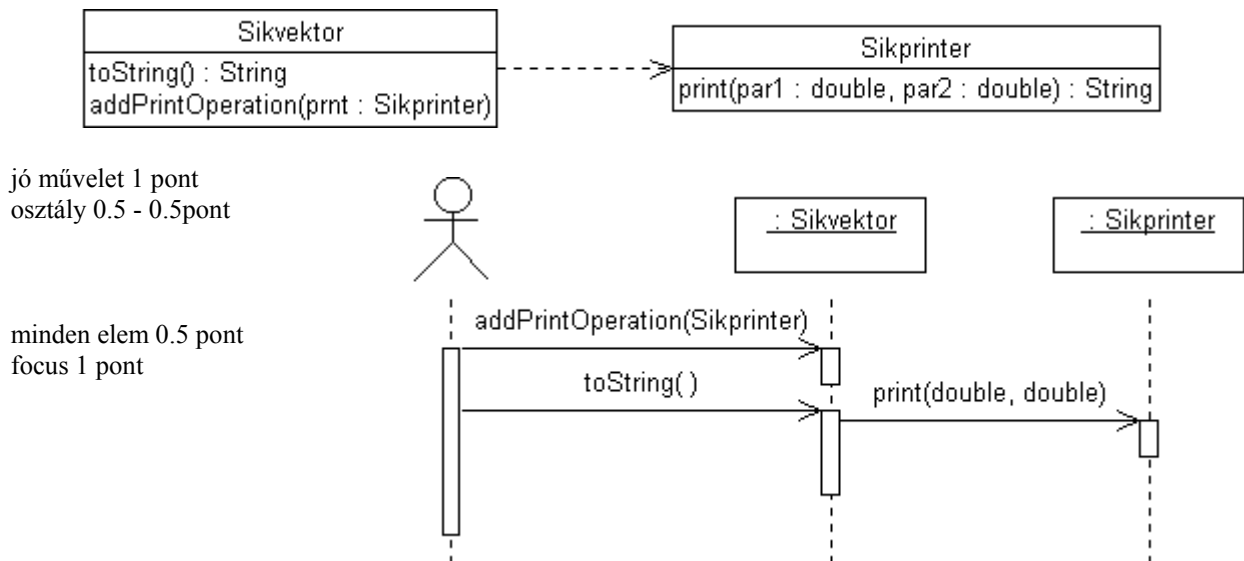
minden elem 1 pont
kapcsolat 0.5 pont



7. A síkbeli vektorok modellezésére legyen egy `Sikvektor` osztályunk, amelynek egyebek mellett legyen egy `toString()` metódusa a vektor kiírására. Definiálja úgy a `Sikvektor` osztályt, hogy annak újrarendezése nélkül, a felhasználó tudjon tetszőleges formátumban kiírni.

Rajzoljon **UML osztálydiagramot** az operációk feltüntetésével! (4 pont)

Rajzoljon **UML szekvencia diagramot** arra az esetre, amikor a felhasználó futás közben megváltoztatja a formátumot, majd a változtatott formátumban kiírja a számot! (4 pont)

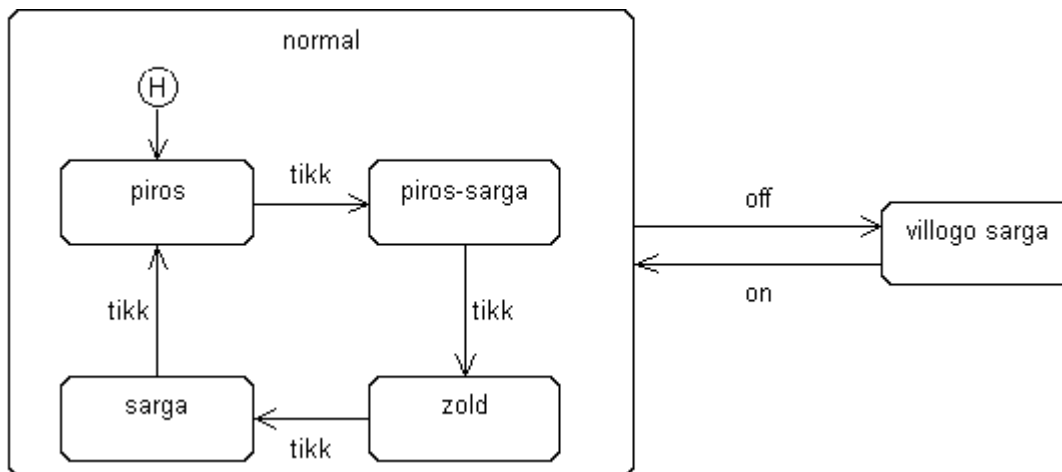


jó művelet 1 pont
osztály 0.5 - 0.5pont

minden elem 0.5 pont
focus 1 pont

8. Egy közlekedési lámpa a beépített óra jeleinek hatására a szokásos módon rendre pirosra, piros-sárgára, zöldre, sárgára majd ismét pirosra vált. A lámpa szekrényében elhelyezett kapcsolóval a lámpa átállítható felfüggesztett módra, amikor sárgán villog. Ugyanezen kapcsolóval visszaállítható a normál működés. A lámpa, mikor visszakapcsolják, a felfüggesztés előtti színnel kezdi a működést.

Rajzoljon **UML state-chartot** a probléma leírására! (5 pont)



villogó sárga +on-off 1 pont, normál 1 pont, ciklus 1 pont, H 2 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

2001. december 18.

1. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (6 pont)

A LÓERŐ című autós magazin havonta 120 oldalon jelenik meg. Egy számban (kiadás éve, hónapja, sorszáma) cikkek (cím, karakterek száma) találhatóak. A szerkesztőség tagjait név, e-mail cím és foglalkozás (szerkesztő, fényképész, riporter, tudósító, stb.) szerint tarjuk nyilván. Megőrizzük, hogy egy-egy cikk vagy fénykép után a szerző mennyi honoráriumot kapott. Egy cikknek több szerzője is lehet. Egyes cikkekhez fényképek (hol, mikor készült, mit ábrázol) is tartozhatnak, de nem minden elkészült fénykép jelenik meg. Cikket és fényképet sohasem ismételnék. A szerkesztőség rendszeresen kiállításokat (hely, nyitás, zárás ideje, látogatók száma) szervez, amelyen a magazinban megjelent legjobb, korábban ki nem állított képeket mutatják be.

SZÁM (év, hó, sorszám)

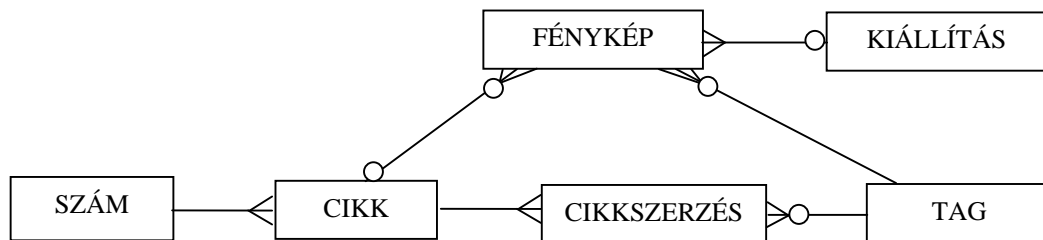
CIKK (cím, szám, karszám)

TAG (név, email, foglalkozás)

CIKKSZERZÉS (tag, cikk, honorárium)

FÉNYKÉP (hol, mikor, mit, tag, cikk, honorárium, kiállítás)

KIÁLLÍTÁS (hely, nyitás, zárás, látogatók)



mindegyik entitás és mindegyik reláció 0.5 pontot ér, max. 6 pont.

2. Melyek a hagyományos komponens rendszerek ? (2 pont)

.....

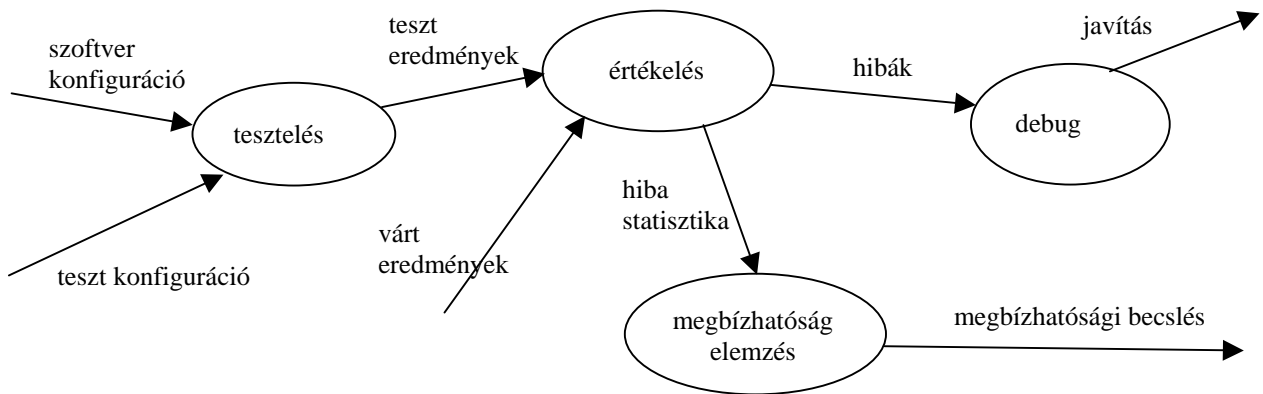
.....

.....

UNIX
XML
modul
osztályok és öröklés
keretrendszerek

(minden jó válasz 0.5 pont, minden hiba -0.5 pont, 5 jó válasz: 2.5)

3. DFD ábra segítségével definiálja a tesztelés információs folyamatát ? (7 pont)



Pontozás: mindegyik processz 1 pontot, mindegyik adatút 0.5 pontot ér, max. 7 pont (8 pont van benne !!!)

4 Adjon **JSD** alapú megoldást az alábbi problémára ! (10 pont)

Bergengőcfalván hócsatával egybekötött Karácsonyi bált rendeznek. A báli belépőt 1 hónappal előre kell megvásárolni. Minden belépővel rendelkező ezután adományokat ajánlhat fel (többet is) a hócsaták legjobbjainak. Hócsatákra a bál kezdetén, a helyszínen lehet (nem kötelező) benevezni. Több csatát tartanak, mindenki, aki nevezett, bármelyik csatában részt vehet védőfelszereléssel vagy anélkül. A rendezőség minden csata után a résztvevőket minősíti. A benevezettnak bármikor jogában áll védőfelszerelést (vaskályha ajtó a mellkas védelmére, bili a fejtálatok ellen) kérni, amit legkésőbb a bál végén vissza kell szolgáltatni. Aki egyszer már leadta a védőfelszerelést, az ismételten nem kérhet. A bál végén minden bálozó emléklapot kap, amelyen felsorolják a hócsatában végrehajtott hőstetteit és a kiválóak részesednek az adományokból.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha valaki másodszor akar védőfelszerelést kérni !

Kérésre adja meg azokat, akik adományokat ajánlottak fel és van náluk védőfelszerelés !

Készítse el az

**entitás/akció leírást, az entitások attribútumaival,
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a funkció processzeket.**

belépőt vesz (bálozó)

eseménylista 2 pont

felajánl (bálozó)

benevez (bálozó, felszerelés)

csatázik (bálozó)

minősítik (bálozó)

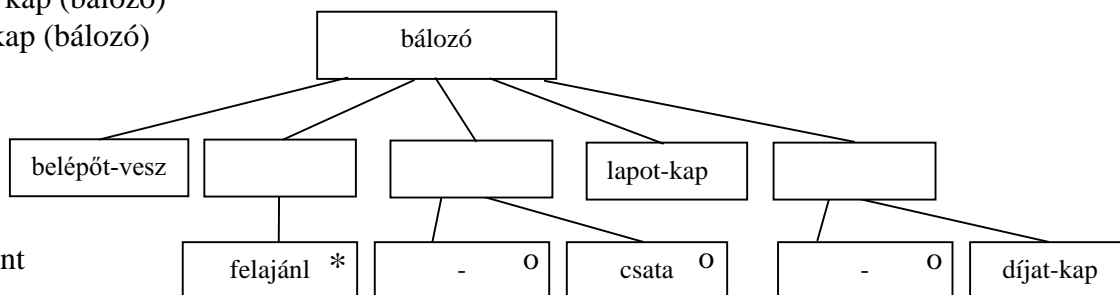
felszerelést felvesz (felszerelés, bálozó)

felszerelést lead (felszerelés, bálozó)

emléklapot kap (bálozó)

adományt kap (bálozó)

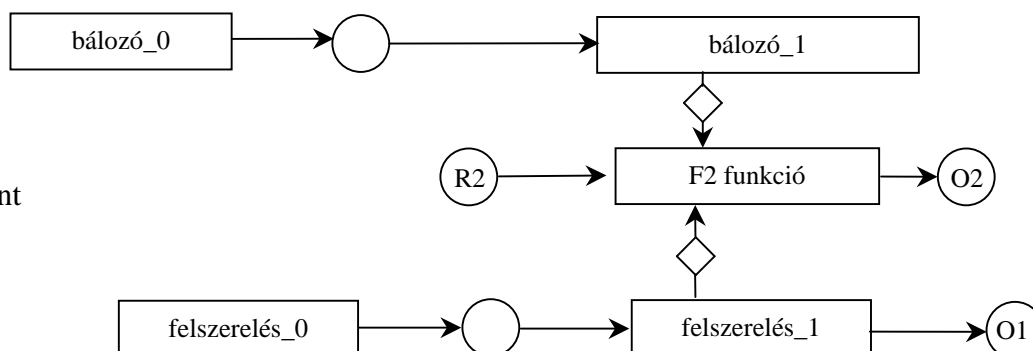
bálozó entitás 5 pont



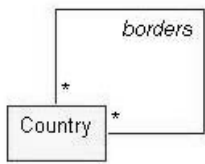
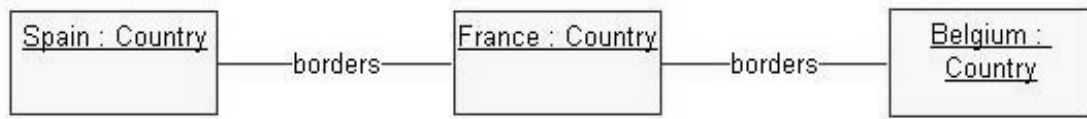
felszerelés entitás 2 pont



teljes SSD 1 pont



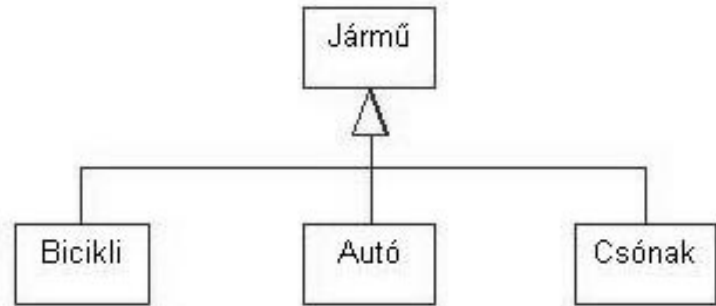
5. Rajzoljon UML osztálydiagramot az alábbi objektumdiagram alapján ! (4 pont)



2 az osztály, 2 az asszociáció

6. Jelölje meg az alábbi osztálydiagramra vonatkozó helyes állításokat ! (3 pont)

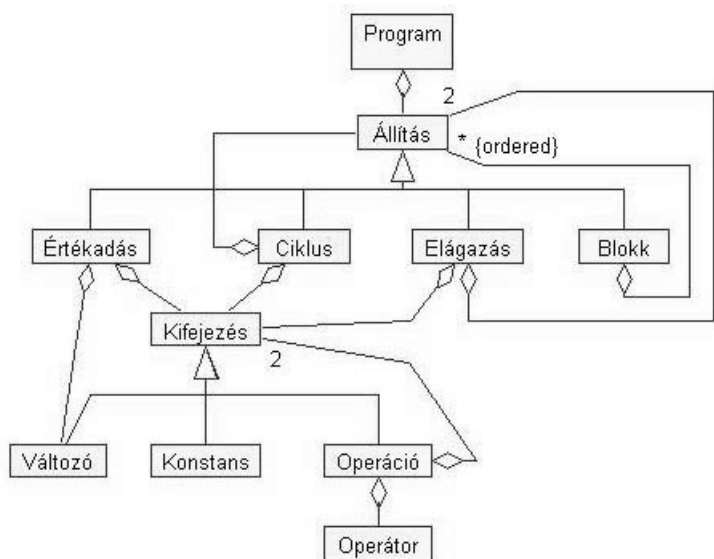
- Autó helyettesíthető Biciklivel
- Bicikli helyettesíthető Autóval
- Autó helyettesíthető Járművel
- Jármű helyettesíthető Csónakkal
- Csónak helyettesíthető Járművel
- Jármű helyettesíthető Autóval



Megoldás : 4, 6.

Minden jó 1.5 pont, minden hiba –1 pont

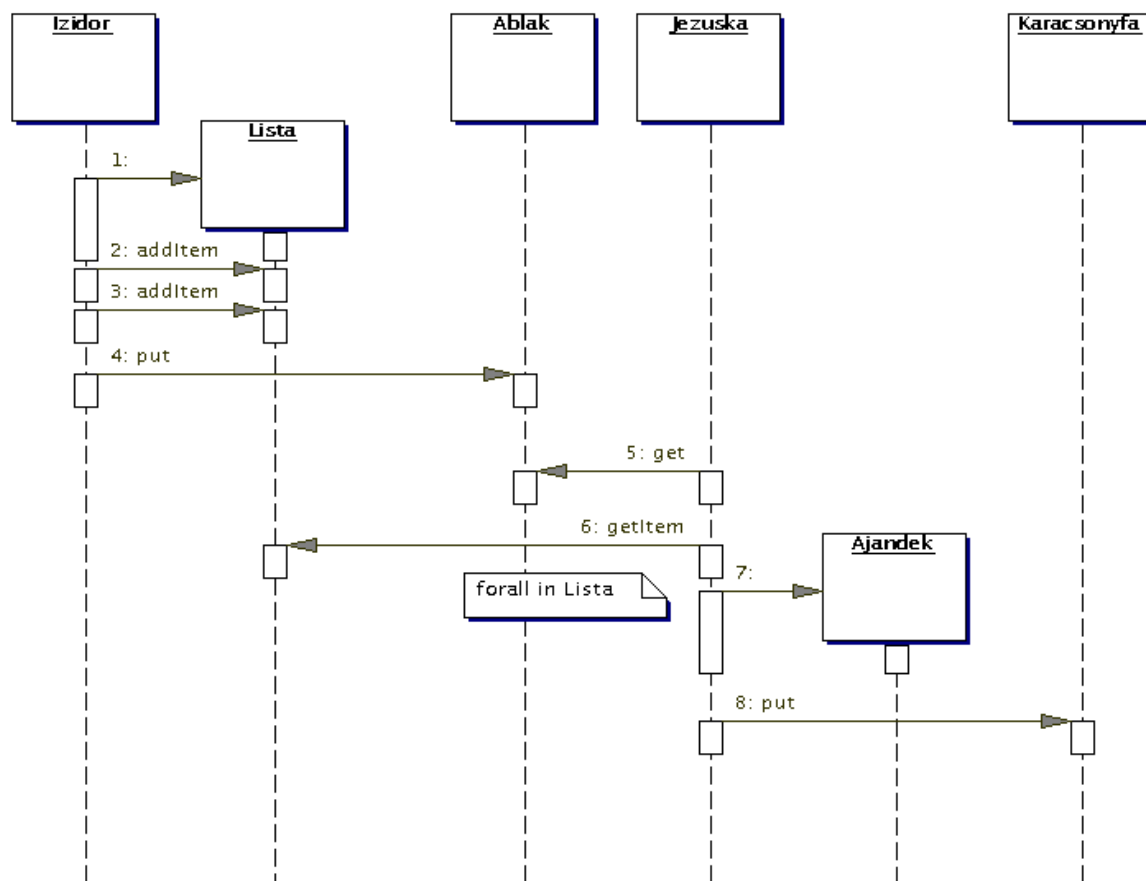
7. Egy egyszerű struktúrált programnyelven megírt program a következő módon épül fel.
 A program egy állításból áll. Az állítás lehet értékadás, blokk, ciklus vagy elágazás. Az értékadás bal oldalán egy változó, a jobb oldalán egy kifejezés állhat. A kifejezés lehet egy változó, egy konstans vagy egy binér operátorból álló operáció. Az operátornak két kifejezés van a két oldalán.
 Az elágazás egy kifejezésből és két állításból (then ág és else ág) áll. A ciklus egy kifejezésből és egy állításból áll, mely utóbbi a ciklusmag. A blokk állítások rendezett halmaza.
 Rajzoljon **UML class diagramot** (9 pont)



minden összeköttetés 0.5 pont + ordered

8. Rajzoljon az eseményekről **UML szekvencia-diagramot** ! (9 pont)

Izidor jó fiú volt idén. Decemberben el is készítette az idei karácsonyi kívánságlistát, amelyet az ablakba tett ki. Jézuska, aki ilyen tájt esténként a jó gyerekek ablakait figyeli, elviszi az ablakból a listát, és a rajta szereplő kívánságoknak megfelelő ajándékokat elkészíti, megveszi vagy elővarázsolja. Szenteste Jézuska a karácsonyfa alá teszi a meglevő ajándékokat.



objektum 1 pont, üzenet 0.5 pont

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

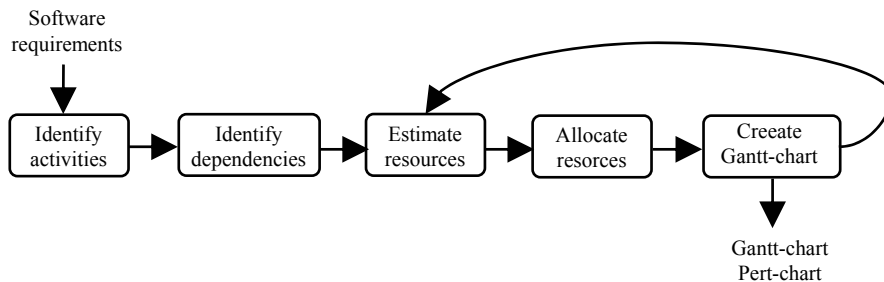
VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból
2005. január 18.

1. Írja fel egy az objektum operációja szignatúrájának általános formáját ! (3 pont)

[oneway] <op_type_spec> <identifier> (param1, ..., paramL) [raises(except1, ..., exceptN)]
[context(name1, ..., nameM)]

minden helyes item 0,5 pont

2. Hogyan származtatja a szoftver követelményekből az ütemezéssel kapcsolatos Gantt és Pert diagramokat ?
Rajzolja fel a folyamat adatfolyamábráját ! (5 pont)



minden helyes gombóc 1 pont

3. A Petri-hálót egy 4 elemű algebrai struktúra (P, T, A, M) írja le. Definiálja (algebrailag és szövegesen) az egyes elemeket ! (5 pont)

$P = \{P_1, \dots, P_n\}$ –

place-ek halmaza

$T = \{T_1, \dots, T_m\}$ –

transition-ok halmaza

$A \subseteq \{(P \times T) \cup (T \times P)\}$ –

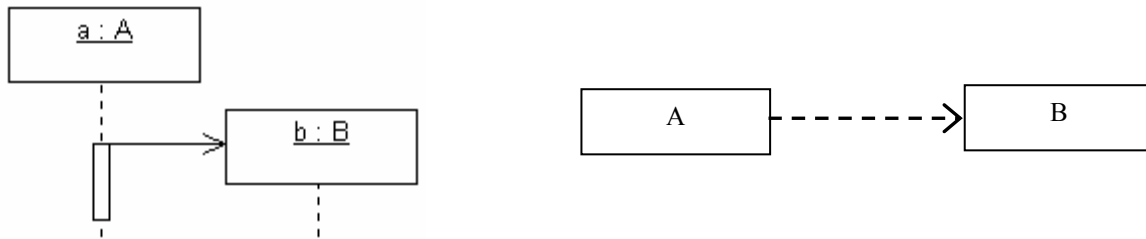
élek halmaza

$M = P \rightarrow \mathbb{N}$ –

markerezés

minden helyes kifejezés 1 pont, min. 3 jó komment 1 pont

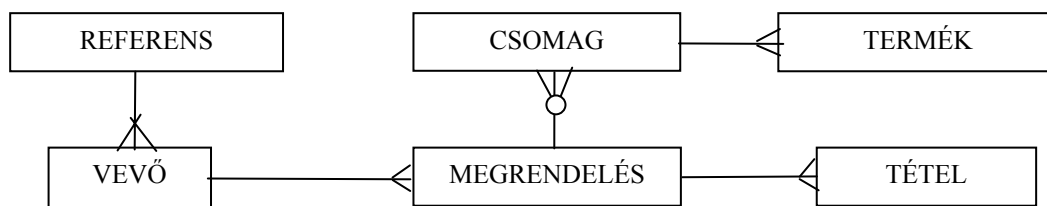
4. Az A osztályú objektum egy példánya létrehoz egy B osztályú objektum példányt. Jelölje A és B közötti kapcsolatot UML szekvencia és osztálydiagramon. (4 pont)



Egy-egy diagram 2-2 pont

5. Készítsen **entitás-relációs modellt** az alábbi problémára! (6 pont)

Egy csomagküldő szolgálatnál referensek (név, beosztás, gyakorlat) intézik a vevőktől (név, cím, adószám) beérkező megrendeléseket. Minden új vevőhöz hozzárendelnek egy referenst, ő felel a továbbiakban a vevőért, egy referens sok vevővel tart kapcsolatot. A megrendelés (dátum, szállítási cím) legalább egy tételt tartalmaz. Egy tétel áll a megrendelt dolog megnevezéséből, cikkszámából és a kívánt darabszámból. A tételeknek megfelelő termékeket (cikkszám, egyedi gyári szám, gyártó) a csomagküldő hosszabb-rövidebb idő alatt beszerzi, majd amikor minden termék beérkezett, azokat csomagokba rendezi és elküldi a vevőnek. Egy megrendeléshez több csomag is tartozhat, de egy csomag mindig egyetlen megrendeléshez kapcsolódik. Egy csomagban egy vagy több megrendelt termék kerül leszállításra.



mindegyik entitás és mindegyik reláció 0.5 pontot ér. 3 pont fölött +0.5 pont.

6. Készítsen **JSD** modellt az alábbi problémára ! (11 pont)

A TV23974583 (röviden TV2) évente megrendezett Gigastar c. vetélkedőjének célja a legjobb gégével (gigával) rendelkező álompolgárok kiválasztása. A vetélkedés gyorsasági sörivásban zajlik. A vetélkedőt a kezdet előtt egy hónappal az interneten hirdetik meg, amire az álompolgárok jelentkeznek. A versenyt nem indítják, ha a jelentkezők száma kevesebb 1000-nél. A sörivó verseny kétfordulós. Az első fordulóból csak az indulók legjobb 30 %-a jut tovább, a többiek kiesnek. A verseny végén díjazást kap a legjobb 100 eredményt elérő álompolgár. Aki korábban már háromszor kiesett vagy két alkalommal díjazták, azt a további vetélkedésekből kizárják, következő vetélkedőkre már nem is jelentkezhet. Új vetélkedőt csak az előző befejezése után kezdenek.

Definiálja az alábbi funkciókat ellátó processzeket:

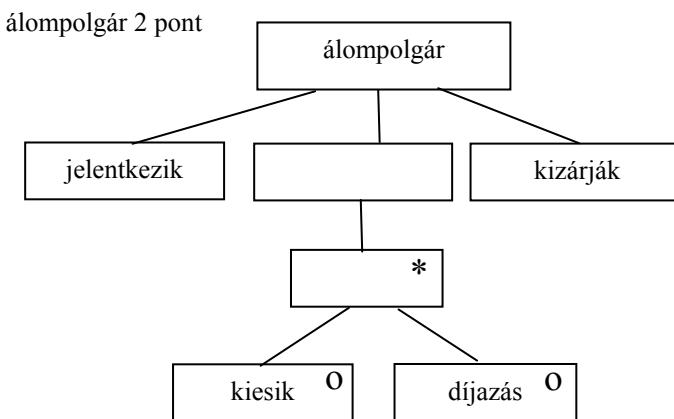
Kérésre készítse el a kizárt álompolgárok listáját !

Kérésre készítse el azok listáját, akik olyan vetélkedőre jelentkeztek, amelyet végül nem tartottak meg és a jelentkező korábban sosem esett ki és díjazást sem kapott.

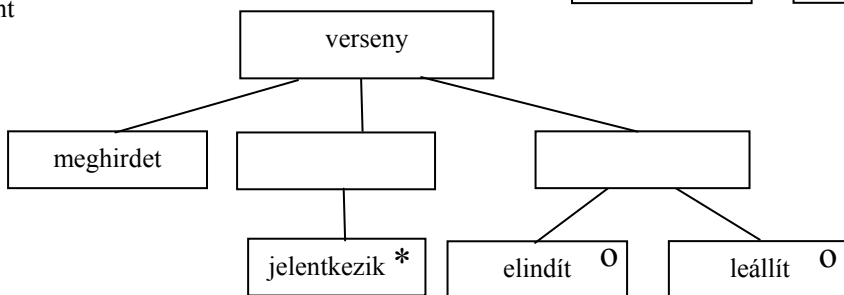
Kérésre adja meg azok listáját, akik egy adott vetélkedőn az első fordulóban a legjobb 100 között voltak, de végül nem lettek díjazottak.

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
 kapcsolja a modellhez a **funkció processzeket.**

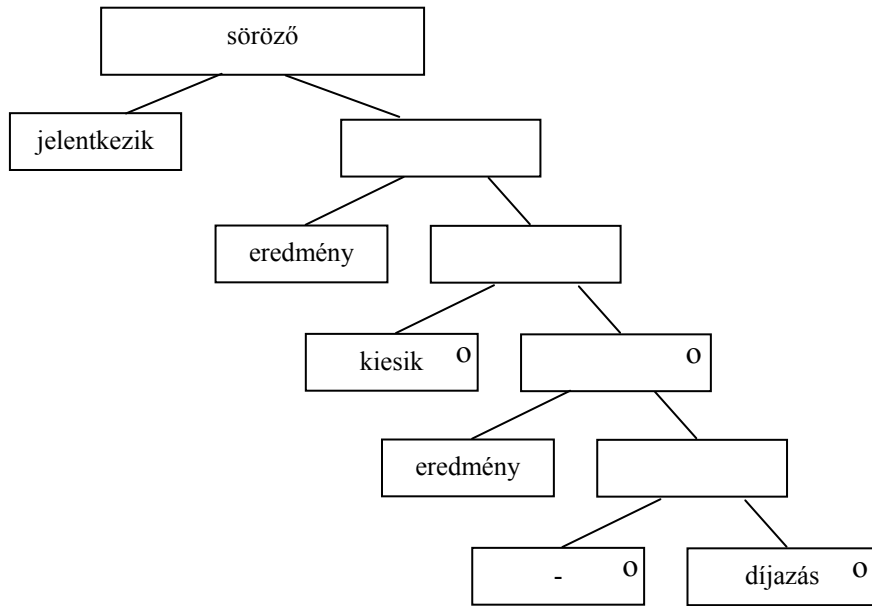
- meghirdet (verseny)
 - jelentkezik (verseny, álompolgár, söröző)
 - elindít (verseny)
 - leállít (verseny)
 - eredmény (söröző)
 - kiesik (söröző, álompolgár)
 - díjazás (söröző, álompolgár)
 - kizárják (álompolgár)
- eseménylista 2 pont



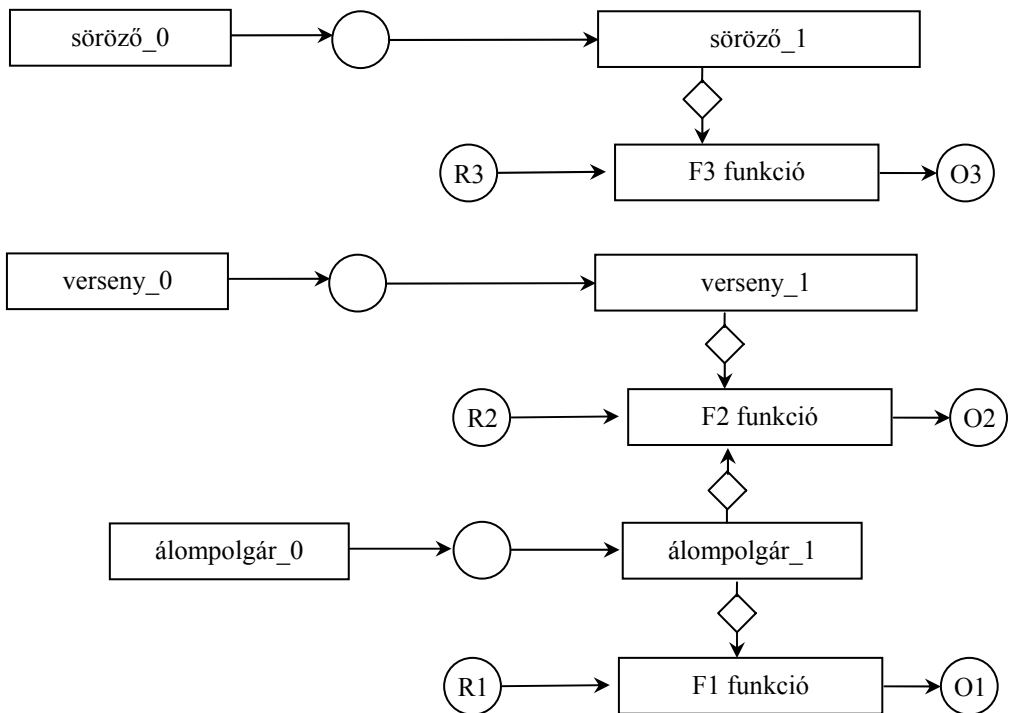
verseny 2 pont



söröző 3 pont

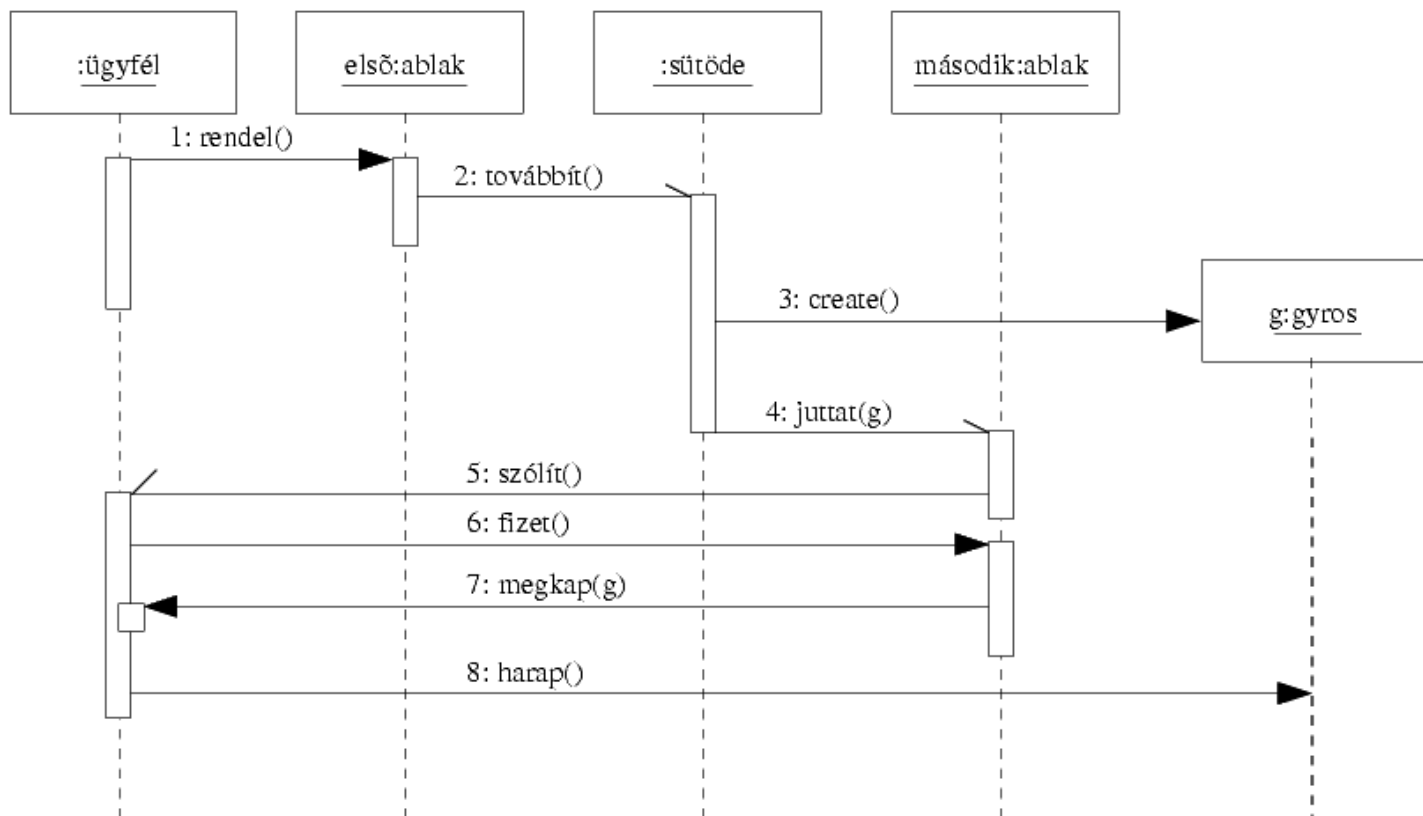


teljes SSD 2 pont



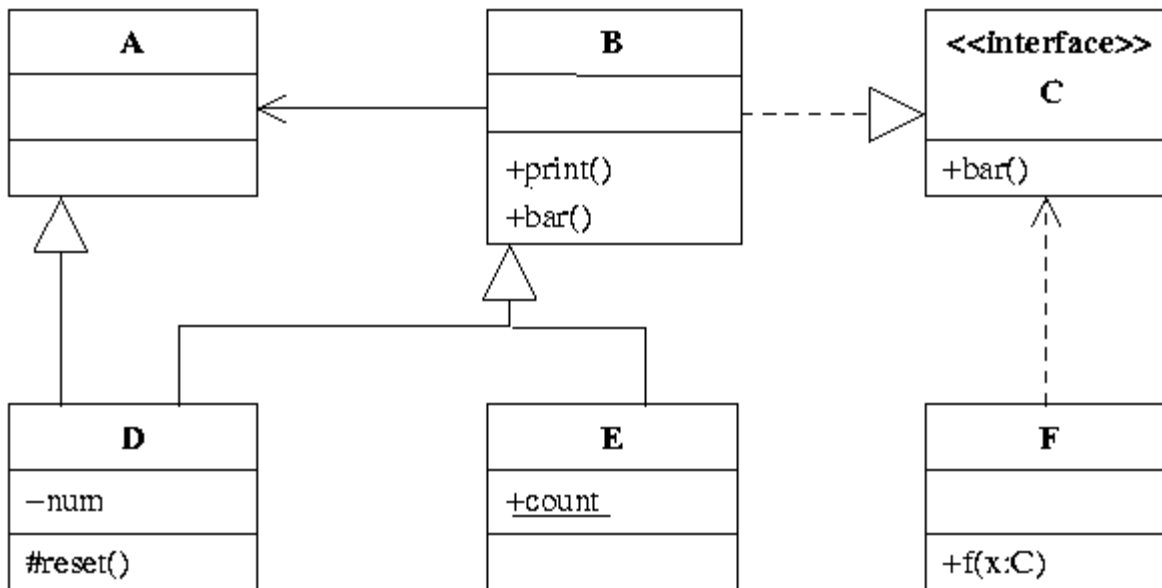
7. Rajzoljon az alábbi történethez UML szekvenciadiagramot! (8 pont)

Az Ibrahim Kayalda étteremlánc biciklistákat szolgál ki. A vevő az első ablaknál leadja gyros-rendelését, majd mikor ezt továbbították a ház belsejében levő sütödének, továbbhalad a második ablakhoz, ahol FIFO sorban szolgálják ki az ügyfeleket. Mikor elkészül egy gyros, a sütőde a második ablakhoz juttatja, ahol a soron következő ügyfelet szólítják, majd folytatják munkájukat. A szólításra az ügyfél fizet, mire megkapja a rendelt ételt, amibe jóízűen beleharap.



minden jó objektum és üzenet 0,5 pont, kivéve: sütőde 1 pont, gyros create 1 pont, juttat aszinkron 1 pont.

8. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)
 (F - az ki van zárva, mert nem tudom)

- [B] D helyettesíthető E-vel, mert közös az ősük.
 [B] E ismeri D-t, mert egyik őse (B) ismeri D egyik ősét (A-t).
 [E] F foo() metódusa nem kaphat paraméterül D-t, mert D nem valósítja meg C interfészt.
 [E] A ismeri B-t, mert B aggregálja A-t.
 [D] F meghívhatja B bar() metódusát, mert B megvalósítja a C interfészt.
 [B] A megvalósítja C interfészt, mert B-vel közös leszármazottja van.
 [B] E-t nem lehet példányosítani, mert van statikus attribútuma.
 [B] D reset() metódusából nem érjük el D num attribútumát, mert a num privát.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

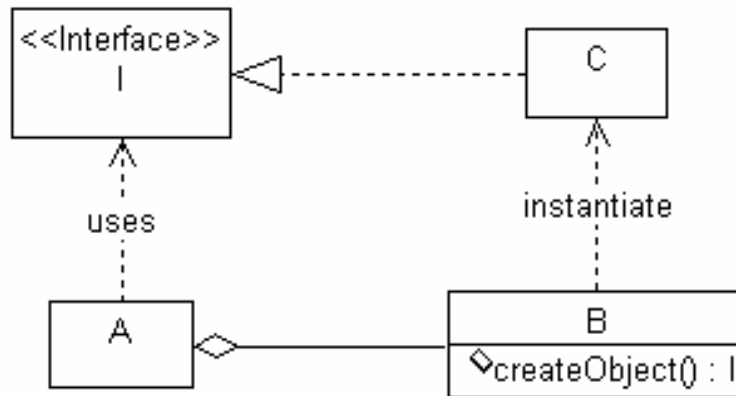
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2005. május 31.

1. Az A osztályú objektum használni akar egy I interfészt megvalósító objektumot, de A nem ismer olyan osztályt, amelyik implementálná I-t. A-nak van egy B osztályú komponense. A C osztály megvalósítja I-t. A meghívja B `createObject()` metódusát, amely visszaadja az általa konstruált C osztályú objektumot. Rajzoljon **UML struktúra diagramot** ! (8 pont)



minden helyes item 1 pont

2. Nevezze meg és jellemezze 1-1 mondattal a RUP életciklus modelljének fázisait ! (6 pont)

Inception phase: establishing the business rationale for the project, and decide on the scope of the project. Specifying the project vision.

Elaboration phase: planning the necessary activities and required resources; specifying the features and designing the architecture.

Construction phase: building and testing a working system according to the previously elaborated plans as a series of incremental iterations.

Transition phase: supplying the product to the user community (manufacturing, delivering, and training)

minden helyes item 1,5 pont

3. Készítsen **entitás-relációs diagramot** az alábbi problémára ! (5 pont)

Egy videotékában DVD-eket lehet kölcsönözni. A DVD-kről nyilvántartják a címét, a kiadóját, a kiadás évét, a műsor hosszát, valamint kategóriáját (pl.: zene, akció, romantikus, vígjáték, krimi, thriller, sex, stb.). Egy DVD-ből több példány is lehet, de olyan DVD-k adatait is tárolják, amelyekből még vagy már nincs példány. Példányonként tárolják a vásárlás idejét, az árat, a szállítót és az adott példány kölcsönzéseinek számát. A szállítóról ismerik a nevét, címét és bankszámla számát. Az ügyfeleket első kölcsönzésük alkalmával veszik nyilvántartásba. Az ügyfelet azonosító jellemzi, de nyilvántartják a nevet, címet, telefonszámot és az éves tagdíj lejárat dátumát is. Egy kölcsönzés alkalmával egy ügyfél egy DVD példányt megadott határidőig kölcsönbe vesz. A kölcsönzés megszűnik, ha az ügyfél a DVD-t visszahozta. Az ügyfelekről profilt készítenek, amely tartalmazza, hogy az ügyfél az élete során a különböző kategóriájú DVD-kből hányszor kölcsönzött (Ennek alapján lehetséges az ügyfeleknek személyre szóló reklámokat küldeni).

DVD (cím, kiadó, kiadási év, hossz, kategória)

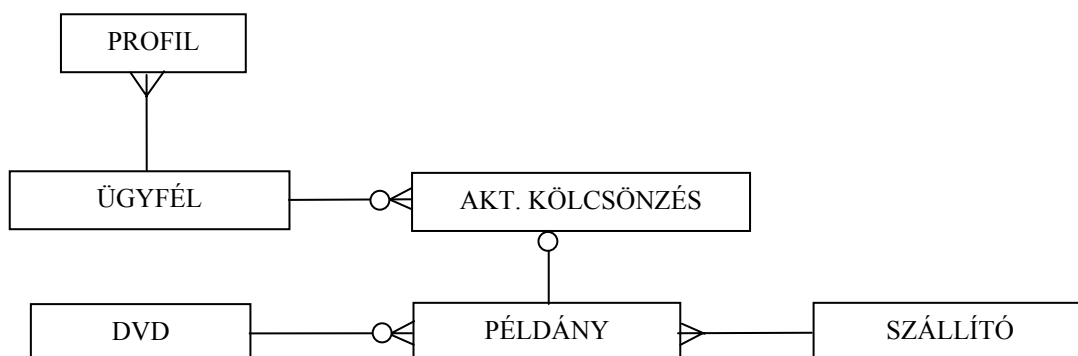
PÉLDÁNY (dvd, azonosító, vásárlás ideje, ára, kölcsönzések száma, *szállító*)

SZÁLLÍTÓ (név, cím, bankszámlaszám)

ÜGYFÉL (azonosító, név, cím, telefon, tagdíj lejárat)

AKT. KÖLCSÖNZÉS (példány, ügyfél, dátum, határidő)

PROFIL (ügyfél, kategória, kérések száma)



minden helyes item 0,5 pont, van benne 0,5 plusz pont is

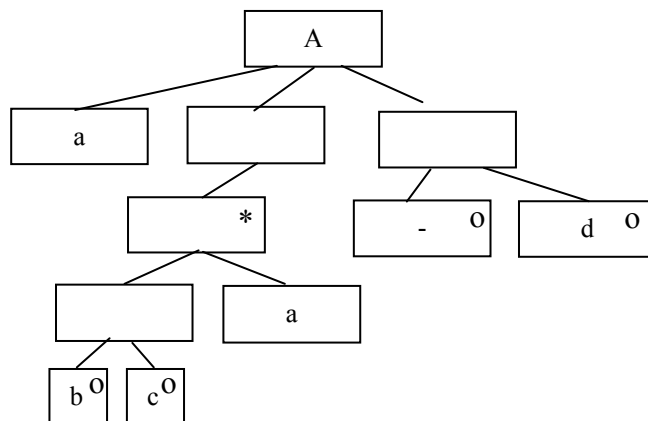
4. Sorolja fel a **JSD** 6 lépését ! (6 pont)

- | | |
|--|--|
| 1. Entitás akció (entity action) | 2. Entitás élettörténet (entity structure) |
| 3. Kiinduló modell (initial model) | 4. Funkció (function) |
| 5. Időzítés (timing) | 6. Megvalósítás (implementation) |

minden helyes válasz 1 pont

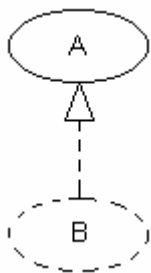
5. Legyen egy X entitás, amelyen az a, b, c és d események fordulhatnak elő. Rajzolja fel az X entitás élettörténetét a JSD szerint, ha az események sorrendjét az alábbi BNF leírással definiáltuk ! (6 pont)

$a \{ [b|c] a \} [d]$



szekvencia, iteráció, szelekció 2-2-2 pont

6. Mi A és B az alábbi UML diagramon ? (2 pont)



- | | |
|--|--|
| <input type="checkbox"/> operáció | <input checked="" type="checkbox"/> kollaboráció |
| <input type="checkbox"/> állapot | <input type="checkbox"/> metódus |
| <input checked="" type="checkbox"/> use-case | <input type="checkbox"/> beágyazott állapot |
| <input type="checkbox"/> sztereotípa | <input type="checkbox"/> komponens |

minden helyes válasz 1 pont, hiba -1 pont, negatív nem lehet

7. Egy C nyelven írt `init()` függvény inicializálja a fájlkezelést és az üzenetbuffert. Milyen a függvény kohéziója ? (2 pont)

- | | |
|--|--|
| <input type="checkbox"/> procedurális | <input checked="" type="checkbox"/> temporális |
| <input type="checkbox"/> szekvenciális | <input type="checkbox"/> funkcionális |

helyes válasz 2 pont, minden rossz -1 pont, de ≥ 0

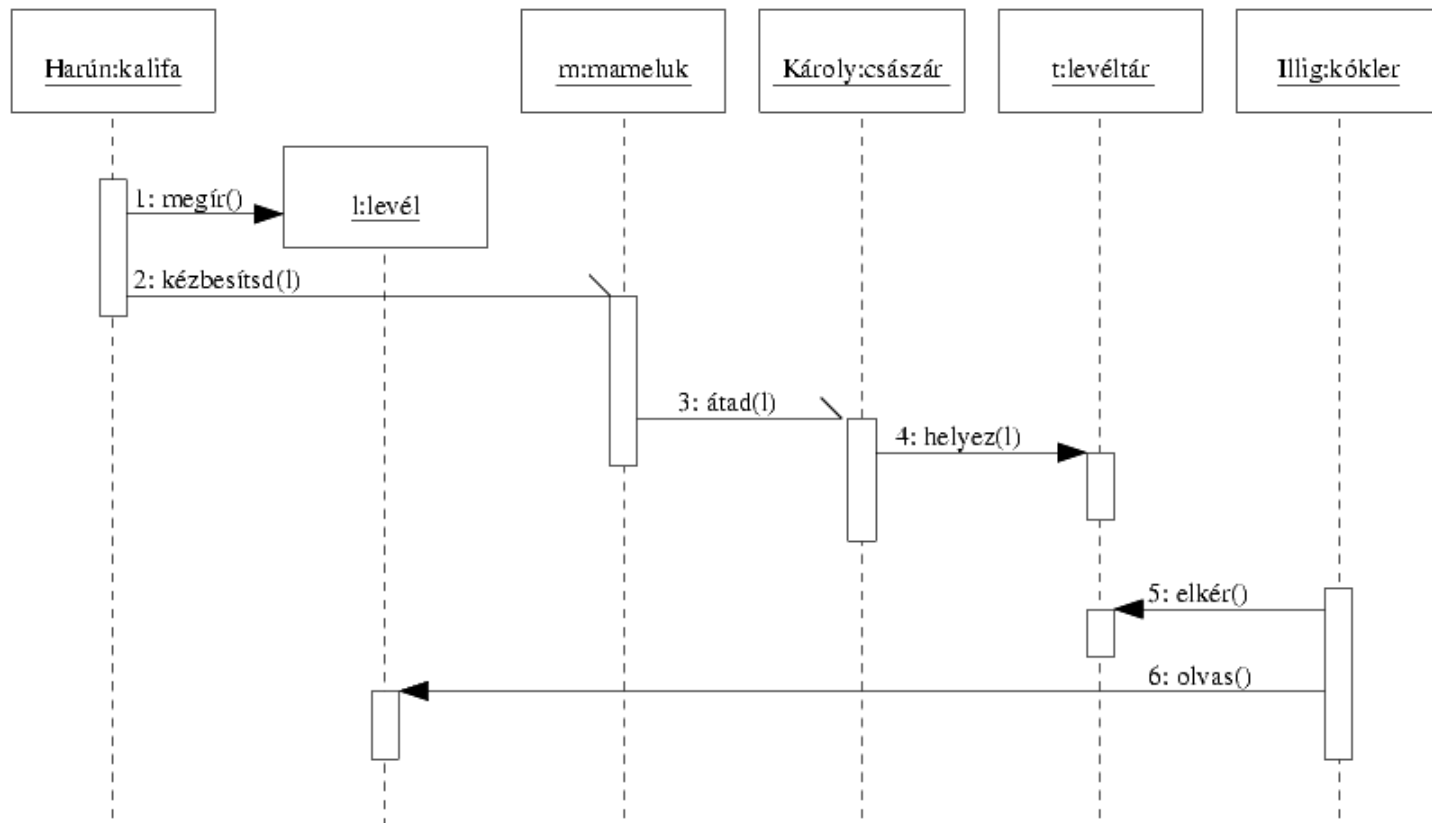
Nevezzen meg legalább egy olyan kohéziós típust, amely a fentiek között nem szerepel ! (1 pont)

kommunikációs vagy **logikai** vagy **esetleges**.....

első helyes 1 pont, minden további helyes 0.5 pont, rossz -1 pont, de total ≥ 0

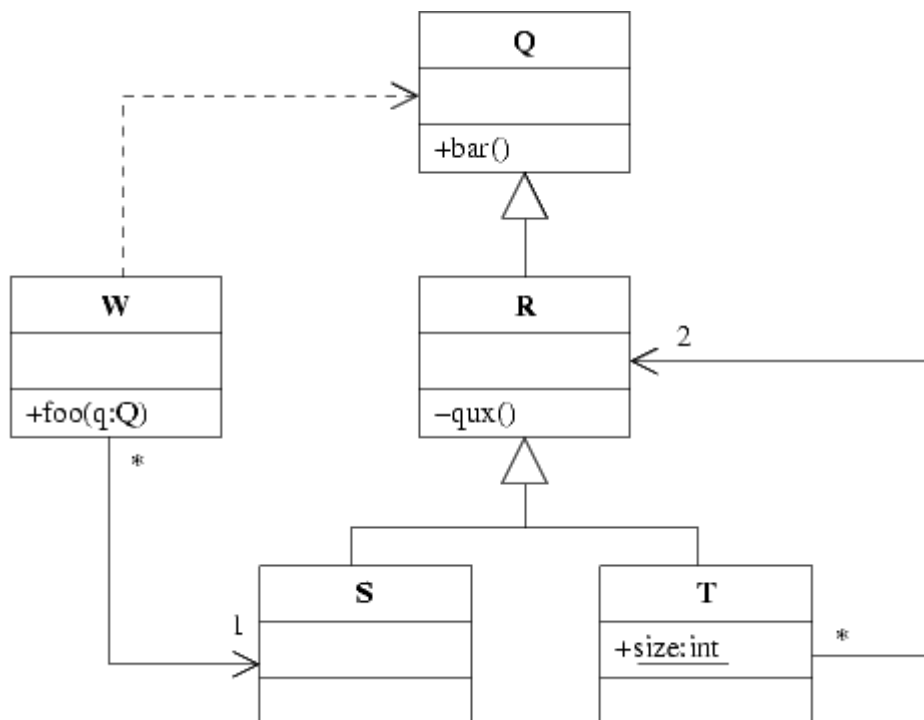
8. Készítsen UML szekvencia-diagramot a következő történet alapján! (6 pont)

Harún ar-Rasíd ír egy levelet Nagy Károlynak, melyben elefántja egészsége után érdeklődik, majd odaadja egyik mamelukjának, hogy kézbesítse. A mameluk kalandos útja végén átadja a levelet a címzettnek, majd hazaindul. Nagy Károly a levelet a levéltárba helyezteti. Tizenkét évszázad múltán Heribert Illig a levéltártól elkéri a levelet, beleolvas, és nem hisz a szemének.



minden helyes ítem 0,5 pont

9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)
 (F - az ki van zárva, mert nem tudom)

- [D] *Q* helyettesíthető *S*-sel, mert *S* *Q* leszármazottja.
 [E] *T* helyettesíthető *R*-rel, mert *R* *T* leszármazottja.
 [D] *T* *size* attribútuma *T* példányosítása nélkül is elérhető, mert az attribútum statikus.
 [B] *T* nem hívhatja meg *R* *bar* metódusát, mert nem ismeri *R* őseit.
 [C] *W* *foo* metódusa kaphat paraméterül *T*-t, mert *T*-nek *S*-sel közös az őse.
 [B] *T* legalább kettő *R*-t ismer, mert a '*' számosságban a 'sehány' is benne van.
 [E] *W* meghívhatja *S* *qux* metódusát, mert *W* ismeri *R*-t.
 [B] *S* meghívhatja *W* *foo* metódusát, mert *S*-t több *W* is ismerheti.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból

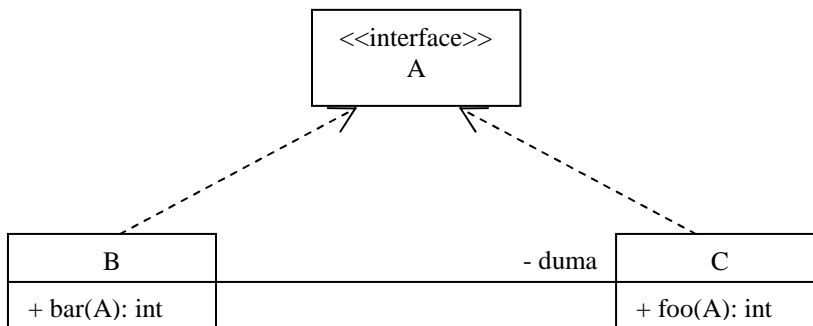
2005. december 20.

1. Melyik az a három elv, amelyet mind adatok, mind utasítások komponálásakor alkalmazunk ? (3 pont)

- | | |
|--|---|
| <input type="checkbox"/> modularizáció | <input checked="" type="checkbox"/> iteráció |
| <input checked="" type="checkbox"/> szekvencia | <input type="checkbox"/> öröklés |
| <input type="checkbox"/> kollaboráció | <input checked="" type="checkbox"/> szelekció |
| <input type="checkbox"/> dereguláció | <input type="checkbox"/> delegáció |

2. Rajzoljon az alábbi programrészletnek megfelelő UML osztály diagramot ! Csak olyan jellemzőket ábrázoljon, amely a kódból kiolvasható ! (8 pont)

```
public interface A {.....}  
public class B {  
    private C duma;  
    public int bar(A a) {  
        int i = duma.foo(a); }  
}
```



Minden elem 1 pont

3. Definiálja a szoftver hibával kapcsolatos alábbi fogalmakat ! (6 pont)

Error human action or omission that results in a fault.....

Fault is a software defect (incorrect step, process, or data definition) that causes a failure

Bug same as Fault.....

Failure the inability of a software to perform its required functions within specified performance requirements

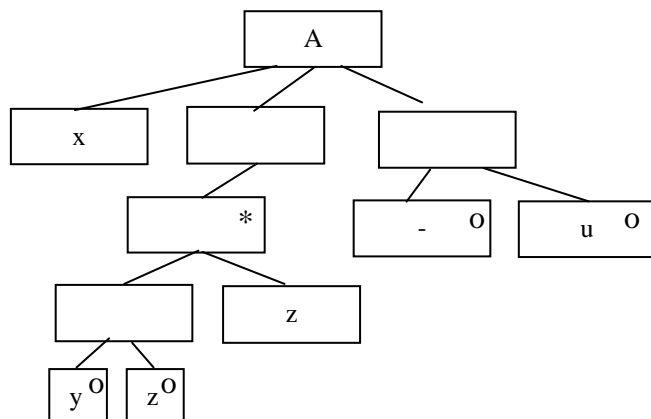
4. Az alábbi XML leírás jól formált ? Ha nem, akkor mi a baja ? (2 pont)

```
<?xml version="1.0"?>  
<targy>Programozas technologiaja</targy>  
<neptun>VIF02228</neptun>  
<vizsga>elso</vizsga>  
<torzs>nem is nehez</torzs>
```

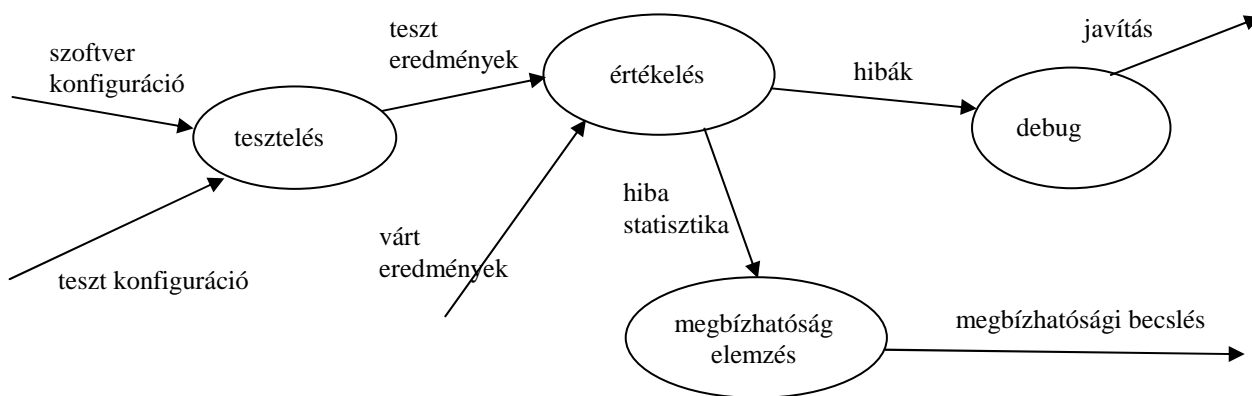
- | | |
|-------------------------------------|------|
| <input type="checkbox"/> | IGEN |
| <input checked="" type="checkbox"/> | NEM |

nincs gyökere.....

5. Legyen egy A entitás, amelyen az x, y, z és u események fordulhatnak elő. Rajzolja fel az A entitás élettörténetét a JSD szerint, ha az események sorrendjét az alábbi BNF leírással definiáltuk ! (6 pont)

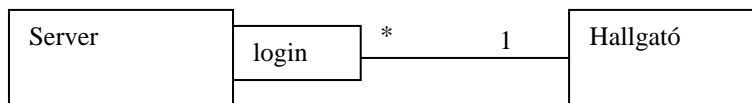
$$x\{[y|z]z\}[u]$$


6. Rajzolja fel a tesztelés információs folyamatának adatfolyamábráját ! (6 pont)



mindegyik gombóc és él 0,5 pont.

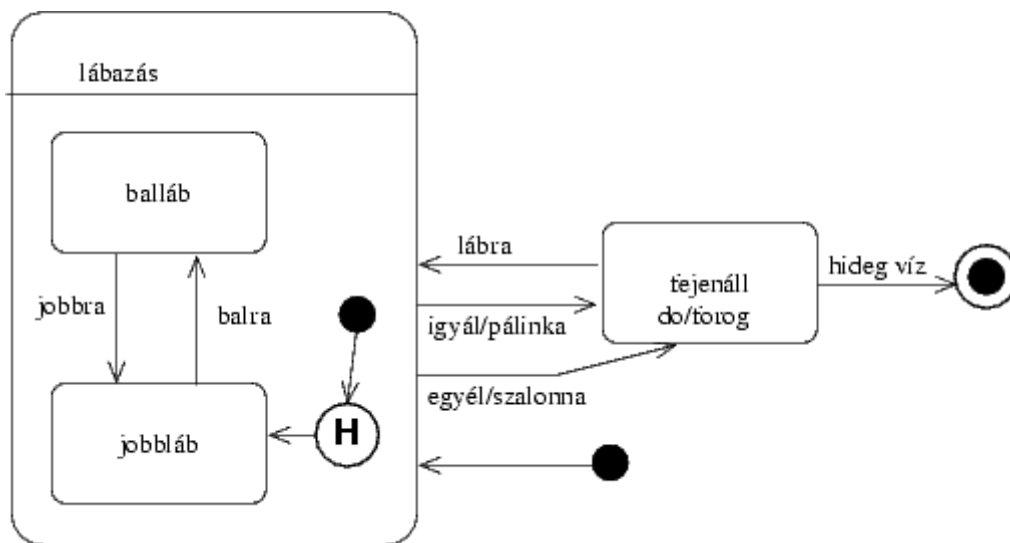
7. Egy hallgató több szervert használ, és egy szerver több hallgatót szolgál ki. Rajzolja fel a szervert, a hallgatót és a közöttük fennálló kapcsolatot UML osztálydiagramon, minősítőként (qualifier) használva a login nevet! Jelölje a multiplicitást ! (3 pont)



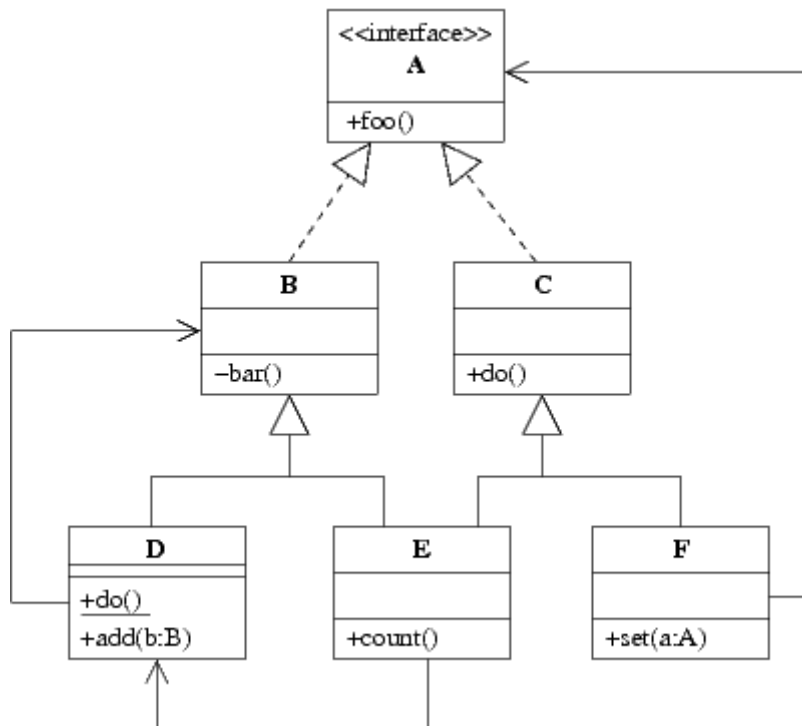
Qualifier és a multiplicitások 1-1 pont

8. Rajzoljon az alábbi történetben szereplő tánchoz UML state-chartot! (8 pont)

Aszályfürdőben évente megrendezik a forrástisztító ünnepélyt. Ekkor a falu legöregebb lakója a forrácsurgató táncot járja. A tánc a lábazzással kezdődik: a táncos a jobb lábán áll, majd a javasasszonyok 'balra' kiáltására balra vált, a 'jobbra' kiáltásra pedig jobbra. Ez így megy egészen addig, amíg vagy a javasurak 'igyál' kiáltására megiszik egy kupica hagymapálinkát, vagy a javasgyerekek 'egyél' kiáltására megesszik egy falat szalonnabórt, majd mindkét esetben fejre áll, és így forogni kezd. Ekkor a javasasszonyok 'lábra' kiáltására ismét féllábra áll, pont arra, amelyiken a fejreállítás előtt állt. A tánc akkor véget, amikor a táncos fejen áll, és a felesége hideg vízzel leönti.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)
F - az ki van zárva, mert nem tudom

- [D] **B** bárhol helyettesíthető **E**-vel, mert az **E** a **B** leszármazottja.
- [E] **C** bárhol helyettesíthető **D**-vel, mert a **D** a **C** leszármazottja.
- [D] **F** *set(a:A)* függvénye kaphat paraméterül **D**-t, mert a **D** megvalósítja az **A** interfészt.
- [B] **D** *add(b:B)* függvénye meghívhatja egy paraméterül kapott **E** *count()* metódusát, mert **E** megvalósítja az **A** interfészt.
- [C] **E** meghívhatja egy **D** *add(b:B)* metódusát, mert közös az ősük.
- [A] **E** nem hívhatja meg egy **D** *bar()* metódusát, mert a metódus *protected*.
- [B] **F** meghívhatja egy **D** *add(b:B)* metódusát, mert **E** egyszerre a **B** és a **C** osztály leszármazottja.
- [E] **D** *add(b:B)* metódusából nem hívhatjuk meg **D** *do()* metódusát, mert a metódus absztrakt.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

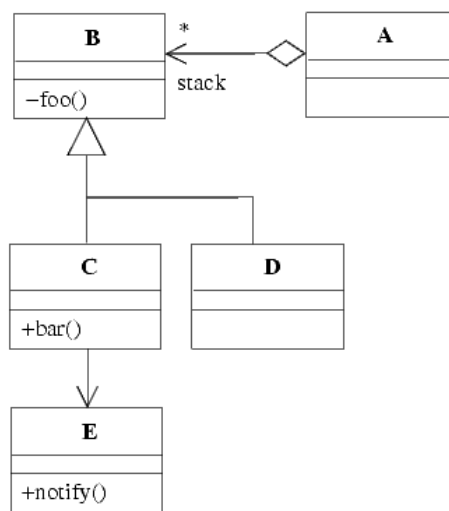
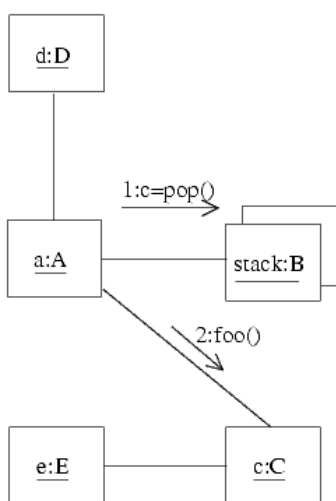
c. tárgyból
2006. január 10.

1. Mi a CMM negyedik szintjének neve (1 pont): Managed (menedzsel, irányított).....
Jellemezze röviden (3 pont):

Both the software process and products are quantitatively understood and controlled.

2. Jelölje be, hogy a mellékelt ábrán milyen UML elemek láthatóak (8 pont)

- aggregáció
- függőség
- osztály metódus
- példányosítás
- multiobject
- objektum
- kollaborációs diagram
- demarkáció
- iteráció
- absztrakt metódus
- proxy



3. Milyen minőségi jellemzőket tesztlünk a különböző FURPS típusú tesztekkel ? (5 pont)

Functionality, funkcionalitás Usability, használhatóság.....

Reliability, megbízhatóság..... Performance, teljesítőképesség

Supportability, támogatottság.....

minden helyes válasz 1 pont, van benne 1 plusz pont is

4. Az alábbi XML leírás jól formált ? Ha nem, akkor mi a baja ? (2 pont)

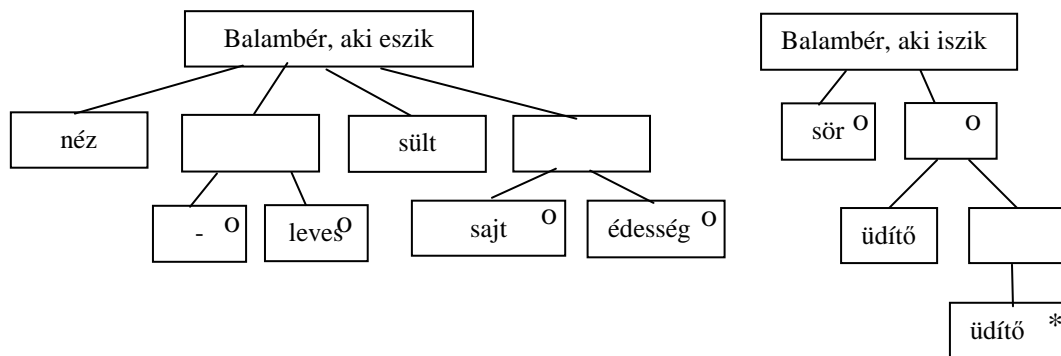
```

<?xml version="1.0"?>
<uzenet>
<cim varos="Budapest">"Budapest"
<utca>Hintalo</utca>777</cim>
<felado>LZ</felado>
</uzenet>
    
```

- IGEN
- NEM

.....

5. Balambér (aki egy átlagos vendég) az étteremben először átnézi az étlapot. Alkalmanként levest is kér, de mindig frissensültet fogyaszt. Az ebédet sajttal vagy édességgel zárja. Ebéd közben megiszik egy sört vagy néhány (egy vagy több) üdítőt. Rajzolja fel az entitás(ok) élettörténetét a JSD szerint ! (7 pont)



6. Egy adatfolyam-ábrán az S1 és S2 adattárak, valamint a P1, P2 és P3 processzek szerepelnek. A vezérlőegység bemenetét a következő specifikáció írja le: [ev1|xxy|ko7]. Töltse ki a vezérlőegység állapottábláját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) az adatfolyam és adatspecifikációval ! (5 pont)

aaa				
bbb				

az oszlopok paraméterezése (sorrend lényegtelen) csak a bemenet konstansai lehetnek 1 pont

a következő állapotok csak a bejelölt két állapot (aaa, bbb) vagy don't care lehet, kell benne lenni állapotváltásnak is. 2 pont

az akció ha nem üres, csak processz (P1, P2, P3) lehet és mindegyiknek legalább egyszer szerepelni kell. 2 pont

7. Elkészítjük a mellékelt X osztály két példányát, x1-et és x2-t. Ezt követően végrehajtjuk a következő műveleteket:

$x2.a = x1.a - x1.b;$

$x1.a += 6;$

Mennyi lesz alábbi attributumok értéke ? (3 pont)

$x1.a = 2...$

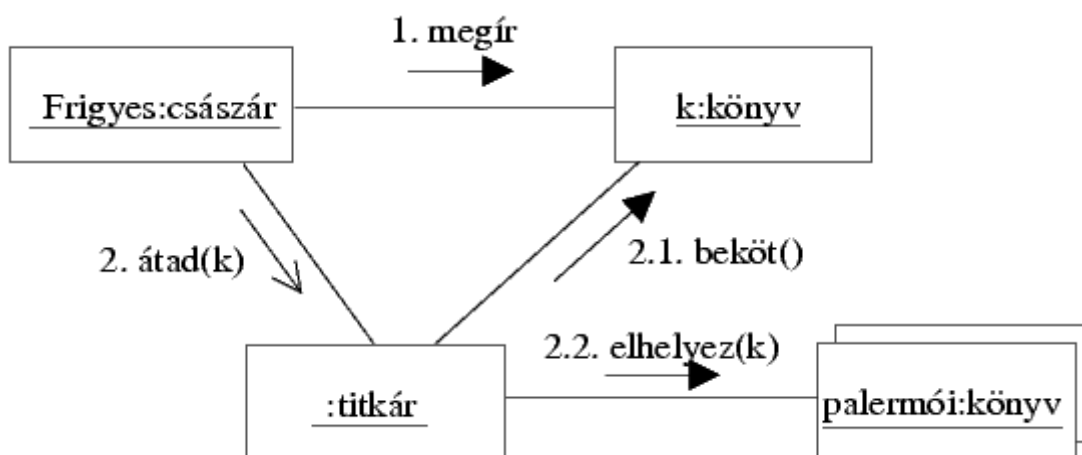
$x2.a = 2...$

mindegyik jó 1,5 pont

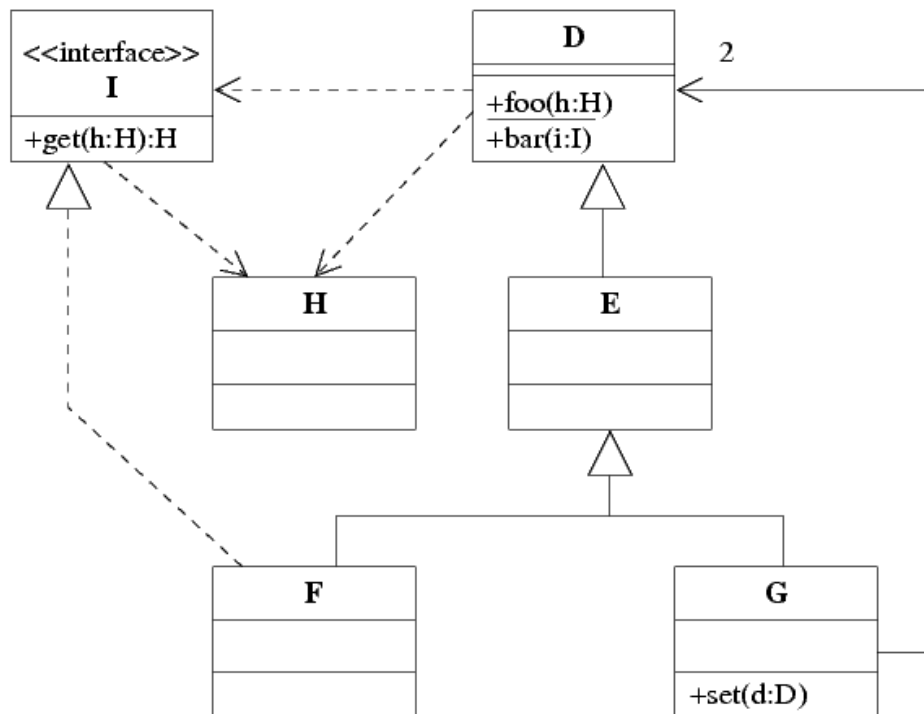
X
$\text{int } a = 0$
$\text{int } b = 4$

8. Rajzoljon UML kollaborációs diagramot az alábbi történetre! Ügyeljen az üzenetek helyes sorszámozására! (8 pont)

II. Frigyes német-római császár megírja "de tribus impostoribus" című munkáját, és átadja titkárának, majd mással kezd foglalkozni. A titkár a művet díszes kötéssel látja el, majd elhelyezi a császár palermói könyvei között.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)
F - az ki van zárva, mert nem tudom)

- [D]** F nem helyettesíthető E-vel, mert E nem leszármazottja F-nek.
[E] G helyettesíthető F-fel, mert D nem közös ősük.
[A] H nem hívhatja meg D foo metódusát, mert a metódus absztrakt.
[E] D bar metódusa nem kaphat paraméterül F-et, mert F nem leszármazottja D-nek.
[A] D foo metódusa nem kaphat paraméterül I-t, mert az I valósítja meg H-t.
[B] G konstruktora pontosan kétszer köteles meghívni D bar metódusát, mert pontosan két D-t ismer.
[A] G set metódusa meghívhatja a paraméterül kapott D objektum bar metódusát, mert a metódus statikus.
[C] F nem hívhatja meg G set metódusát, mert mindketten függnek az I interfésztől.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

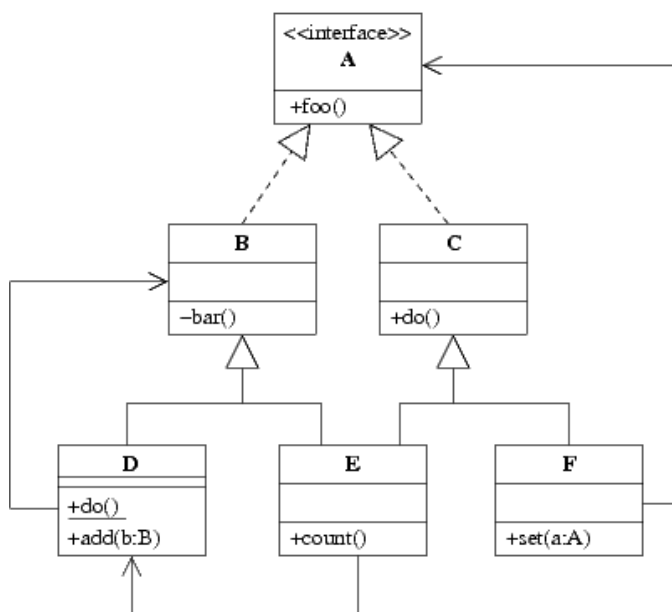
c. tárgyból
2006. január 10.

1. A kockázatok tervezése során (risk planning) milyen stratégiákat alkalmaznak ? (3 pont)

elkerülés (avoidance) minimalizálás (minimisation)
folytatás (contingency)

2. Jelölje be, hogy a mellékelt ábrán milyen UML elemek láthatóak (8 pont)

- aggregáció
- függőség
- osztály metódus
- példányosítás
- multiobject
- realizálás (implementálás)
- kollaboráció
- navigáció
- qualifier (minősítő)
- absztrakt osztály
- sztereotípus



3. Adja meg a fejlesztési folyamatban előforduló tesztelési fokozatokat (stages) (6 pont)

unit (egység, modul) test integration (integrációs) test
system (rendszer) test acceptance (átadás, elfogadás) test

.....
minden helyes válasz 1,5 pont

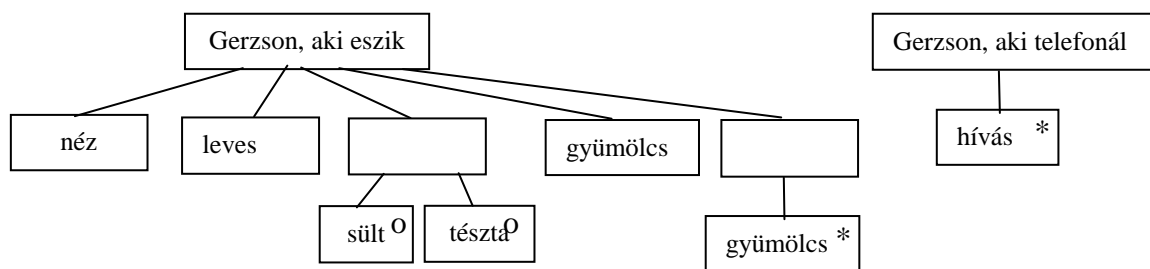
4. Az alábbi XML leírás jól formált ? Ha nem, akkor mi a baja ? (2 pont)

```
<?xml version="1.0"?>  
<uzenet>  
<cim irányitoszam=1117>Budapest  
<utca>Magyar tudosok korutja</utca></cim>  
<felado>LZ</felado>  
</uzenet>
```

- IGEN
- NEM

attributum nincs idézőjelek között.....

5. Gerzson (aki egy átlagos vendég) a menzán először szemügyre veszi a kínálatot. Levest mindig eszik, majd sültet vagy tésztát fogyaszt. Az ebédet egy vagy több gyümölccsel zárja. Ebéd közben is fogadja a mobilon érkező hívásokat. Rajzolja fel az entitás(ok) élettörténetét a JSD szerint ! (7 pont)



6. Egy adatfolyam-ábrán szereplő vezérlőegységet az alábbi állapotgéppel írunk le:

	X1	X2	X3
S1	S1/alfa	S2/	S1/beta
S2	S2/	S2/xxx	S1/alfa

Feltéve, hogy a DFD-n csak az adott állapotmodellel kapcsolatos elemek állnak

Specifikálja a vezérlő egység bemenetét ! (2 pont): [X1|X2|X3].....

Sorolja fel a processzeket ! (2 pont): **alfa, beta, xxx**

Adja meg az adattárákat ! (1 pont): **nem tudjuk**

7. Elkészítjük az alábbi Y osztály két példányát, x1-et és x2-t. Ezt követően végrehajtjuk a következő két műveletet:

```
x2.a = x1.a * 2;
x1.a +=5;
```

Y
<u>int a = 3</u>
aX()

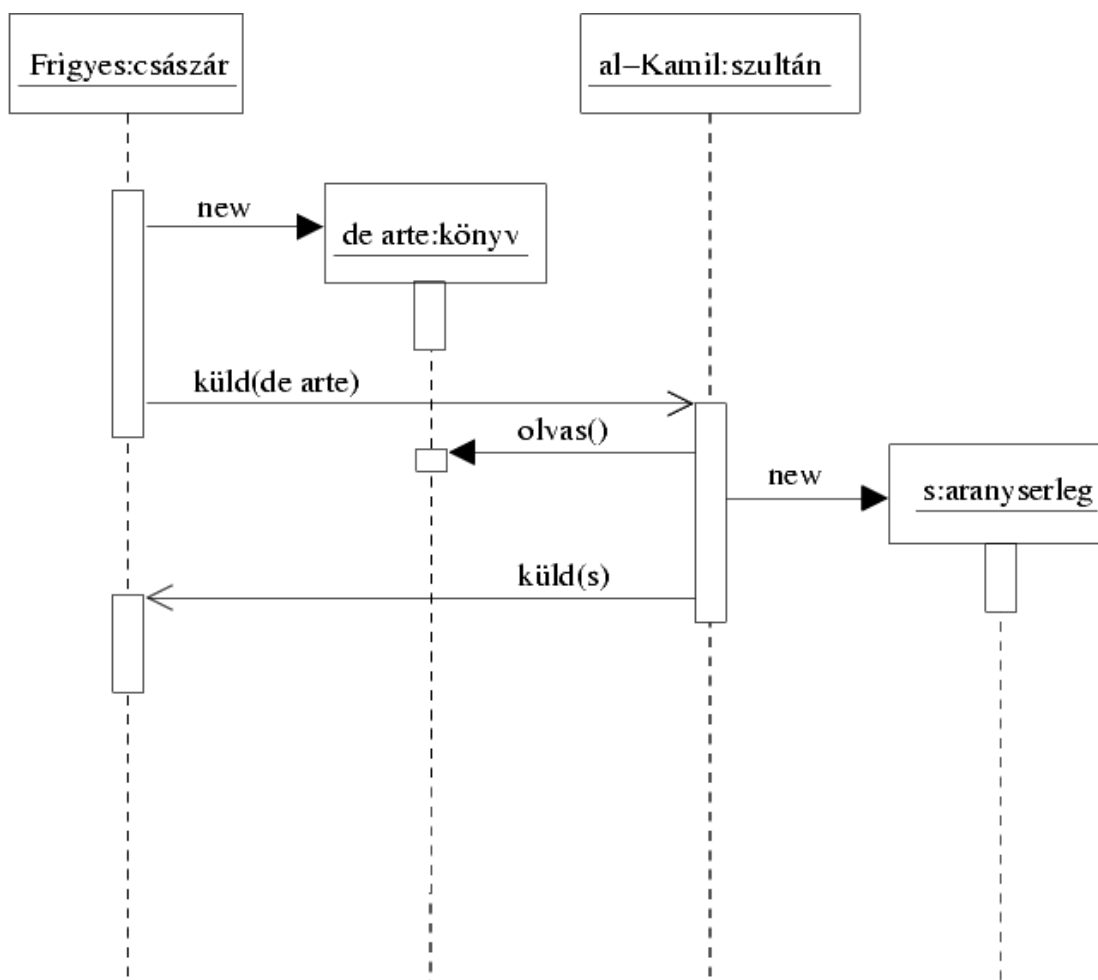
Mennyi lesz x1.a és x2.a ? (3 pont)

x1.a = **11** x2.a = **11**.....

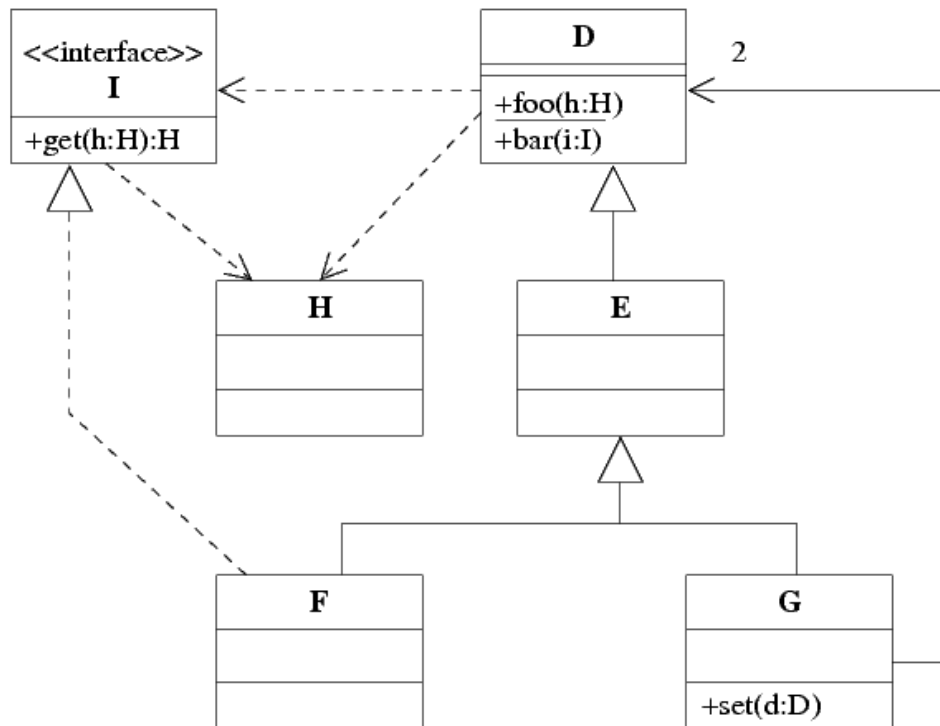
mindegyik jó 1,5 pont

8. Rajzoljon UML szekvenciadiagramot az alábbi történetre! (8 pont)

II. Frigyes német-római császár könyvet ír "de arte venandi cum avibus" címmel. A könyvet ajándékol küldi az egyiptomi szultánnak, al-Kamilnak. Al-Kamil elolvassa, majd készítet egy aranyserleget, és köszönetképp a tudós munkáért elküldi a császárnak.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)
 (F - az ki van zárva, mert nem tudom)

- [B] F helyettesíthető E-vel, mert F az E leszármazottja.
 [B] G helyettesíthető F-fel, mert D közös ősük.
 [E] H meghívhatja D foo metódusát, mert a metódus absztrakt.
 [C] D bar metódusa kaphat paraméterül F-et, mert F a D leszármazottja.
 [E] D foo metódusa kaphat paraméterül I-t, mert az I megvalósítja H-t.
 [B] G konstruktora pontosan kétszer köteles meghívni D bar metódusát, mert pontosan két D-t ismer.
 [E] G set metódusa nem hívhatja meg a paraméterül kapott D objektum bar metódusát, mert a metódus statikus.
 [B] F meghívhatja G set metódusát, mert mindketten függenek az I interfésztől.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

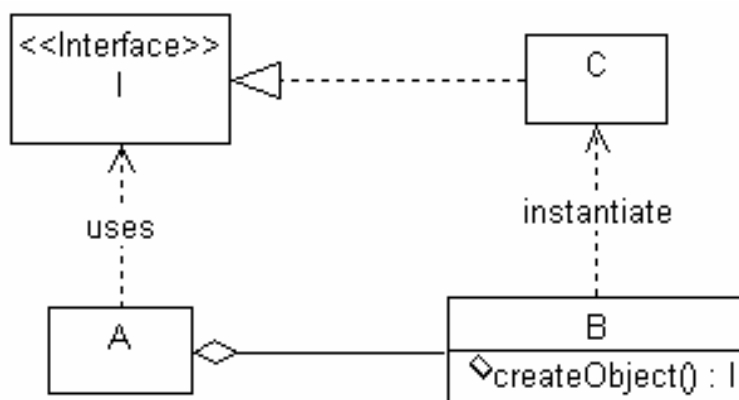
VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2006. január 24.

1. Nevezze meg és jellemezze röviden a RUP utolsó (negyedik) fázisát ! (3 pont)

Transition phase: supplying the product to the user community (manufacturing, delivering, and training)

2. Az A osztályú objektum használni akar egy I interfészt megvalósító objektumot, de A nem ismer olyan osztályt, amelyik implementálná I-t. Viszont A-nak van egy B osztályú komponense, amely ismeri az I-t megvalósító C osztályt. A meghívja B createObject() metódusát, amely konstruál egy az A elvárásainak megfelelő objektumot. Rajzoljon UML osztálydiagramot ! (8 pont)



Mindegyik kapcsolat 2 pont

3. Sorolja fel a JSD 6 lépését ! (6 pont)

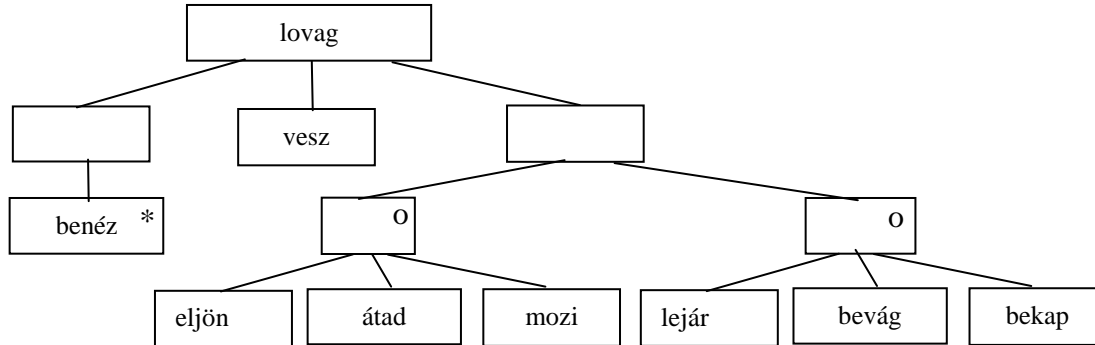
- | | |
|--|--|
| 1. Entitás akció (entity action) | 2. Entitás élettörténet (entity structure) |
| 3. Kiinduló modell (initial model) | 4. Funkció (function) |
| 5. Időzítés (timing) | 6. Megvalósítás (implementation) |
- minden helyes válasz 1 pont

4. Az alábbi XML leírás jól formált? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0"?>
<dolgozat>
<egyebkent nyul="Na mi lesz ?"> Marha-sag
<tilos>Loch Ness-i szorny</tilos></egyebkent>
<szemely>MZ/X</szemely>
<szemely>MZ</szemely>
</dolgozat>
```

- IGEN
 NEM

5. Balambér (aki egy nagyszerű lovag) randira megy. Kedvesének ornitogallumot akar vinni. Mivel ez csak kevés helyen kapható, Balambér általában kénytelen több virágáruhoz is benézni, amíg olyat talál, amelyiknél kapható ez a virág. Ha kedvese megérkezik, átadja a virágot, majd együtt moziba mennek. Ha az imádott hölgy nem jön el, akkor Balambér a drága csokrot bevágja az első kukába és a kocsmában bekap egy dupla whiskyt jéggel és szódával. Rajzoljon entitás-élettörténetet a JSD szerint! (7 pont)



6. Egy C nyelven írt shutdown () függvény felszabadítja a lefoglalt memóriát és lezárja a fájlokat. Milyen a függvény kohéziója ? (2 pont)

- | | |
|---------------------------------------|--|
| <input type="checkbox"/> procedurális | <input checked="" type="checkbox"/> temporális |
| <input type="checkbox"/> esetleges | <input type="checkbox"/> logikai |

helyes válasz 2 pont, minden rossz -1 pont, de >= 0

Nevezze meg a fentiek között nem szereplő kohéziós típusokat ! (3 pont)

Kommunikációs, funkcionális, szekvenciális

7. Az Aspektus Orientált Programozásban (3 pont)

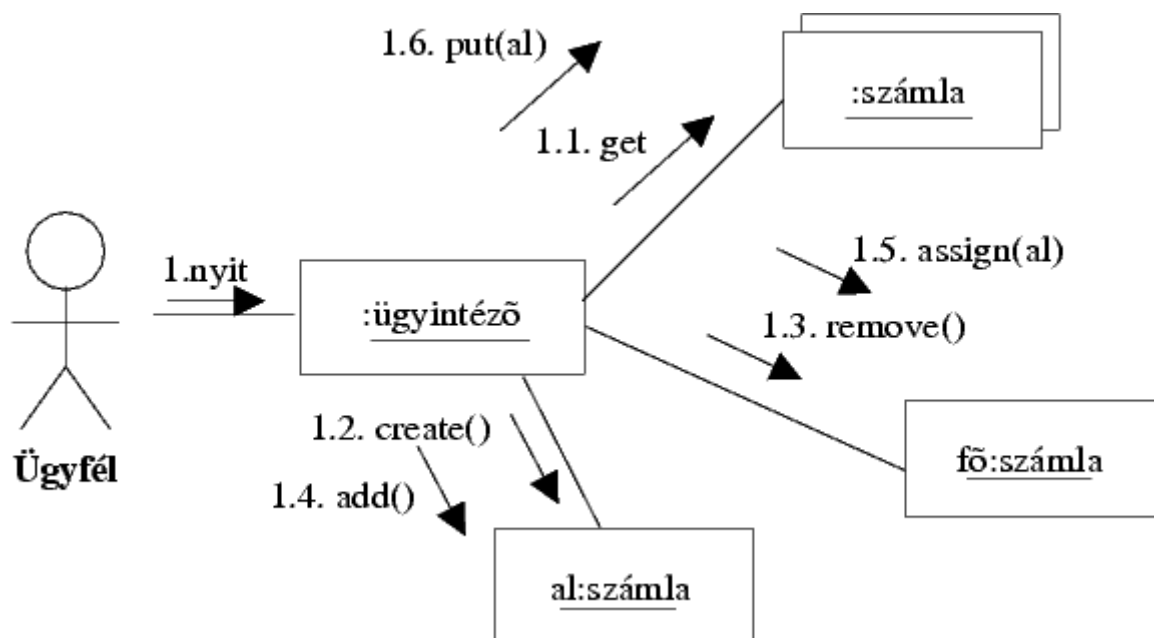
mi biztosítja a genericitást: aspektusok szövése.....

mik az interfészek: a csatlakozó pontok (join points) a szövéshez

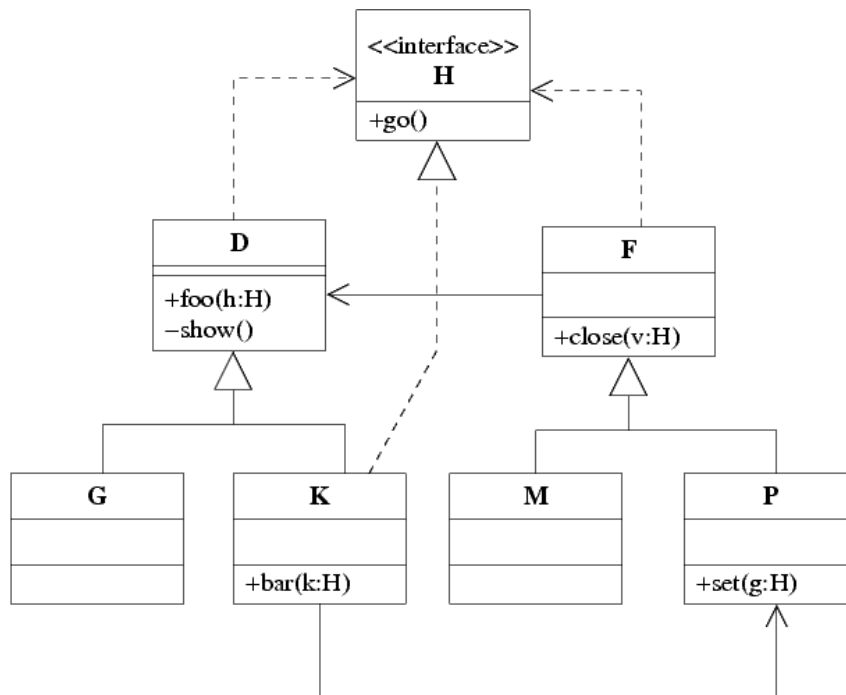
hogyan adaptálunk: ragasztó kódok és csomagolók szövése, aspektusok változtatása.....

8. Rajzoljon az alábbi use-case-hez kollaborációs diagramot! (8 pont)

Az ügyfél az ügyintézőnél kezdeményezi egy meglévő számlájához alszámla nyitását adott kezdőösszeggel. Az ügyintéző kiválasztja a számlák közül a megfelelőt (főszámla), létrehoz egy alszámlát, a főszámláról leemeli a kezdőösszeget, ezt ráteszi az alszámlára, majd az alszámlát hozzárendeli a főszámlához, végül az alszámlát elhelyezi a számlák között.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
- B - csak a második tagmondat igaz (- +)
- C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
- D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
- E - egyik tagmondat sem igaz (- -)

- [E] F meghívhatja D show() metódusát, mert a metódus statikus.
- [A] M meghívhatja D foo(h:H) metódusát, mert mindketten implementálják a H interfészt.
- [B] G helyettesíthető D-vel, mert G a D leszármazottja.
- [C] K bar(k:H) metódusából átadhatjuk a k paramétert P set(g:H) metódusának, mert K implementálja H-t.
- [C] F close(v:H) metódusa kaphat paraméterül K-t, mert K a D leszármazottja.
- [B] G helyettesíthető K-val, mert van közös ősök.
- [A] K meghívhatja P set(g:H) metódusát, mert van közös ősök.
- [D] K bar(k:H) metódusából meghívhatjuk a k paraméteren a go() metódust, mert K ismeri a H interfészt.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2006. május 23.

1. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

gyakorlati problémák megoldására szolgáló **dolgok**

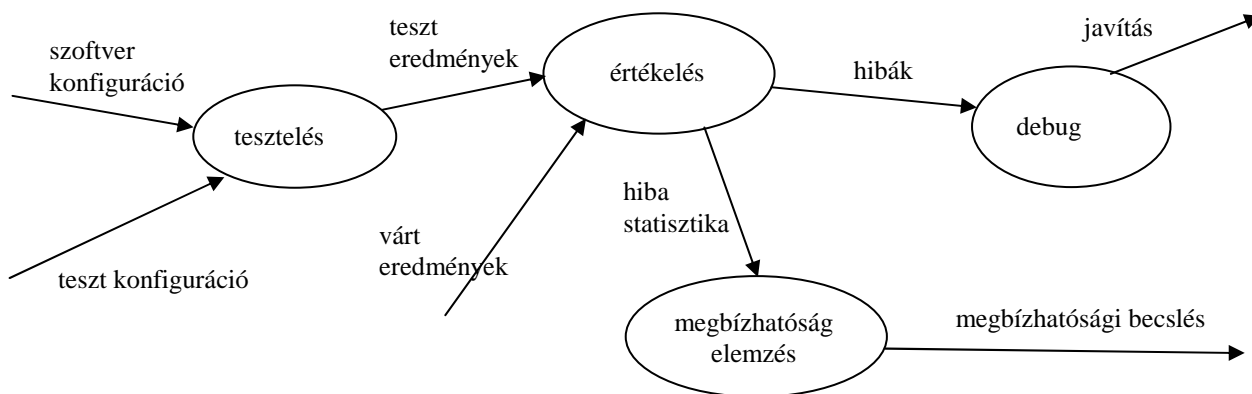
a **társadalom** által szabályozott **tudományos** alapokon álló

gazdaságos

előállításának mikéntje

mindegyik komponens 1 pont.

2. Rajzolja fel a tesztelés információs folyamatának adatfolyamábráját ! (6 pont)



mindegyik gombóc és él 0,5 pont.

3. Sorolja fel a **JSD** 6 lépését ! (6 pont)

1. Entitás akció (entity action) 2. Entitás élettörténet (entity structure)

3. Kiinduló modell (initial model) 4. Funkció (function)

5. Időzítés (timing) 6. Megvalósítás (implementation)

minden helyes válasz 1 pont

4. Az alábbi XML leírás jól formált? Ha nem, akkor mi a baja? (3 pont)

```
<?xml version="1.0"?>
```

```
<pt-vizsga>
```

```
<pelda_xml datum="tegnap"> 2006 majus 23
```

```
</pelda_xml>
```

```
<xxx/>kapufa
```

```
</pt-vizsga>
```

IGEN

NEM

.....

5. Legyen egy O objektumnak egy $int\ y(int\ x)$ metódusa, ahol csak $10 < x < 50$ megengedett, és akkor a $50 < y < 100$ igaz. Az O leszármazottjában felüldefiniáljuk y -t. Az alábbi elő- és utófeltételek közül jelölje meg azokat, amelyek ha igazak a felüldefiniált y -on, akkor az kielégíti a Liskov-elvet ! (6 pont)

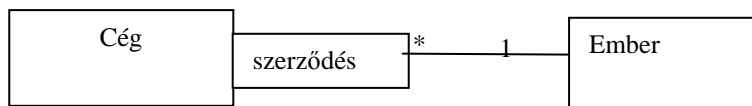
- | | |
|---|---|
| <input type="checkbox"/> $20 < x < 40$ | <input checked="" type="checkbox"/> $60 < y < 90$ |
| <input checked="" type="checkbox"/> $00 < x < 50$ | <input type="checkbox"/> $00 < y < 120$ |
| <input checked="" type="checkbox"/> $00 < x < 1000$ | <input type="checkbox"/> $50 < y < 1000$ |
| <input type="checkbox"/> $30 < x < 60$ | <input type="checkbox"/> $40 < y < 110$ |

minden helyes válasz 2 pont, minden rossz -1 pont

6. Nevezze meg sorrendben a CMM (Capability Maturity Model) szintjeit ! (5 pont)

1. kezdetleges (initial).....
2. ismétlődő (repeatable)
3. definiált (defined)
4. irányított (managed)
5. optimalizált (optimizing)

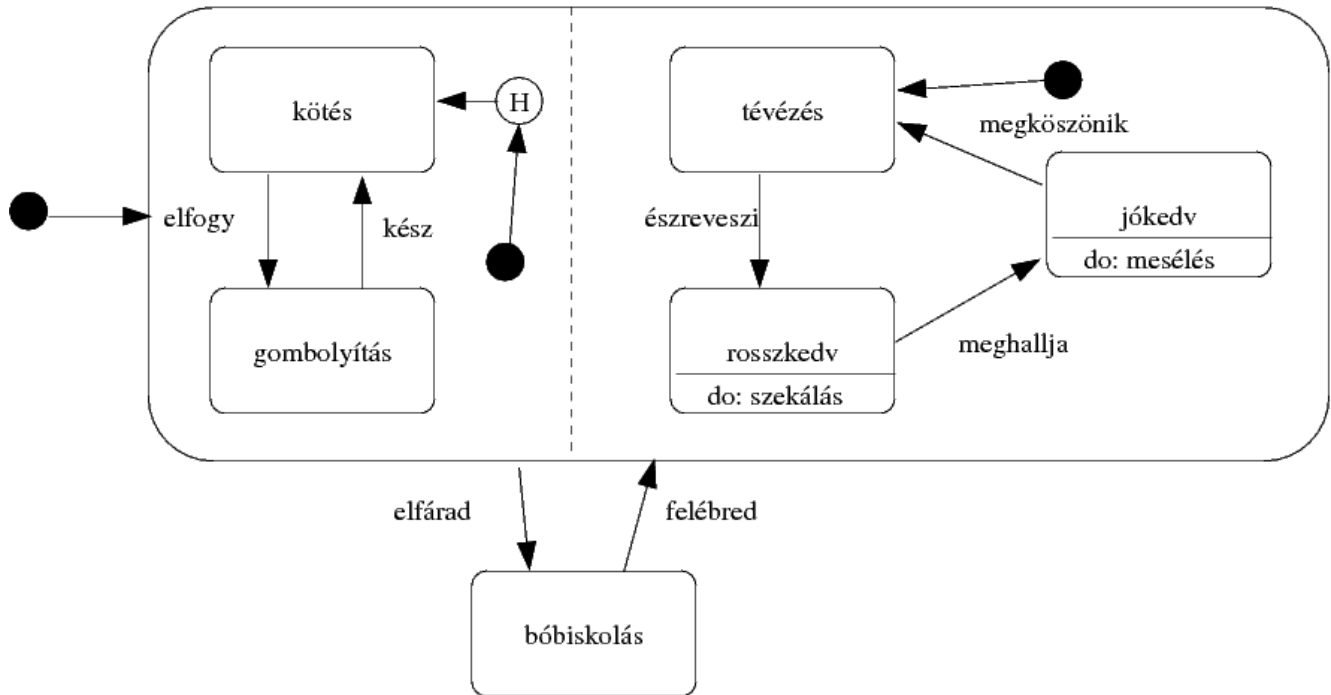
7. Egy cégnél sok ember dolgozik, egy embernek több cégnél is lehet állása. Készítsen statikus UML modellt, minősítőként (qualifier) használva a munkaszerződés számát ! Jelölje a multiplicitást ! (3 pont)



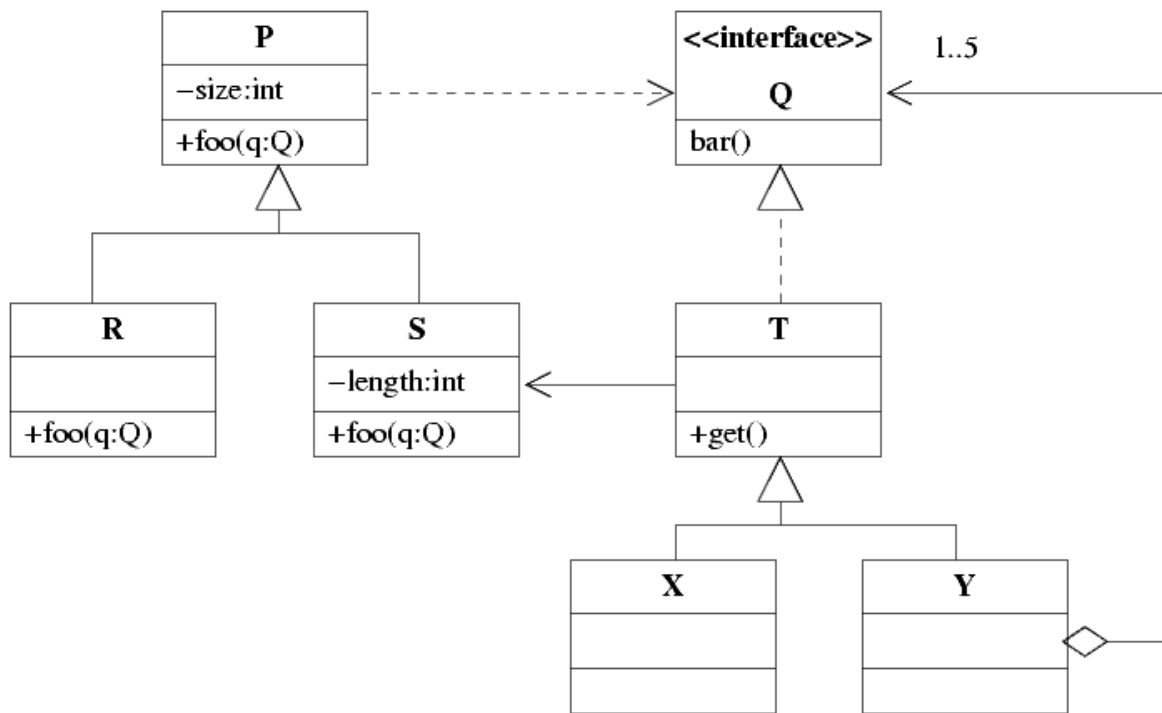
Qualifier és a multiplicitások 1-1 pont

8. Készítse el a Nagymama állapot modelljét UML-ben!(8 pont)

A nagymama alapesetben köt. Ha elfogy a fonal, gombolyít. Ha a gombolyag kész, újra köt. Mindeközben a televíziót is nézi (lehetőség szerint kedvenc sorozatát, a 'Csinos de Csintalan'-t, de ha más megy, az sem baj.) Ha észreveszi a menyét, rossz kedve lesz, és elkezd székálni (közben a kötést/gombolyítást sem hagyja abba). Ha eközben meghallja az unokái hangját, abbahagyja a székálást, jobb kedvre derül, és mesél nekik a gyermekkoráról. Ha a gyerekek megköszönik, akkor újra a műsorral foglalkozik. Néha azonban, mikor elfárad, elbóbiskol, ami bármikor megtörténhet. Mikor felébred, a kötést/gombolyítást ott folytatja, ahol abbahagyta, és a televíziót kezdi figyelni.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B]** S bárhol helyettesíthető **P**-vel, mert **S** a **P** leszármazottja.
- [B]** **T** nem hívhatja meg **S** **foo(q:Q)** metódusát, mert **T** nem ismeri **P**-t.
- [E]** **R** **foo(q:Q)** metódusa nem kaphat paraméterül **X**-et, mert **X** nem valósítja meg a **Q** interfészt.
- [B]** **X** nem hívhatja meg **S** **foo(q:Q)** metódusát, mert az **S-T** asszociáció **T** felé nem navigálható.
- [E]** **P** tartalmaz **bar()** metódust is, mert **P** megvalósítja **Q**-t.
- [E]** **S** **foo(q:Q)** metódusában nem hívhatjuk meg a **q** paraméter **bar()** metódusát, mert **S** nem ismeri a **Q** interfészt.
- [D]** **Y** nem olvashatja **S** **length** attribútumát, mert az attribútum privát.
- [B]** **X** bárhol helyettesíthető **Y**-nal, mert van közös ősük.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2006. május 30.

1. Jellemezzünk egy stringet az alábbi műveletekkel ! Adja meg a CUT műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket.(4 pont)

CRT() új (üres) stringet hoz létre.
SET(s,x) az s string elejére rakja az x karaktert.
LGTH(s) az s string karaktereinek számát adja.
CUT(s, n) az s string legrégebbi n darab karakterének levágása után maradó stringet adja
Ha n nem kisebb, mint s string hossza, üres stringet kapunk. Tételezze fel, hogy $n > 0$!

$CUT(CRT()) = CRT()$		-0.5 pont, ha hiányzik
$CUT(SET(s, x), n) ==$	if ($n > LGTH(s)$) $CRT()$	1 pont
	if ($n == LGTH(s)$) $SET(CRT(), x)$	1 pont
	if ($n < LGTH(s)$) $SET(CUT(s, n), x)$	2 pont

2. Definiálja a konfigurációs menedzsment (CM) fogalmát ! (3 pont)

*CM is the discipline of **controlling the evolution** of complex systems; software CM is its specialization for computer programs and associated documents.*

3. Definiálja a szoftver hibával kapcsolatos alábbi fogalmakat ! (4 pont)

Error human action or omission that results in a fault.....

Fault is a software defect (incorrect step, process, or data definition) that causes a failure

Bug same as Fault.....

Failure the inability of a software to perform its required functions within specified performance requirements

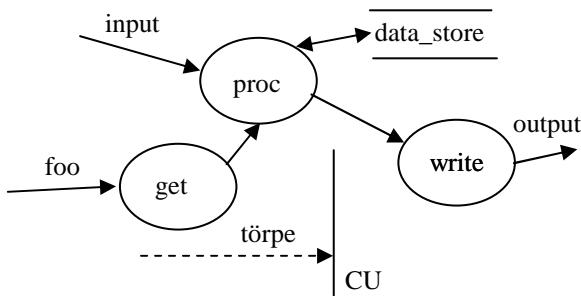
A tesztelés kapcsán az egyik legfontosabb kérdés, hogy “mikor fejezzük be a tesztelést?”. Nyilván akkor, ha a programban maradó hibák száma az előírt érték alá csökken. A maradó hibák számát azonban csak becsülni tudjuk. Milyen módszerrel ? (2 pont)

vadszámlálás

4. A szoftver projekt COCOMO szerinti költségbecslése során alkalmazott egyik módosító/hangoló tényező (Effort Adjustment Factor) a “turnaround time” (TURN). Magyarázza el a fogalom jelentését ! (3 pont)

Egy hiba javítását követően a program új változatának előállításához (fordítás, szerkesztés, konszolidálás, stb.) szükséges idő.

5. Definiálja az adatfolyam-ábrán szereplő CU vezérlőegység egy lehetséges állapotábráját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) az adatfolyam és adatspecifikációval ! (5 pont)

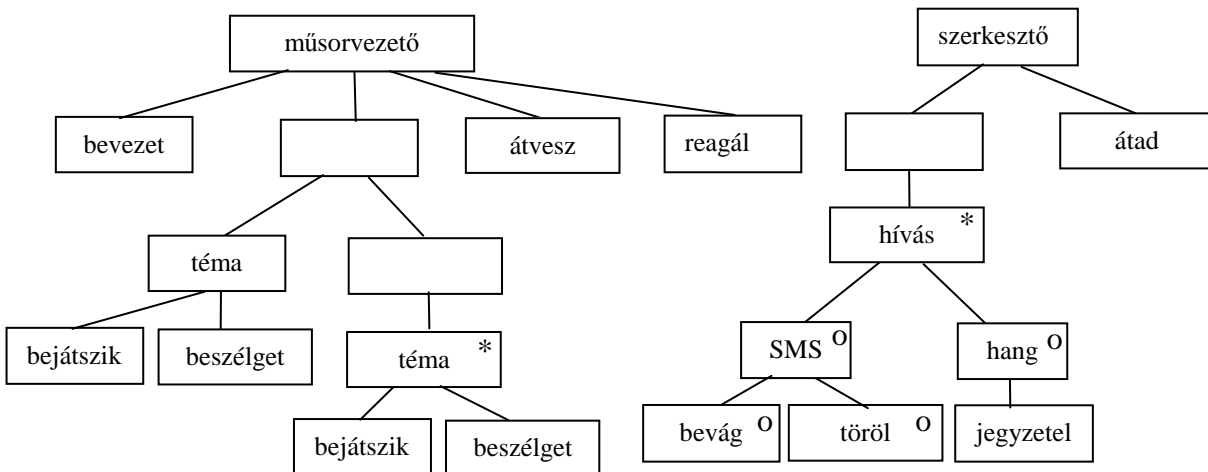


x				
y				

törpe = [tudor | vidor | kuka]

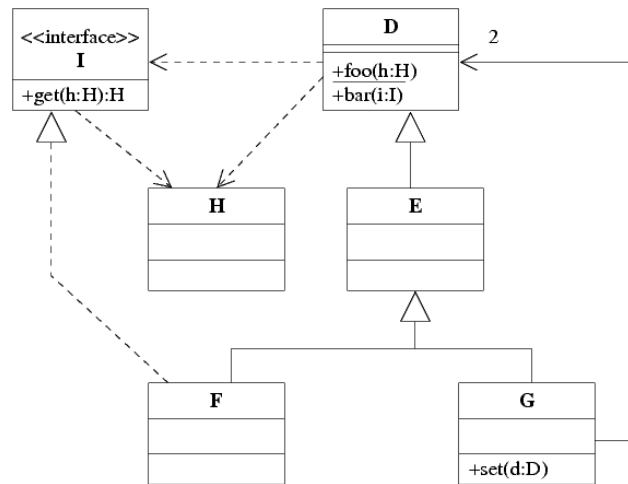
az oszlopok paraméterezése (sorrend lényegtelen) csak a törpe konstansai lehetnek 1 pont
 a következő állapotok csak a bejelölt két állapot (x, y) vagy don't care lehet, kell benne lenni állapotváltásnak is. 2 pont
 az akció ha nem üres, csak gombóc (proc, write, get) lehet és mindkettőnek legalább egyszer szerepelni kell. 2 pont

6. A televíziókban gyakran látható egyik műsortípus forgatókönyve a következő. A műsorvezető bevezetőjében röviden ismerteti az adás tartalmát. Minden adásban egy vagy több téma kerül terítékre. Minden téma egy felvezető filmbejátszással kezdődik, majd a témáról a műsorvezető a vendéggel beszélget. Az adás közben érkező hívások, amelyeknek egyik része SMS, a szerkesztőhöz futnak be. Az SMS-ek közül egyeseket bevág az adásba (az ernyő alján megjelenik), a többit pedig törli. A hívások fennmaradó részében a néző megmondja a frankót, amiről a szerkesztő jegyzetet készít. A műsor végén átadott jegyzet alapján a műsorvezető reagál a nézők véleményére. Rajzoljon entitás-élettörténetet a JSD szerint! (7 pont)



7. Jelölje be, hogy az ábrán milyen UML elemek láthatóak ! (6 pont)

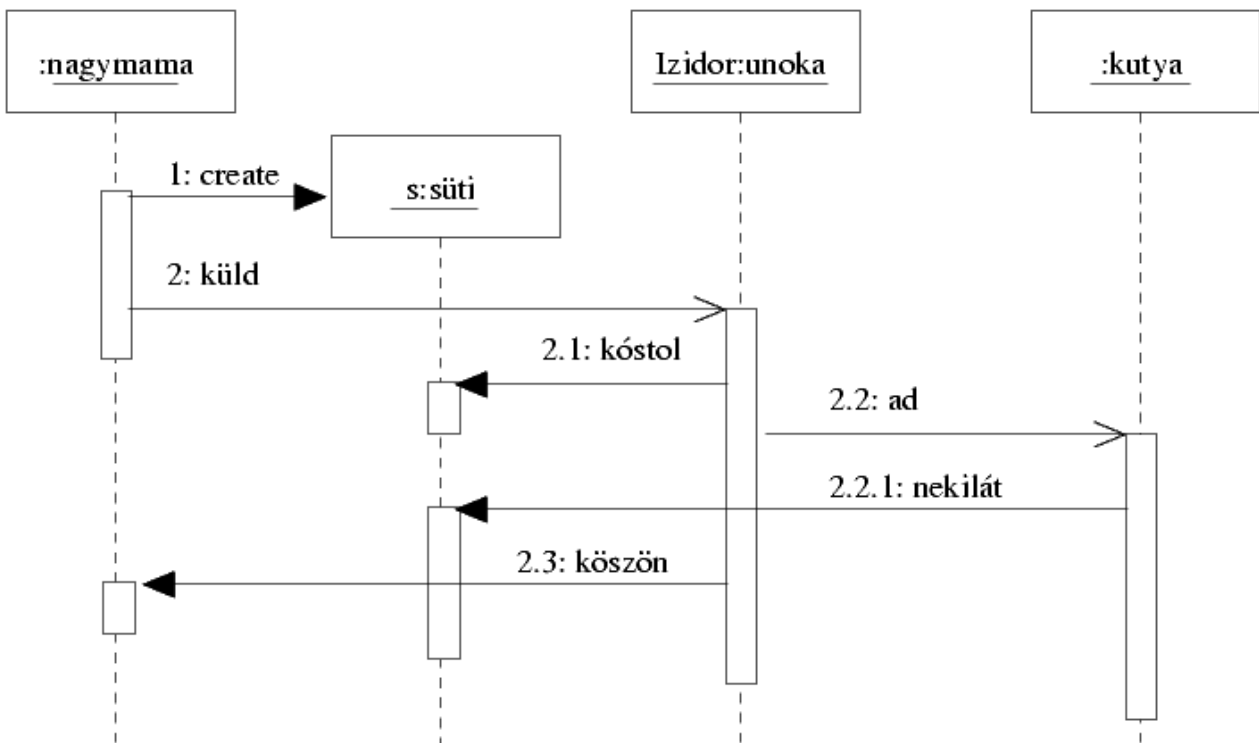
- beágyazott osztály (nested class)
- metaclass
- osztály metódus
- példányosítás
- multiobject
- realizálás (implementálás)
- paraméterezett osztály
- függőség (dependencia)
- aggregáció
- statikus attribútum



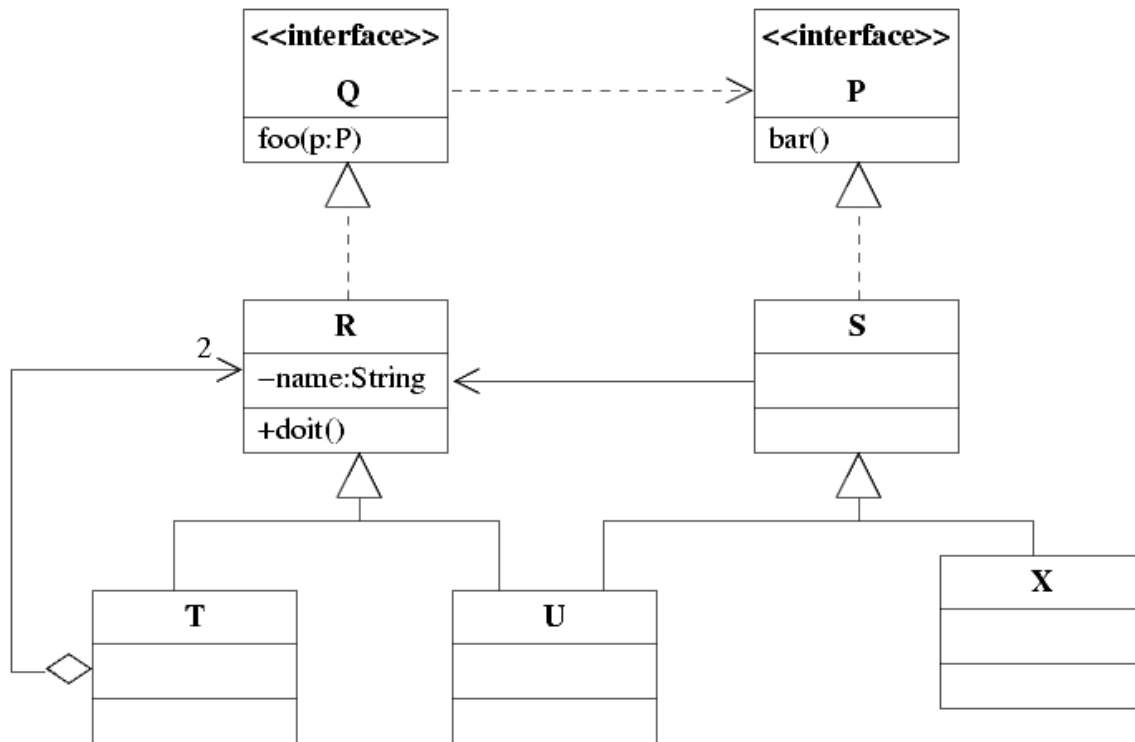
minden találat 2 pont, minden rossz válasz -1 pont, de total >= 0

8. Készítsen UML szekvencia diagramot az alábbi történet modellezésére !(8 pont)

Kovács nagymama süteményt süt, és elküldi a kedvenc unokájának, Izidornak. Izidor, mikor megkapja, azonmód megkóstolja, de nem ízlik neki, ezért odaadja a kutyájának, aki egyből nekilát a kőkemény ajándék feldolgozásának. Közben Izidor felhívja a nagymamát, hogy megköszönje a finom csemegét.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)

- [E]** P mindig helyettesíthető Q-val, mert Q a P leszármazottja.
[D] T foo(p:P) metódusa kaphat paraméterül X-et, mert X megvalósítja a P interfészt.
[B] U foo(p:P) metódusa nem kaphat paraméterül U-t, mert R-nek és S-nek nincs közös őse.
[B] R foo(p:P) metódusa nem hívhatja meg egy paraméterül kapott S bar() metódusát, mert R nem ismeri S-t.
[B] T nem tartalmazhat egyszerre egy T és egy U objektumot, mert az U S leszármazottja is.
[E] X-ből nem hozhatunk létre példányt, mert nincs asszociációban senkivel.
[D] S nem látja R name attribútumát, mert az attribútum privát.
[B] X bárhol helyettesíthető T-vel, mert U-val mindkettőnek van (külön-külön) közös őse.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

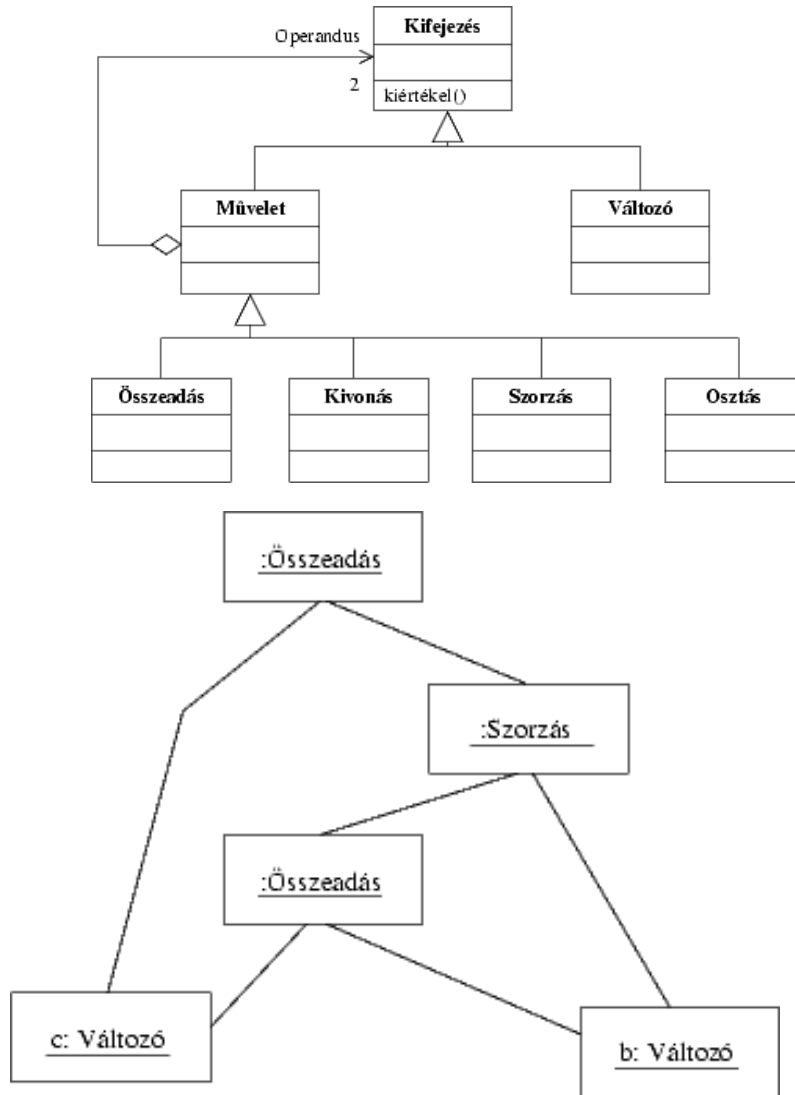
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2006. június 17.

1. Az ábrán egy - kifejezéseket leíró - osztálydiagram látható. Rajzoljon **UML objektum diagramot** a $b * (b + c) + c$ kifejezés leírására ! (6 pont)



2. Adja meg a fejlesztési folyamatban előforduló tesztelési fokozatokat (stages) (6 pont)

unit (egység, modul) test.....	integration (integrációs) test.....
system (rendszer) test	acceptance (átadás, elfogadás) test.....

.....
minden helyes válasz 1,5 pont

3. Egy adatfolyam-ábrán szereplő vezérlőegységet az alábbi állapotgéppel írunk le:

	beta	X	c3
A1	A1/alfa	S2/	A1/ubul
S2	S2/	S2/x2	A1/alfa

Feltéve, hogy a DFD-n csak az adott állapotmodellel kapcsolatos elemek állnak

Specifikálja a vezérlő egység bemenetét ! (2 pont): [beta|X|c3]

Sorolja fel a processzeket ! (2 pont): alfa, x2, ubul

Adja meg az adattárákat ! (1 pont): **nem tudjuk**

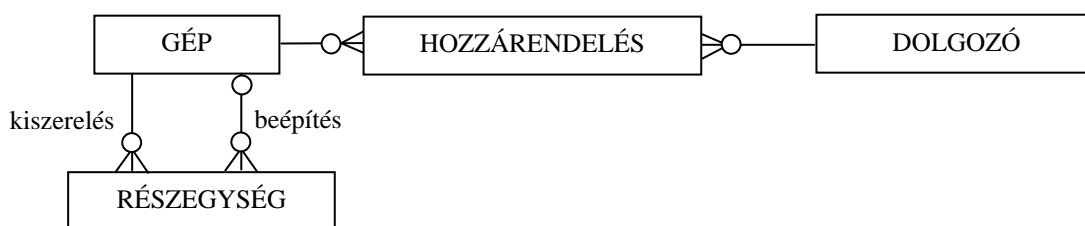
4. Az X cégnél nyilvántartják, hogy a dolgozóknál (név, adószám) milyen számítógép van. A gépet jellemzi típusa, gyári száma és ára. Aki új gépet kap, annak korábbi gépét általában más dolgozónak adják tovább, vagyis a gépek vándorolnak. A nyilvántartásból vissza kell tudni keresni, hogy egy gép mely időszakban melyik dolgozónál volt és a gépet ki installálta ott. Vannak olyan dolgozók, akiknek nincs gépük, és vannak tartalék gépek is. Bármely gépből (különböző okokból) részegységeket (tápegység, alaplap, diszk stb.) szerelhetnek ki. A részegységet jellemzi a megnevezése és száma, valamint az, hogy melyik gépben volt eredetileg. A működőképes részegységek más gépekbe beépíthetők. Egy gépből több részegységet is kiserelhetnek, egy gépben több beépített részegység is lehet. A részegységről azt is tudni kell, hogy aktuálisan melyik gépben van, mikor és ki installálta ott. Rajzoljon entitás-relációs diagramot !(8 pont)

GÉP (típus, gyári szám, ár)

DOLGOZÓ (név, adószám)

HOZZÁRENDELÉS (gép, dolgozó, időszak, installátor)

RÉSZEGYSÉG (megnevezés, szám, eredeti_gép, aktuális_gép, beépítési_idő, installátor)



Minden entitás és reláció 1 pont

5. Legyen egy az alábbi metódusokat megvalósító osztályunk ! Az azonos nevű attribútumok szemantikailag is azonosak. Mekkora lesz az osztályban a Lack of Cohesion Metric (a kohézió hiányának mértéke) ? (2 pont)

opA(a, x, c2)
 opB(b, c2, t)
 opC(c1, z, g)
 opD(a, x5, y)

$$LCOM = |P| - |Q| = 4 - 2 = 2$$

6. Jelölje meg a funkciópont elemzés (Function Point Analysis, FPA) komponenseit ! (5 pont)

- | | |
|--|--|
| <input type="checkbox"/> External Memory | <input checked="" type="checkbox"/> External Interface Files |
| <input checked="" type="checkbox"/> External Inputs | <input checked="" type="checkbox"/> Internal Logical Files |
| <input type="checkbox"/> Internal States | <input checked="" type="checkbox"/> External Outputs |
| <input checked="" type="checkbox"/> External Enquiry | <input type="checkbox"/> Internal Data Flows |
| <input type="checkbox"/> Internal Memory | <input type="checkbox"/> Internal Logical Interfaces |

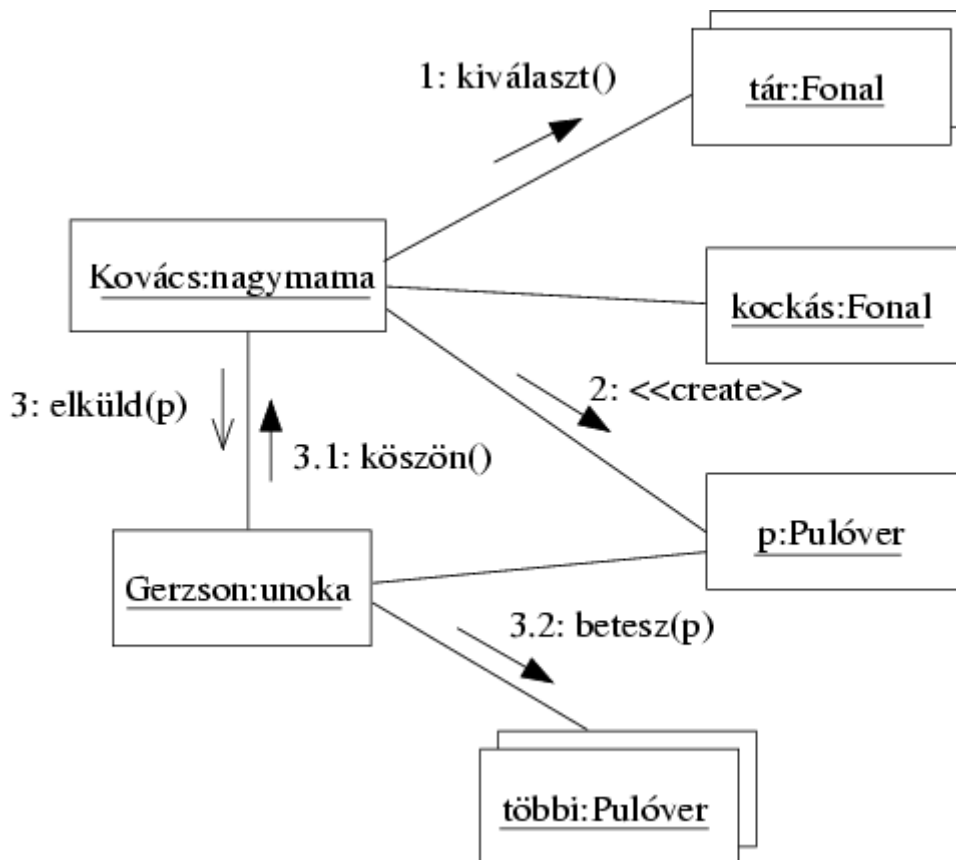
minden helyes válasz 1 pont, minden hiba -1 pont, negatív nem lehet

7. Az alábbi XML leírás jól formált ? Ha nem, akkor mi a baja ? (2 pont)

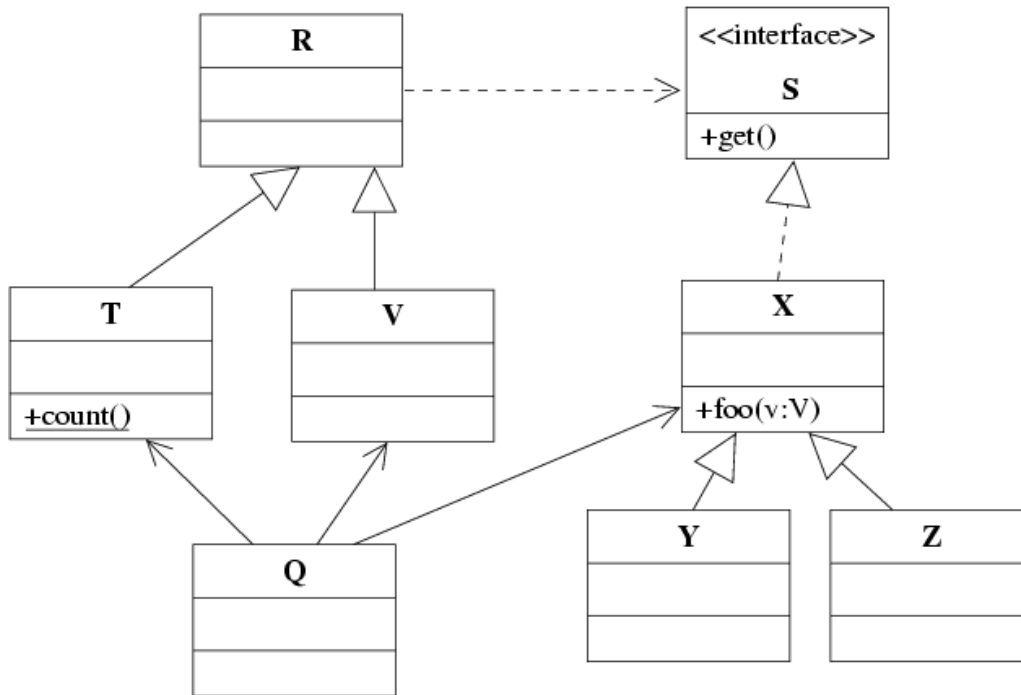
<?xml version="1.0"?>
 <targy>Programozas technologiaja</targy> IGEN
 <neptun>VIF02228</neptun> NEM
 <vizsga>harmadik</vizsga>
 <torzs>szombaton tartjuk</torzs>

nincs gyökere

8. Rajzoljon UML kollaborációs diagramot az alábbi tőtenet alapján! A sorrendiséget jelölje helyesen! (8 pont)
 Kovács nagymama pulóvert akar kötni kedvenc unokájának. Kiválaszt egy szép kockás színű fonalat a fonaltárából, megköti a pulóvert, majd elküldi az unokájának, Gerzsonnak. Az unoka megköszöni, azután a pulóvert beteszi a többi közé.



9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)

- [E]** T bárhol helyettesíthető Q-val, mert Q a T leszármazottja.
[E] S helyén bárhol állhat R, mert R megvalósítja az S interfészt.
[D] S helyén bárhol állhat Y, mert Y megvalósítja az S interfészt.
[B] Z helyén bárhol állhat Y, mert a két osztálynak van közös őse.
[B] V meghívhatja X foo() metódusát, mert a metódus publikus.
[B] Q meghívhatja X foo(v:V) metódusát T osztályú paraméterrel, mert Q ismeri mind V-t, mind T-t.
[E] T nem hívhatja meg egy S-et megvalósító objektum get() metódusát, mert nem ismeri az S interfészt.
[C] Q meghívhatja T count() metódusát, mert a metódus statikus.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2005. június 18.

1. Mi(k)hez kapcsolódik a CORBA rendszerben az "IDL váz" (angolul: "IDL Skeleton") ? (4 pont)

- | | |
|--|---|
| <input type="checkbox"/> implementációs háttértár (repository) | <input checked="" type="checkbox"/> szervant |
| <input type="checkbox"/> ORB mag (core) | <input type="checkbox"/> entitás bean |
| <input type="checkbox"/> objektum attribútum | <input checked="" type="checkbox"/> objektum adapter |
| <input type="checkbox"/> kliens | <input type="checkbox"/> interfész háttértár (repository) |

mindegyik jó 2 pont, mindegyik hiba -1, de total ≥ 0

2. Adja meg a fejlesztési folyamatban előforduló tesztelési fokozatokat (stages) (6 pont)

unit (egység, modul) test..... integration (integrációs) test.....

system (rendszer) test acceptance (átadás, elfogadás) test.....

.....
mindegyik jó 1,5 pont, mindegyik hiba -1, de total ≥ 0

3. Nevezze meg sorrendben a CMM (Capability Maturity Model) szintjeit ! (5 pont)

1. kezdetleges (initial).....
2. ismétlődő (repeatable)
3. definiált (defined)
4. irányított (managed)
5. optimalizált (optimizing)

4. Mik a csatolás dimenziói ? (3 pont)

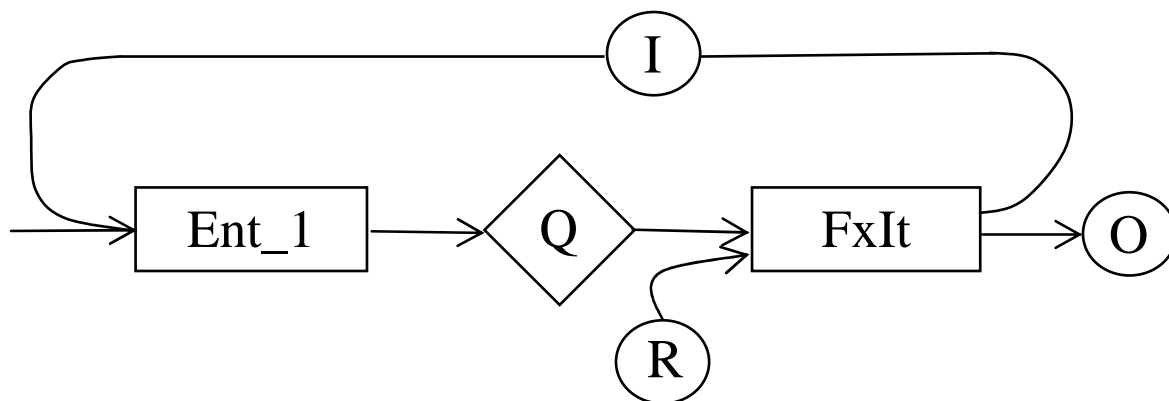
Kind of communication Size of connection.....

Time of connection.....

Egy A objektum miközben a B objektum egy metódusát hívja, paraméterül egy C objektumot ad át. Milyen a hívó és hívott objektumok (A és B) közötti csatolásban (coupling) a "kommunikáció fajtája" (kind of communication) ? (2 pont)

- | | |
|--|---|
| <input type="checkbox"/> Content (tartalom) | <input type="checkbox"/> Common (közös) |
| <input checked="" type="checkbox"/> Stamp (bélyeg) | <input type="checkbox"/> Control (vezérlés) |
| <input type="checkbox"/> Data (adat) | <input type="checkbox"/> Logic (logika) |

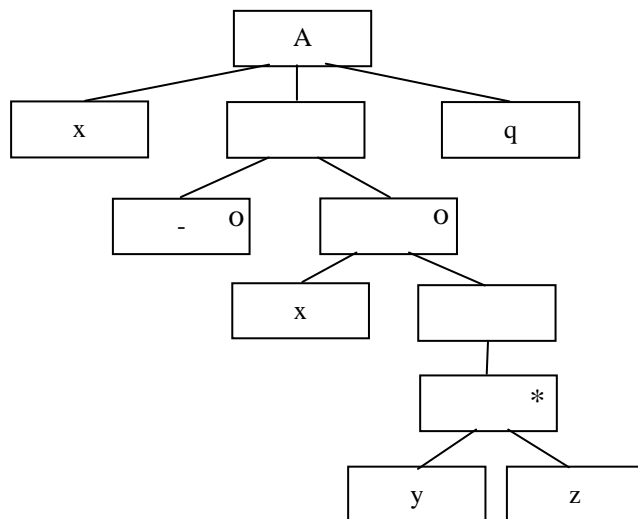
5. A JSD-ben az Ent_1 modell processzhez kapcsoljon a modell állapotán alapuló, felhasználói kérésre induló FxIt iteratív funkciót ! Rajzoljon SSD-t ! (6 pont)



mindegyik korrektul kapcsolódó item 1 pont, rossz kapcsolattal 0,5 pont

6. Legyen egy A entitás, amelyen az x, y, z és q események fordulhatnak elő. Rajzolja fel az A entitás élettörténetét a JSD szerint, ha az események sorrendjét az alábbi BNF leírással definiáltuk ! (5 pont)

$x[x\{yz\}]q$



7. Elkészítjük az alábbi X osztály két példányát, x1-et és x2-t. Ezt követően végrehajtjuk a következő műveleteket:

```
x1.a = 3; x2.a = -2;
x1.a = x1.a + 8;
x2.b = x2.a + x1.a;
```

X
<u>int a = 0</u> - int b
private aX(): int

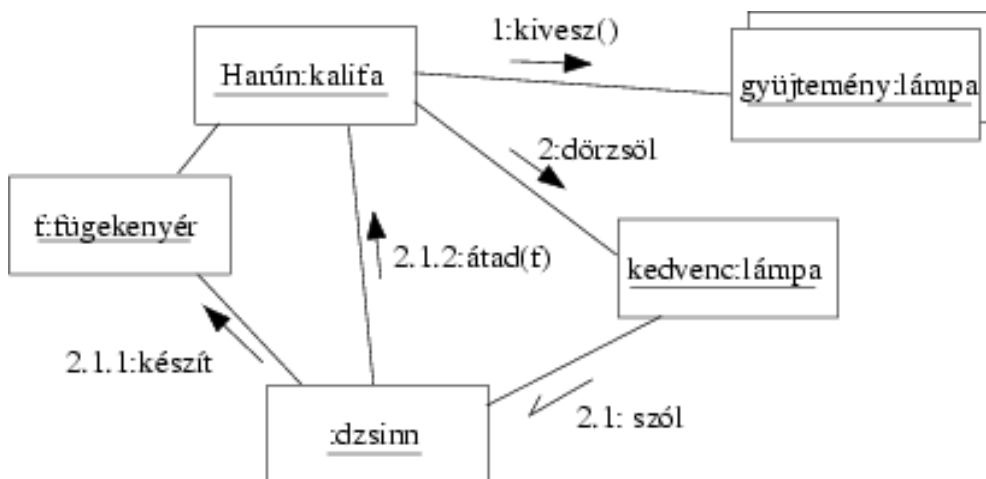
Mennyi lesz a változók értéke (3 pont)

x1.a = **6** x2.b = **12**.....

mindegyik jó 1,5 pont

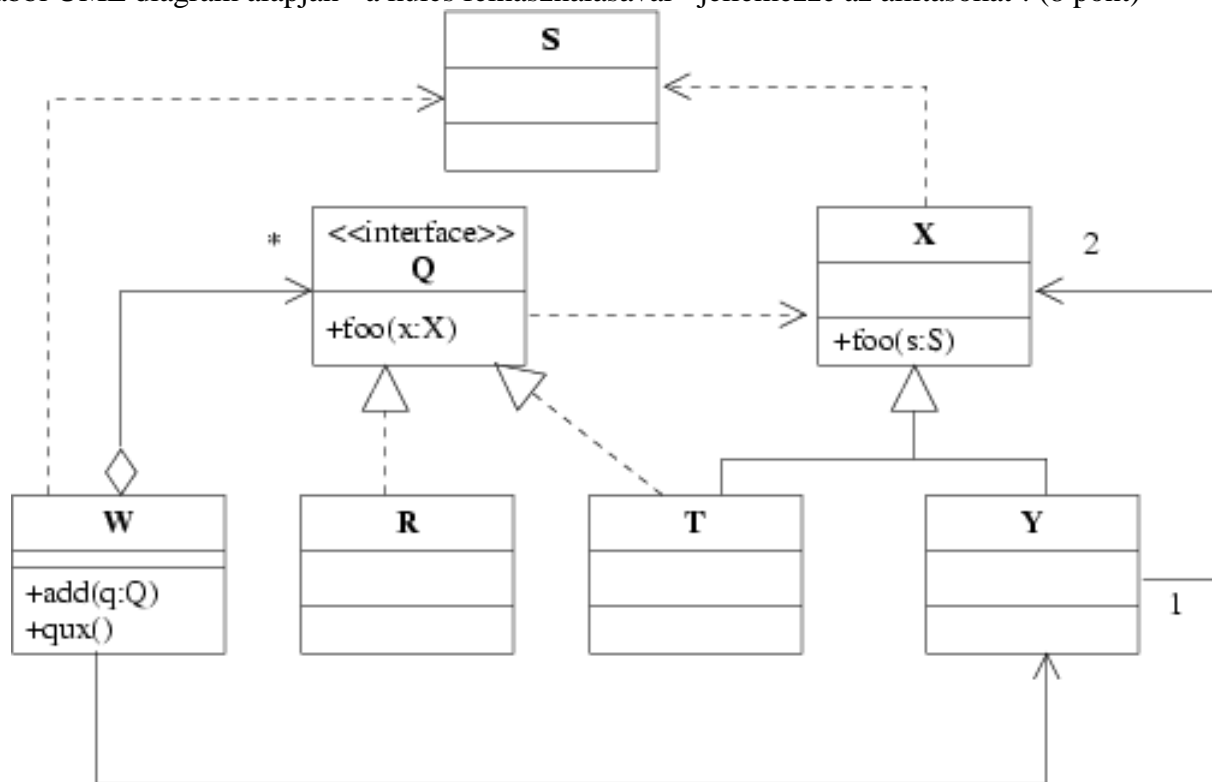
8. Rajzoljon az alábbi történethez UML kollaborációs diagramot, szabványosan jelölve rajta az interakciókat! (8 pont)

A bagdadi kalifa, Harún ar-Rasíd kiveszi lámpagyűjteményéből a kedvencét, és megdörzsöli. A dörzsölés hatására a lámpa szólítja a benne rejlő dzsinn, aki készít egy fügekenyert, és átadja az uralkodónak.



multiobject 2 pont, többi object összesen 1 pont
minden korrekt üzenet 1 pont

9. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)
 F - az ki van zárva, mert nem tudom

- [B] T helyettesíthető R-rel, mert közös interfészt implementálnak.
 [E] T helyettesíthető X-szel, mert X a T leszármazottja.
 [D] W add metódusa kaphat paraméterül R-t, mert R megvalósítja a Q interfészt.
 [E] W meghívhatja egy T objektum foo(s:S) metódusát, mert a metódus szerepel a Q interfészben.
 [B] W nem hívhatja meg Y foo(s:S) metódusát, mert nem ismeri X osztályt.
 [B] Y pontosan egy X-et ismer, mert a kapcsolatuk asszociáció.
 [E] T foo(x:X) es foo(s:S) metódusa azonos, mert egyforma a szignatúrájuk.
 [B] Y meghívhatja W qux() metódusát, mert mindketten függnek S-től.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

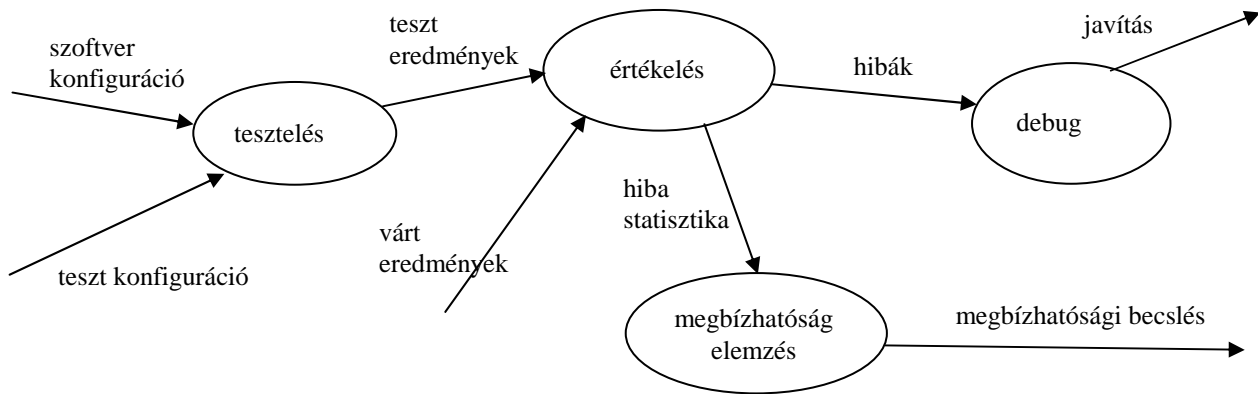
Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK PROGRAMOZÁS TECHNOLÓGIÁJA

c. tárgyból
2005. május 24.

1. Rajzolja fel a tesztelés információs folyamatának adatfolyamábráját ! (6 pont)

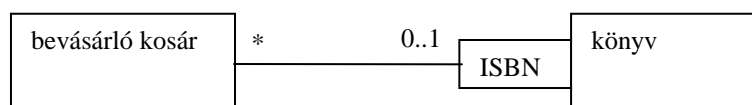


mindegyik gombóc és él 0,5 pont.

2. Hogyan értelmezzük az UML-ben a szekvenciális konkurenciát ? (A konkurencia szemantikája szekvenciális) (2 pont)

callers must coordinate outside the object so that only one flow is in the object at a time.

3. Egy webes bevásárló kosárba legalább egy könyvet teszünk, és ugyanazon könyvet sokan vásárolják meg. Rajzolja fel a bevásárló kosarat, a könyvet és a közöttük fennálló kapcsolatot UML osztálydiagramon, minősítőként (qualifier) használva a könyv ISBN számát ! (3 pont)



4. A kockázat tervezése során milyen stratégiákat választunk ? (6 pont)

A kockázat valószínűségének csökkentés

A bekövetkező káros hatások minimalizálása

A nem kívánatos eseményt követően felmerülő tevékenységek tervezése

5. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett, maximum 10 elemet tartalmazó halmazt (minden elem különböző) ! Az axiómák felírásakor megengedett összehasonlító művelet használata. (6 pont)

CRT()	új (üres) halmazt hoz létre.
INS(s, i)	az s halmazhoz adja az i elemet, ha a halmaz elemeinek száma nem nő 10 fölé. Ellenkező esetben a művelet hatástalan.
MBR(s, i)	igaz, ha az i elem az s halmaz eleme.
RMV(s, i)	az s halmazból az i elem eltávolításával kapott halmazt adja meg. Ha i nem volt a halmazban, akkor művelet hatástalan.
SIZE(s)	megadja az s halmaz elemeinek számát

$SIZE(CRT()) = 0$

$RMV(CRT(), i) = CRT()$

$MBR(CRT(), i) = false$

$SIZE(INS(s, i)) = if (SIZE(s) < 10 \ \&\& \ !MBR(s, i)) \ SIZE(s)+1 \ else \ SIZE(s)$

$RMV(INS(s, i), j) = if (SIZE(s) < 10 \ \&\& \ i \neq j) \ INS(RMV(s, j), i) \ else \ RMV(s, j)$

$MBR(INS(s, i), j) = if (SIZE(s) < 10 \ \&\& \ i == j) \ || \ MBR(s, j)$

INS-es axiómák 2-2 pont, a CRT-sek hiánya, hibája -0,5 pont

6. Készítsen **JSD** modellt az alábbi problémára ! (11 pont)

A ABC Kft. végzi a homálykőböző javítását, karbantartását. A homálykőbözőt használó cégek együttműködési szerződést kötnek az ABC-vel. Ezt követően ha bármi probléma adódik, a cég képviselője telefonál az ABC gyorsszolgálatának. A szerelő azonnal kivonul a helyszínre és a hiba természetétől függően a hibás homálykőbözőt megjavítja, vagy kicseréli. Az elvégzett munkának megfelelő javítási költséget a cégnek kiszámlázzák. Ha a cég a számlán megadott fizetési határidőig nem fizet, akkor az ABC fizetési meghagyással leemeli a pénzt a cég bankszámlájáról. Ha egy céggel kapcsolatosan háromszor előfordul, hogy nem fizet, akkor az ABC az együttműködést felmondja. Az ABC a cégekkel az együttműködést évente megújítja, de ez nem vonatkozik azokra, akikkel az együttműködést az ABC felmondta. Bizonyos idő elteltével, a megfelelő feltételek esetén a korábban felmondott cégekkel ismét hajlandó szerződést kötni az ABC.

Definiálja az alábbi funkciókat ellátó processzeket:

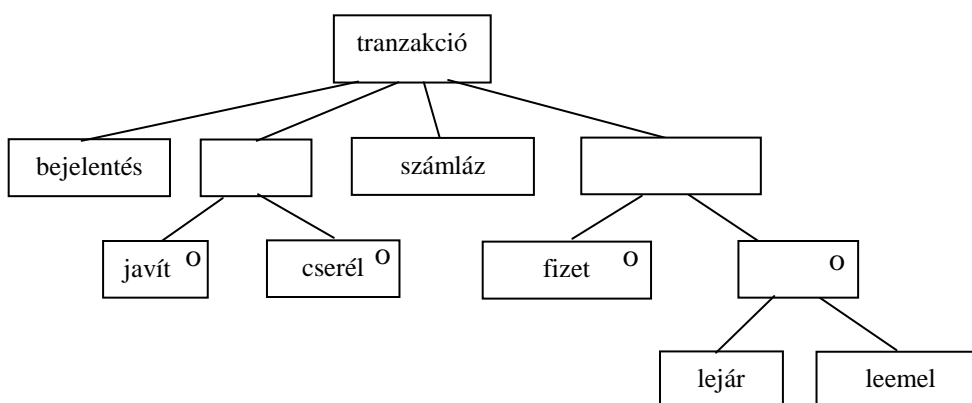
Automatikusan generáljon felmondást, amikor harmadik alkalommal kell leemelni a cégtől a pénzt !

Listázza ki azokat a kivonulásokat, ahol cserére volt szükség és még legalább 3 nap van hátra a fizetési határidő lejártáig !

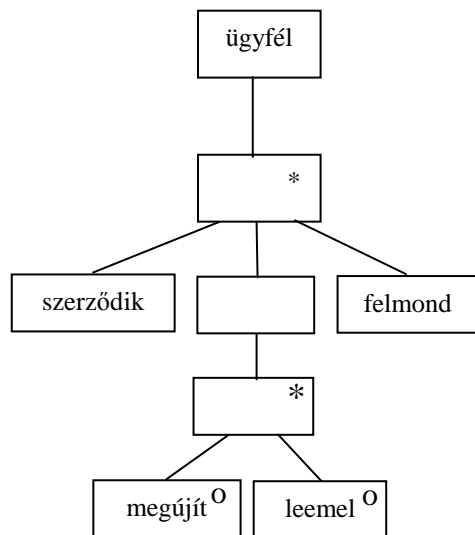
Azonnal jelezzen, ha olyan ügyfél telefonál, akivel az ABC az együttműködést korábban bármikor felmondta !

Készítse el az **entitás/akció leírást,**
az entitás struktúra diagramot,
az initial modellt,
kapcsolja a modellhez a **funkció processzeket.**

- bejelent (tranzakció) 2 pont
- javít (tranzakció)
- cserél (tranzakció)
- számláz (tranzakció)
- fizet (tranzakció)
- lejár (tranzakció) - nem fizet –NEM JÓ**
- leemel (tranzakció, ügyfél)
- szerződik (ügyfél)
- megújít (ügyfél)
- felmond (ügyfél)

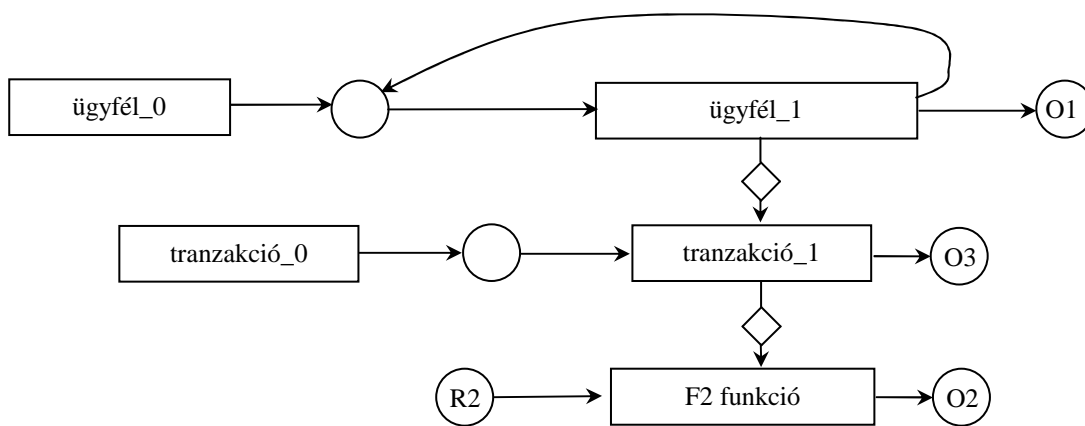


4 pont



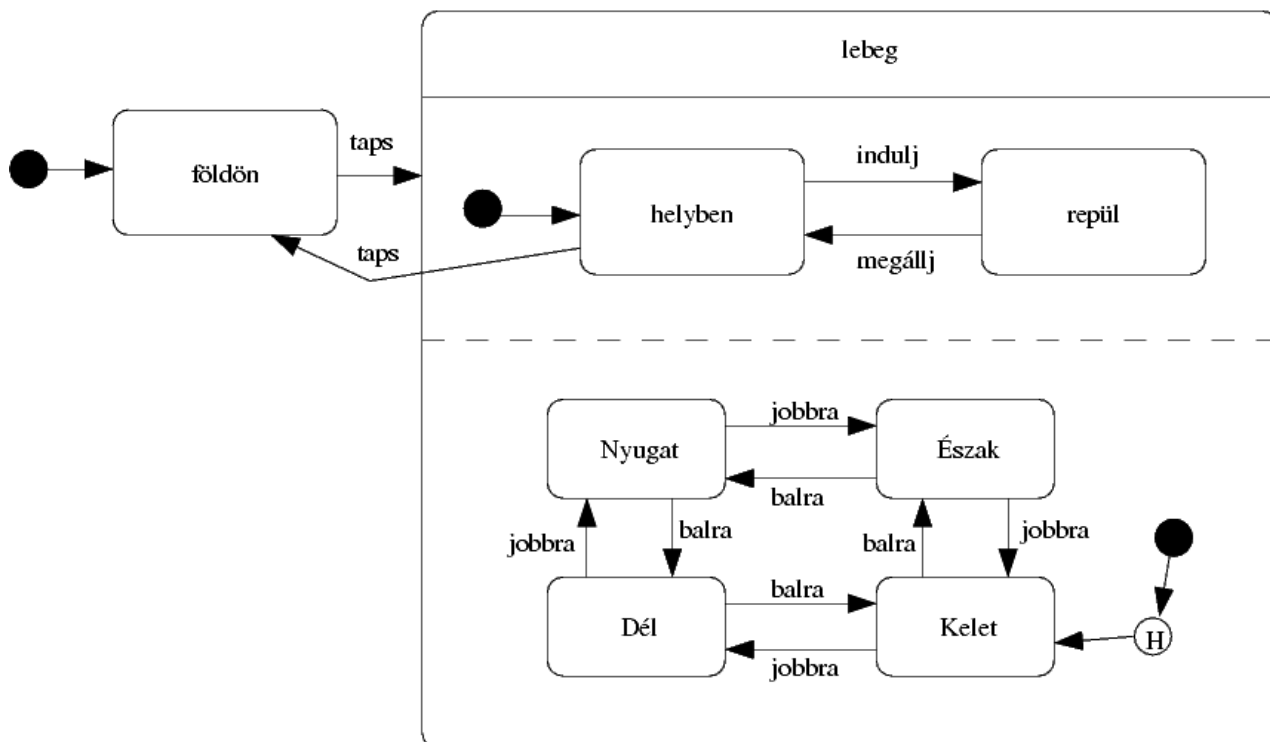
3 pont

2 pont



7. Rajzoljon az alábbi történetnek megfelelően UML state-chartot! (8 pont)

Harún ar-Rasíd repülő szőnyeget kap ajándékba tengerjáró Szindbádtól. A szőnyeg alapesetben a földön pihen. Tapsra felemelkedik, lebegni kezd. Ekkor az 'indulj' parancsra elindul, a 'megállj'-ra megáll, és ismét csak lebeg. Ha egyhelyben lebeg, tapsra a földre ereszkedik. Mikor lebeg vagy repül, az iránya módosítható: a 'jobbra' illetve 'balra' szavakra a megfelelő égtáj fele (pl. 'jobbra' esetén északról keletre) fordul. Mikor a földről felemelkedik, abba az irányba áll, amiben utolsó lebegésekor volt. Szindbád elmondta, hogy a szőnyeg legelőször kelet fele repült.



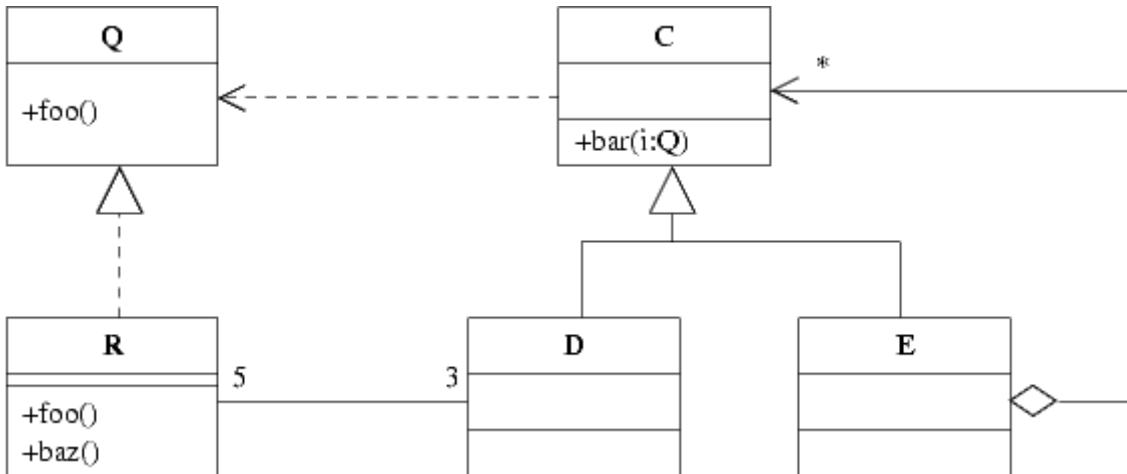
égtáj körbejár – 2 pont, belépés, history – 2 pont

áll-mozog, belépéssel – 1 pont

2 régió – 2 pont

föld és átmenetek – 1 pont

8. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)
F - az ki van zárva, mert nem tudom)

- [B]** D helyettesíthető E-vel, mert közös az ősük.
[D] E bar metódusa kaphat paraméterül R-t, mert R megvalósítja a Q interfészt.
[A] Egy R pontosan 3 darab D-vel állhat kapcsolatban, mert az asszociáció irányított.
[D] E aggregálhat más E-ket, mert E C leszármazottja.
[B] R nem hívhatja meg D bar metódusát, mert nem ismeri D ősét.
[C] D létrehozhat R-t, mert ismeri a Q interfészt.
[B] E meghívhatja R baz metódusát, mert D közvetlenül ismeri R-t.
[E] C bar metódusa paraméterül kaphat D-t, mert C megvalósítja a Q interfészt.

Blank és F az 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

ZH FELADATSOR
SZOFTVERTECHNOLÓGIA

c. tárgyból

2007. október 27. 14:15

A szöveges választ igénylő feladatokra a magyar nyelv szabályainak megfelelő, olvasható MONDATokkal válaszoljon ! A válasz ne legyen hosszabb összesen 24 szónál !

1. Mi az Ariane 501 esettanulmányból leszűrhető legfontosabb tanulság ? (2 pont)

Egy hétköznapi, elemi hiba - tervezési és menedzsment hibák és tévedések sorozatán keresztül felnagyítva - katasztrófához vezethet. A technológia feladata az, hogy a veszélyes láncolatok kialakulását megakadályozza.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

2. Mire, hogyan használjuk a tudományt a technológiában ? (2 pont)

A tudomány felhasználásával modellt alkotunk, majd a modellt tudományosan megalapozott és igazolt módszerekkel ellenőrizzük.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

3. Jellemezze a CMM negyedik (menedzselt) szintjét! (2 pont)

The defined software processes are instrumented and controlled quantitatively. Managers are able to measure progress and problems.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

4. Mit jelent a "formalizáltság" ? (2 pont)

Annak mértéke, hogy mennyire szigorú formai szabályoknak, előírásoknak felel meg egy leírás, reprezentáció.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

5. Mit jelent a "csatolás" fogalma ? (2 pont)

egy (al)rendszer komponensei között kapcsolat erőssége.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értékén történő elfogadását.]

6. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a,c*)|b)>
  <!ATTLIST a d CDATA #IMPLIED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <a></a>
  <c>zh</c>
</x>
```

- IGEN
- NEM

Hiba:

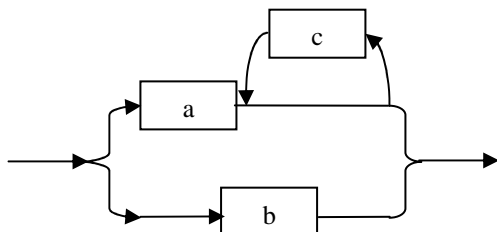
Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van attribútum is ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <a d="x">aa</a>
</x>
```

Írja fel az adatszerkezet algebrai definícióját ! (2 pont)

$$x = [(a + \{c\}) | b]$$

Rajzolja fel az adatszerkezet szintaxisgráfját ! (2 pont)



7. Jellemezzünk egy stringet az alábbi műveletekkel ! Adja meg a TAIL műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket. (4 pont)

- NEW()** új (üres) stringet hoz létre.
- ADD(s,x)** az **s** string végére rakja az **x** karaktert.
- LGTH(s)** az **s** string karaktereinek számát adja.
- TAIL(s, n)** az **s** string legrégebbi **n** darab karakterének levágása után maradó stringet adja
Ha **n** nem kisebb, mint **s** string hossza, üres stringet kapunk. Tételezze fel, hogy **n** > 0 !

TAIL(NEW()) = NEW()		-0.5 pont, ha hiányzik
TAIL(ADD(s, x), n) ==	if (n > LGTH(s)) NEW()	1 pont
	if (n == LGTH(s)) ADD(NEW(), x)	1 pont
	if (n < LGTH(s)) ADD(TAIL(s, n), x)	2 pont

8. Mi a specifikálás lényege ? (2 pont)

A követelményeket kielégítő és a korlátozásokat betartó rendszer formális leírásának elkészítése.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

Mely fázis előzi meg és követi a specifikálást ? (2 pont)

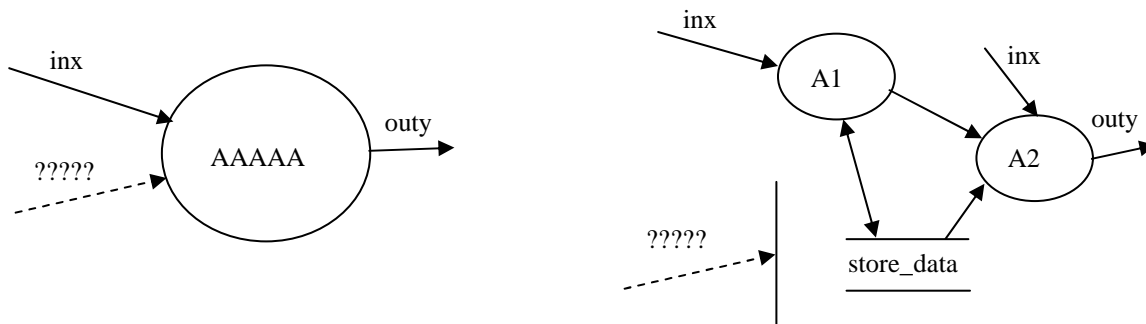
Előző: **követelmény** Következő: **tervezés**

9. Mi a kockázat elemzésének célja ? (2 pont)

Assess probability and seriousness of each risk

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

10. Tételezzük fel, hogy az AAAAA folyamatot DFD-ként akarjuk megjeleníteni a jobboldali ábrarészlet szerint. Mi a ????-el jelölt ábraelem ? Kapcsolja a kért ábraelemet korrekt módon a jobboldali ábrához ! Van-e szintaktikai hiba a jobboldali ábrán ? (3 pont)



Vezérlő folyamat, vezérlés, NINCS.....

11. A Grimbusz társaság aktuális és tervezett buszjáratait megtaláljuk a menetrendben. A járat jellemzője a járatszám, az induló és célállomás. Egy adott napon a járatra vonatkozóan nyilvántartjuk a késést. Egy napon nem indul többször ugyanazon számú járat. A konkrét napi járatokhoz a buszt és a vezetőt az éppen rendelkezésre állók közül választják. Egy konkrét járatnál csak egy vezető megy. Minden vezetőről tudjuk, hogy melyik járaton hány alkalommal vezetett. A napi járatokat 60 nap elteltével töröljük. Rajzoljon entitás-relációs diagramot ! (5 pont)

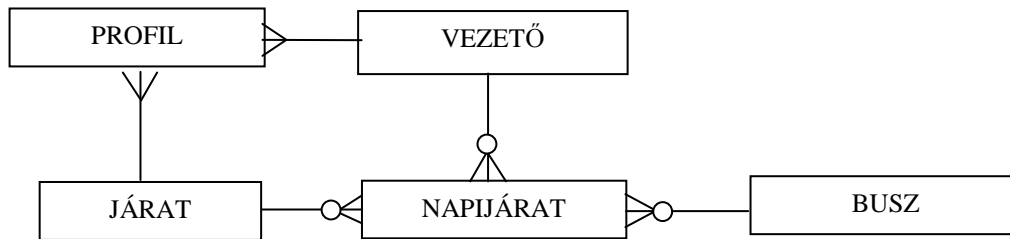
JÁRAT (szám, indulás, cél)

NAPIJÁRAT (járat, dátum, késés, *busz*, *vezető*)

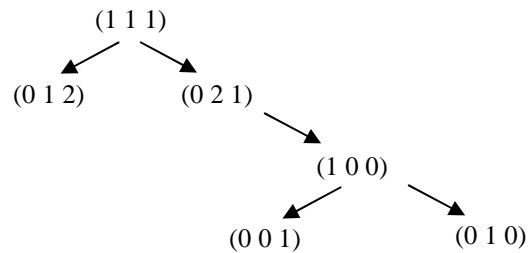
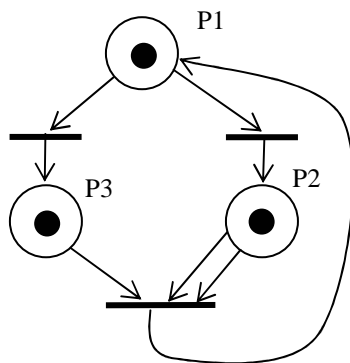
BUSZ (rendszám, megtett km)

VEZETŐ (név, jogszi száma)

PROFIL (járat, vezető, darab)



12. Rajzolja fel az alábbi Petri háló elérhetőségi gráfját (reachability tree) ! (2 pont)



13. Adjon példát arra, amikor az adatfluxus meghatározó követelmény ! (2 pont)

SZJA bevételek 90%-a 10 nap alatt érkezik be.

0. hét a neptunon.

Minimálisan elérendő pontszám: 16/40

Vizsgára továbbított pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{2})$

ZH FELADATSOR
SZOFTVERTECHNOLÓGIA

c. tárgyból

2007. október 27. 15:15

A szöveges választ igénylő feladatokra a magyar nyelv szabályainak megfelelő, olvasható MONDATokkal válaszoljon ! A válasz ne legyen hosszabb összesen 24 szónál !

1. Mi a Mars Climate Orbiter esettanulmányból levonható legfontosabb tanulság ? (2 pont)

Egy hétköznapi, elemi hiba - tervezési és menedzsment hibák és tévedések sorozatán keresztül felnagyítva - katasztrófához vezethet. A technológia feladata az, hogy a veszélyes láncolatok kialakulását megakadályozza.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

2. A technológia fogalmának definíciójában szerepel az “emberiség szolgálatában” (in the service of mankind) kitétel. Miben nyilvánul meg ez a kitétel a mindennapi mérnöki munkában ? (2 pont)

A társadalmi elvárások jogszabályokban, szabványokban, egyéb előírásokban jelennek meg. Ezeket be kell tartani.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

3. Mi a neve és a lényege a CMM 3. szintjének (level 3) ! (2 pont)

Defined - definiált

The software process activities are documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard process for developing and maintaining software.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

4. Mit jelent az implementációs függőség ? (2 pont)

Annak mértéke, hogy a feladat leírásában szereplő entitások (pl. dolgok, események, kapcsolatok) mennyire függenek/függetlenek a megvalósítás során alkalmazott (számítástechnikai, informatikai) entitásoktól.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

5. Egy A program miközben meghívja a B functiont, paraméterül egy C struct-ot ad át. Milyen a hívó és hívott (A és B) közötti csatolásban (coupling) a "kommunikáció fajtája" (kind of communication) ? (1 pont)

- | | |
|---|--|
| <input type="checkbox"/> Content (tartalom) | <input type="checkbox"/> Common (közös) |
| <input type="checkbox"/> Logic (logika) | <input type="checkbox"/> Data (adat) |
| <input type="checkbox"/> Control (vezérlés) | <input checked="" type="checkbox"/> Stamp (bélyeg) |

6. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a,c*)|b)>
  <!ATTLIST a d CDATA #REQUIRED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <a d></a>
  <c>zh</c>
</x>
```

- IGEN
 NEM

Hiba: A paraméterből hiányzik az egyenlőségjel és az érték.

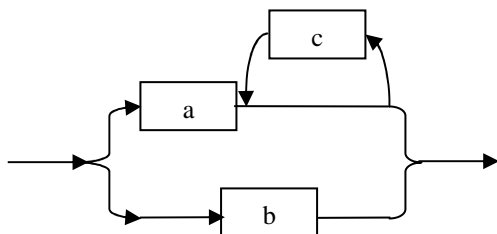
Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van c elem ! (3 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <a d="x">aaa</a>
  <c>ccc</c>
</x>
```

Írja fel az adatszerkezet algebrai definícióját ! (2 pont)

$$x = [(a + \{c\}) | b]$$

Rajzolja fel az adatszerkezet szintaxisgráfját ! (2 pont)



7. Jellemezzünk egy listát az alábbi műveletekkel ! Adja meg a CUT műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket. (4 pont)

- CRT()** új (üres) listát hoz létre.
- SET(l,x)** az l lista elejére rakja az x elemet.
- LGTH(l)** az l lista elemeinek számát adja.
- CUT(l, n)** az l lista legrégebbi n darab elemének levágása után maradó listát adja
Ha n nem kisebb, mint l lista hossza, üres listát kapunk. Tételezze fel, hogy $n > 0$!

- CUT(CRT()) = CRT() -0.5 pont, ha hiányzik
- CUT(SET(l, x), n) == if (n > LGTH(l)) CRT() 1 pont
- if (n == LGTH(l)) SET(CRT(), x) 1 pont
- if (n < LGTH(l)) SET(CUT(l, n), x) 2 pont

8. Mi a szoftver tervezés lényege ? (2 pont)

A szoftver termék belső struktúrájának, a feldolgozás részleteinek kidolgozása, valamint a termék létrehozásához szükséges lépések, folyamatok részletes kimunkálása.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

Mely fázis előzi meg és követi a tervezést ? (2 pont)

Előző: **specifikáció**

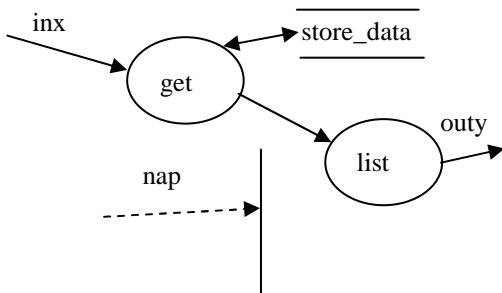
Következő: **implementáció**

9. Mi az eshetőségi terv (contingency plan)? (2 pont)

Egy kockázati esemény bekövetkezte esetére definiálja a projekt folytatásának lépéseit.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

10. Definiálja a vezérlőegység egy lehetséges állapotábláját úgy, hogy az konzisztens legyen (ne legyen ellentmondásban) a DFD-vel és az adatspecifikációval (3 pont)



A			
B			

nap = [munkanap | szünnap]

az oszlopok paraméterezése (sorrend lényegtelen) csak a nap konstansai lehetnek
a következő állapotok csak a bejelölt két állapot (A, B) vagy don't care lehet, kell benne lenni állapotváltásnak is.
az akció ha nem üres, csak gombóc (get, list) lehet és mindkettőnek legalább egyszer szerepelni kell.

11. A Globe színházban esténként egy előadás van, de egyszerre több darabot is műsoron tartanak. Egy színdarabot jellemez a címe, szerzője, rendezője és a bemutató éve. Az esti előadásról nyilvántartják a nézők számát, illetve azt, hogy történt-e rendkívüli esemény. Az előadást esténként egy ügyelő irányítja. A színháznak több ügyelője van, mindegyikük az összes futó darabot ügyelheti. Az ügyelőkről tudjuk, hogy az egyes darabokat már hányszor ügyelték. Egy előadásra legalább egy, de inkább több színész szerepel. Az előadásokra vonatkozó adatokat 60 nap elteltével töröljük. Rajzoljon entitás-relációs diagramot ! (5 pont)

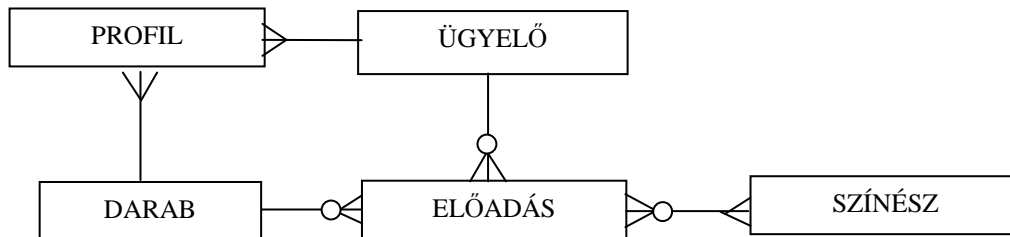
DARAB (cím, szerző, rendező, bemutató éve)

ELŐADÁS (darab, dátum, nézők száma, esemény, színész, ügyelő)

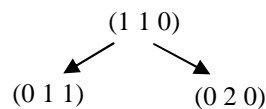
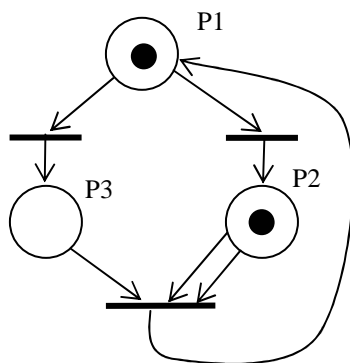
SZÍNÉSZ (név, szerepek száma)

ÜGYELŐ (név, ügyelések száma)

PROFIL (darab, ügyelő, darab)



12. Rajzolja fel az alábbi Petri háló elérhetőségi gráfját (reachability tree) ! (2 pont)



13. A követelmények definiálásakor milyen típusú hibák fordulnak elő ? (2 pont) !

ténybeli tévedések (hamis tények) elhanyagolások, kihagyások, felejtések

inkonzisztencia (ellentmondások)..... többértelműség.....

Minimálisan elérendő pontszám: 16/40

Vizsgára továbbvitt pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{2})$

ZH FELADATSOR
SZOFTVERTECHNOLÓGIA

c. tárgyból

2006. december 1. 14:15

1. A "V model" szoftver életciklus modell alapján milyen tesztelési szinteket azonosíthatunk (4 pont)

unit test..... integration test.....
system test..... acceptance test.....

2. Egy adatfolyam-ábrán szereplő vezérlőegységet az alábbi állapotgéppel írunk le:

	F5	xxx	LO
X1	X1/store	X2/	X1/yy
X2	X2/	X2/tar2	X1/tar2

Feltéve, hogy a DFD-n csak az adott állapotmodellel kapcsolatos elemek állnak

Specifikálja a vezérlő egység bemenetét ! (2 pont): [F5|xxx|LO]

Sorolja fel a processzeket ! (2 pont): store, tar2, yyy

Adja meg az adattárákat ! (1 pont): nem tudjuk

3. Jellemezze a CMM negyedik (menedzselt) szintjét! (3 pont)

The defined software processes are instrumented and controlled quantitatively. Managers are able to measure progress and problems.

4. Az alábbi XML leírás jól formált? Ha nem, akkor mi a baja? (2 pont)

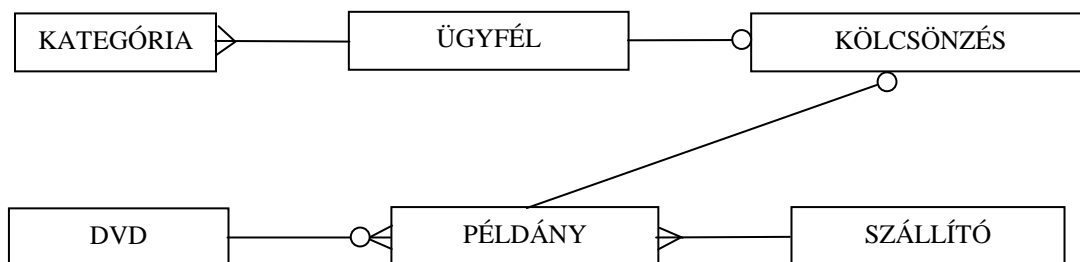
```
<?xml version="1.0"?>  
<zh>  
<most>Szoftvertechnologia</most>  
<kod>viia217</kod><zh/>  
<most>december elsejen</most>  
<torzs>penteken tartjuk</torzs>  
</zh>
```

- IGEN
- NEM

.....

5. Egy videotékában DVD-ket lehet kölcsönözni. A DVD-kről nyilvántartják a címét, a kiadóját, a kiadás évét, a műsor hosszát. Egy DVD-ből több példány is lehet, de olyan DVD-k adatait is tárolják, amelyekből még vagy már nincs példány. Példányonként tárolják a vásárlás idejét, az árat, a szállítót és az adott példány kölcsönzéseinek számát. A szállítónak (már szállított) ismerik a nevét, címét és bankszámla számát. Az ügyfelet azonosító jellemzi, de nyilvántartják a nevet, címet, telefonszámot. Egy kölcsönzés alkalmával egy ügyfél egy DVD példányt megadott határidőig kölcsönbe vesz. A kölcsönzés megszűnik, ha az ügyfél a DVD-t visszahozta. Az ügyfelekről profilt készítenek, amely tartalmazza, hogy az ügyfél az élete során a különböző kategóriájú (pl.: zene, akció, romantikus, vígjáték, krimi, thriller, sex, stb.) DVD-kből hányszor kölcsönzött.

Az alábbi ERD ábrán szerepelnek az entitások, de a relációk nem. Rajzolja be a relációkat, és jelölje számosságukat! (4 pont)



Minden reláció 1 pont (5 pont van benne)

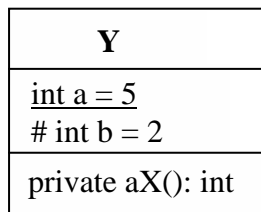
6. Jellemezzünk egy stringet az alábbi műveletekkel! Adja meg a TAIL műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket. (4 pont)

NEW() új (üres) stringet hoz létre.
ADD(s,x) az s string végére rakja az x karaktert.
LGTH(s) az s string karaktereinek számát adja.
TAIL(s, n) az s string legrégebbi n darab karakterének levágása után maradó stringet adja
 Ha n nem kisebb, mint s string hossza, üres stringet kapunk. Tételezze fel, hogy $n > 0$!

TAIL(NEW()) = NEW()		-0.5 pont, ha hiányzik
TAIL(ADD(s, x), n) ==	if (n > LGTH(s)) NEW()	1 pont
	if (n == LGTH(s)) ADD(NEW(), x)	1 pont
	if (n < LGTH(s)) ADD(TAIL(s, n), x)	2 pont

7. Elkészítjük az alábbi Y osztály két példányát, y1-et és y2-t. Ezt követően végrehajtjuk a következő műveleteket:

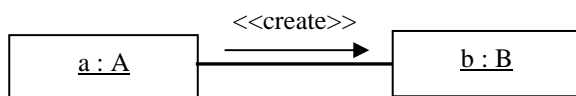
```
y2.a = 3; y1.a = -3;
y1.b = y2.a + 4;
y2.b = y2.a + y1.b;
```



Mennyi lesz a változók értéke ? (4 pont)

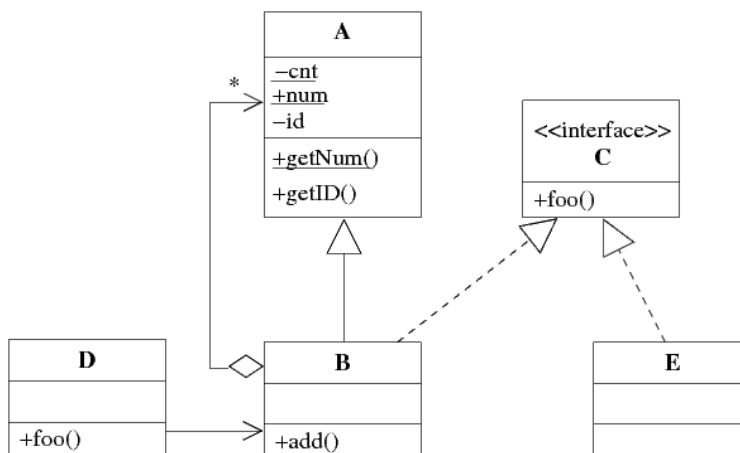
y1.b = **1** y2.b = **-2**.....

8. Az A osztályú objektum egy példány létrehoz egy B osztályú objektum példányt. Jelölje A és B közötti kapcsolatot UML kollaborációs diagramon. (2 pont)



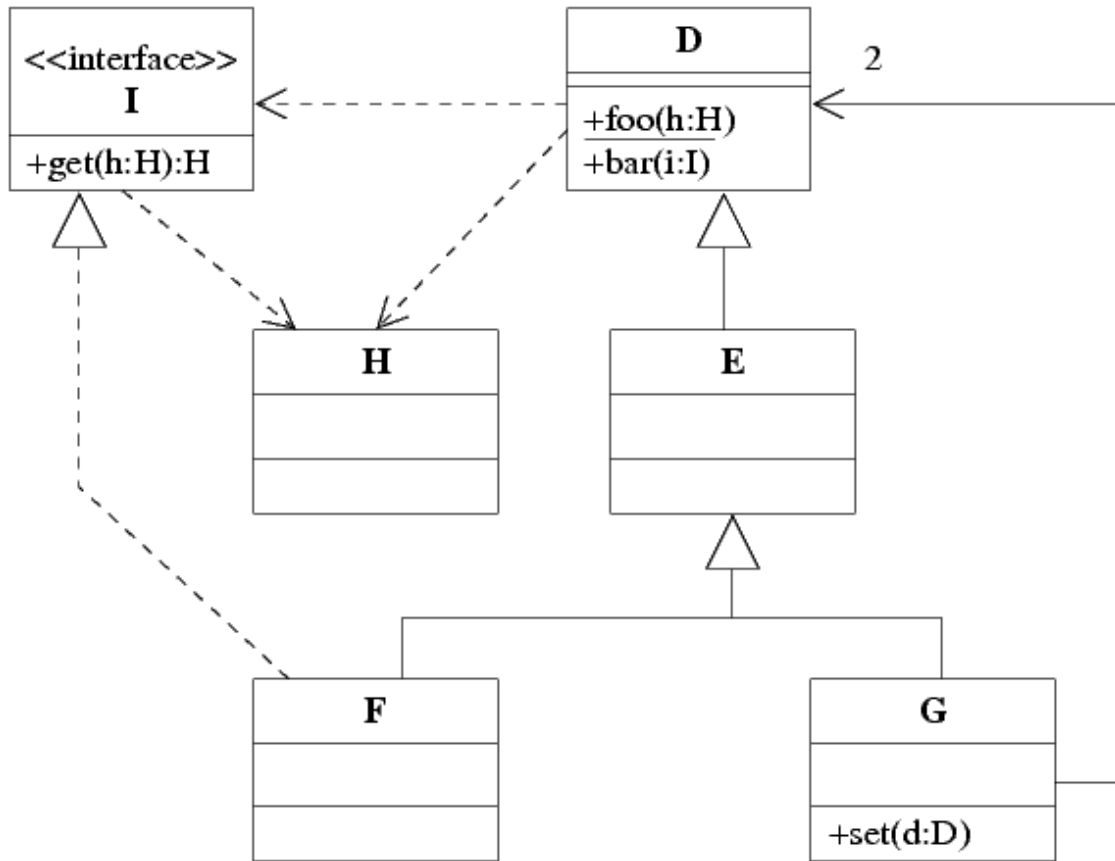
9. Jelölje be, hogy a mellékelt ábrán milyen UML elemek láthatóak (4 pont)

- aggregáció
- függőség
- sztereotípiá
- példányosítás
- tagged value
- objektum
- absztrakt objektum
- osztály metódus
- kvalifikáció
- private attribútum
- terminátor



minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B] F helyettesíthető G-vel, mert D közös ősük.
- [E] F-nek nincs mindig `get` metódusa, mert csak feltételesen örököl I-től.
- [A] H nem hívhatja meg D `foo` metódusát, mert a metódus absztrakt.
- [C] D `bar` metódusa kaphat paraméterül F-et, mert F leszármozottja D-nek.
- [A] D `foo` metódusa nem kaphat paraméterül I-t, mert `foo` nem osztály-metódus.
- [C] G kétszer is meghívhatja D `bar` metódusát, mert G D-nek a leszármozottja.
- [E] G nem tud H paramétert átadni D `foo` metódusának hívásakor, mert G nem ismeri H-t.
- [B] F meghívhatja G `set` metódusát, mert mindketten függnek az I interfésztől.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total ≥ 0

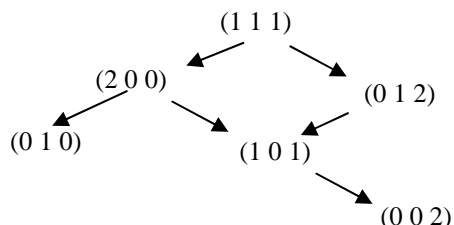
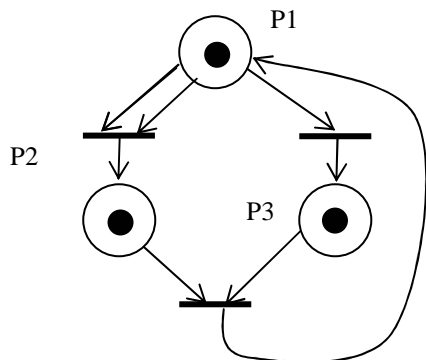
Minimálisan elérendő pontszám: 16/40

Vizsgára továbbvitt pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{2})$

ZH FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2006. december 1. 15:15

1. Rajzolja fel az alábbi Petri háló elérhetőségi gráfját (reachability tree) ! (3 pont)



2. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

gyakorlati problémák megoldására szolgáló **dolgok**

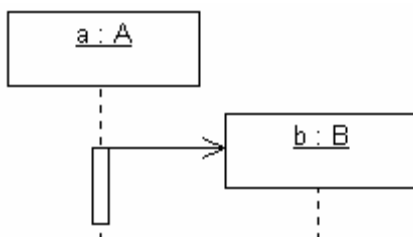
a **társadalom** által szabályozott **tudományos** alapokon álló

gazdaságos

előállításának mikéntje

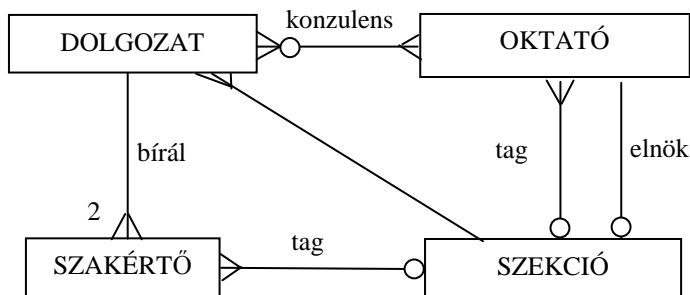
mindegyik komponens 1 pont.

3. Az A osztályú objektum egy példánya létrehoz egy B osztályú objektum példányt. Jelölje A és B közötti kapcsolatot UML szekvencia diagramon. (3 pont)



4. A kari TDK konferenciára száznál több dolgozatot nyújtottak be. A dolgozatok oktatók irányításával (konzultálásával) készültek. Egy oktató több dolgozatot is konzultálhatott, és esetenként van olyan dolgozat is, aminek több konzulense is van. A bíráló objektivitása érdekében minden dolgozatot pontosan két egyetemen kívüli szakértővel bíráltatnak el. Egy szakértő csak egyetlen dolgozatot bírál. Minden dolgozatot a témája alapján egy névvel ellátott szekcióba sorolják; egy szekcióba 7-10 dolgozat kerül. Egy szekció munkáját az elnök irányítja, aki egyetemi oktató. A szekció munkájában szekciónként részt vesz még legalább egy oktató és legalább egy, a bírálásban is érintett szakértő.

Az alábbi ERD ábrán szerepelnek az entitások, de a relációk nem. Rajzolja be a relációkat, és jelölje annak számosságát ! (5 pont)



Minden reláció 1 pont (6 pont van benne)

5. Mik a csatolás dimenziói ? (3 pont)

Kind of communication Size of connection.....

Time of connection.....

Egy A objektum miközben a B egy metódusát hívja, paraméterül egy C objektumot ad át. Milyen a hívó és hívott objektumok közötti csatolás (coupling) ? (2 pont)

- | | |
|--|---|
| <input type="checkbox"/> Content (tartalom) | <input type="checkbox"/> Common (közös) |
| <input checked="" type="checkbox"/> Stamp (bélyeg) | <input type="checkbox"/> Control (vezérlés) |
| <input type="checkbox"/> Data (adat) | <input type="checkbox"/> Logic (logika) |

6. Mi a döntés hasítás ? (3 pts)

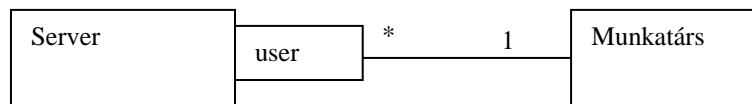
Vezérlési hatáskör < Döntési hatáskör

7. Az alábbi XML leírás jól formált ? Ha nem, akkor mi a baja ? (2 pont)

```
<?xml version="1.0"?>
<zh>
<most>Szoftvertechnologia</most>
<most/>viiaa217<most/>viiaa217
<most>december elsejen
<torzs>penteken tartjuk</torzs>
</most></zh>
```

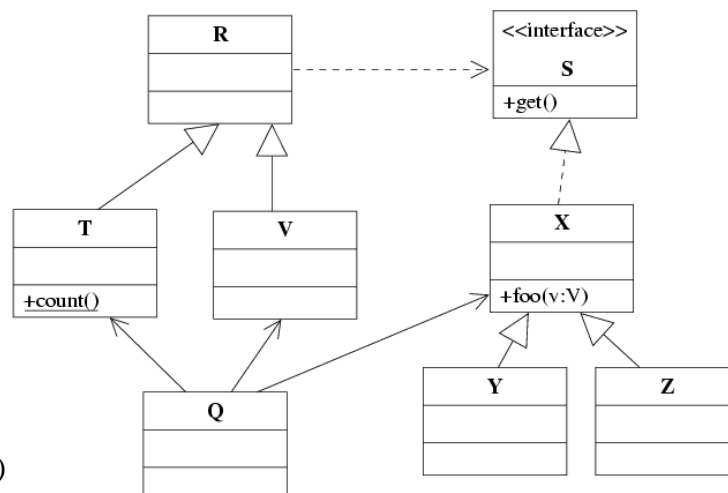
- IGEN
 NEM

8. Egy munkatárs több szervert is elérhet, és egy szerverhez sok munkatárs kapcsolódik. Rajzolja fel a szervert, a munkatársat és a közöttük fennálló kapcsolatot UML osztálydiagramon, minősítőként (qualifier) használva a user nevet ! Jelölje a multiplicitást ! (2 pont)



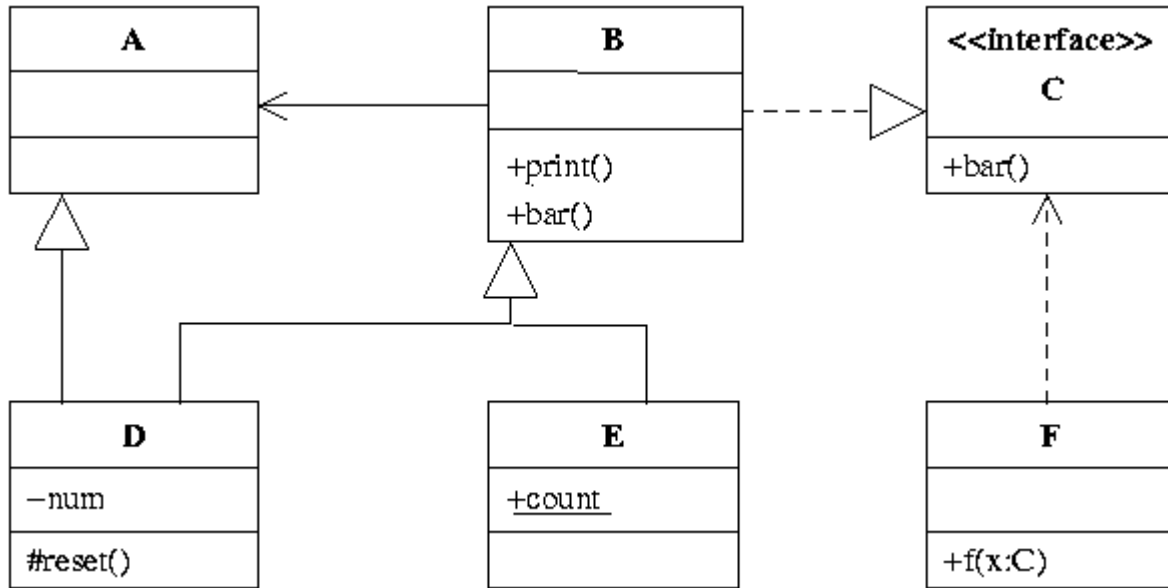
9. Jelölje be, hogy a mellékelt ábrán milyen UML elemek láthatóak (4 pont)

- aggregáció
- függőség
- osztály metódus
- példányosítás
- qualifier (minősítő)
- objektum
- asszociáció
- kollaboráció
- iteráció
- protected metódus
- realizálás (implementálás)



minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B] D helyettesíthető E-vel, mert van közös ősük.
- [E] D ismeri E-t, mert D egyik őse (A) ismeri E ősét.
- [D] F f metódusa kaphat paraméterül D-t, mert D megvalósítja a C interfészt.
- [A] B ismeri A-t, mert B komponense A.
- [B] F hozzáfér E count attribútumához, mert E megvalósítja a C interfészt.
- [B] A is megvalósítja C interfészt, mert van B-vel közös leszármazottja (D).
- [A] E-nek van osztály-attribútuma, ezért E-t nem lehet példányosítani.
- [B] D reset metódusából nem érjük el D num attribútumát, mert a num privát.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Minimálisan elérendő pontszám: 16/40

Vizsgára továbbvitt pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{2})$

PPZH FELADATSOR
SZOFTVERTECHNOLÓGIA

c. tárgyból

2007. december 20. 18:00

1. Jelölje az állítások mellett 1-5-ig, hogy minimálisan melyik CMM szinttől igazak ! Ha az állítás nem értelmezhető, akkor tegyen X-et ! (7 pont)

minden rossz válasz a helyes válaszáért járó pont felének elvonásával jár, de total ≥ 0

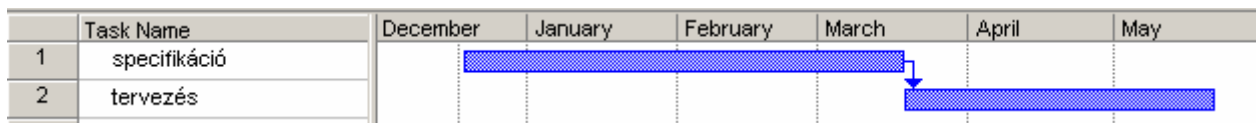
5	Ellenőrzött módon új és javított szoftverfejlesztési eszközöket próbálnak ki
2	A vevői igények és a munkatermékek már kontrolláltak
X	A folyamat számszerű mérése lehetővé teszi az ipari szabványok javítását
2	Alapvető projekt-menedzsment gyakorlatok kialakultak
3	A szoftverfolyamat tevékenységei már beintegrálódtak a szervezet szabványos szoftver-folyamatában
5	Számszerű visszacsatolás segíti az állandó folyamatfejlesztést
4	A szoftverfolyamatról módszeresen adatokat gyűjtenek
2	Lehetővé vált a költségek, az ütemterv és a funkcionalitás nyomonkövetése
4	A vezetők képesek a folyamatok előrehaladásának és a problémáknak a mérésére
X	A termék életciklusának tervezése beépült a folyamatfejlesztés részfeladatai közé
5	Innovatív ötletek és technológiák segítik az állandó folyamat-fejlesztést
X	A vezetők képesek a termék minőségének közvetlen ellenőrzésére
3	Minden projekt a szervezet szabványos fejlesztési és karbantartási folyamatának egy jóváhagyott, személyre szabott verzióját követi
X	A mérések eredményei lehetővé teszik a hatékony vezetőváltást

2. A szoftver fejlesztési folyamat V-modelljében a béta teszt a szoftver életciklus mely fázisainak ellenőrzésére szolgál ? (3 pont)

minden rossz válasz a helyes válaszáért járó pont felének elvonásával jár, de total ≥ 0

- Követelmény
- Specifikáció
- Architektúrális tervezés
- Részletes tervezés
- Implementáció
- Nincs ilyen teszt

3. Egy szoftver projektben a specifikáción 3 ember 3 hónapig dolgozik, majd a tervezésen 2 ember 2 hónapig. Rajzoljon Gantt-diagramot ! (4 pont)



4. Az alábbi DTD leírás helyes-e ? Ha nem, akkor mi(k) a hiba(k)? (3 pont)

```
<!DOCTYPE EMPLOYEES [
<!ELEMENT COMPANY (DIVISION+)>
<!ELEMENT DIVISION EMPTY>
<!ATTLIST COMPANY NAME #FIXED>
<!ATTLIST DIVISION EMPS CDATA
#IMPLIED> ]>
```

- IGEN
 NEM

Hiba: EMPLOYEES hiányzik, NAME attrib típusa hiányzik, FIXED értéke hiányzik.....

5. Legyen az alábbi XML szerkezetünk !

```
<?xml version="1.0" ?>
<note>
  <to>Jack</to>
  <from>John<!-- Doe -->
  </from><heading>Reminder
  </heading>
  <body>Don't forget
  "<me name="John" />"
  this weekend!</body>
</note>
```

A fenti XML leírást alapul véve jelölje az IGAZ állításokat ! (3 pont)

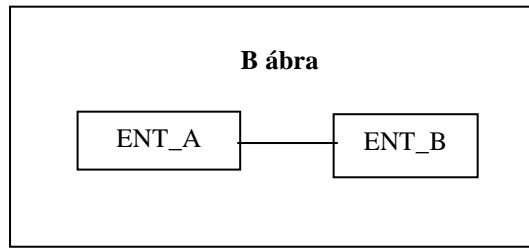
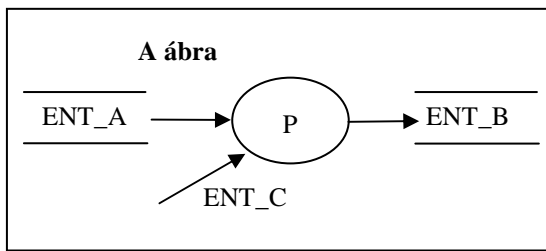
minden rossz válasz a helyes válaszáért járó pont felének elvonásával jár, de total >= 0

- A leírás jól formált
- A leírás hibás, mert a komment nem megfelelő helyen áll
- A leírás hibás, mert a me elem nem tehető idézőjelek közé
- A leírás hibás, mert a body elem nem lehet összetett
- A leírás hibás, mert a dokumentum szerkezete nem ismert
- A leírás hibás, mert

6. Az alábbi két ábra (A ábra DFD, B ábra ERD) alapján jelölje meg az IGAZ állításokat ! (4 pont)
 Tételezze fel, hogy az azonos nevű elemek ugyanazon dolgokat jelentik és semmiféle korlátozó feltételt nem támasztunk ! Hagyja figyelmen kívül a multiplicitást !

minden rossz válasz a helyes válaszáért járó pont felének elvonásával jár, de total >= 0

- Az A ábrához csakis és kizárólag a B ábra tartozhat
- A két ábra akkor nem mond ellent egymásnak, ha az A ábrán is összekötjük ENT_A-t és ENT_B-t
- Az A ábra és a B ábra között ellentmondás van, mert
- A két ábra között nincs ellentmondás
- Az A ábra hibás, mert
- A B ábráról hiányzik ENT_C
- A B ábráról hiányzik P



7. A korszerű réteges (layered) architektúrában a GUI réteget mi valósítja meg ? (3 pont)

Böngésző, browser

8. Mi a kapcsolat az adatszerkezetet definiáló axiómák és a szerkezeten értelmezett műveletek típusa között ? (2 pont)

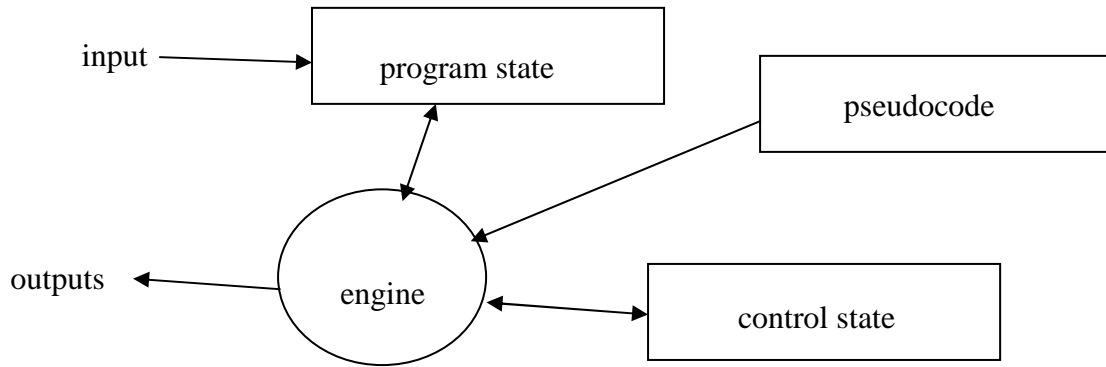
Behavior(Constructor)=
 Modifier(Constructor)=

9. Az ideális réteges architektúrában az i-ik réteg csak az i+1-ik réteg szolgáltatásait használja. Gyakorlati megfontolások miatt előfordul, hogy más rétegek szolgáltatásait is használni kell. Mi a legfőbb előnye és hátránya ennek az esetnek ? (3 pont)

Előny: teljesítmény javul

Hátrány: kompatibilitás megszűnik

10. Az interpreter szoftver architektúra általános vázlatát a következő !



Az alábbi állapotgéppel specifikált programot az interpreter architektúra szerint kívánjuk megvalósítani.

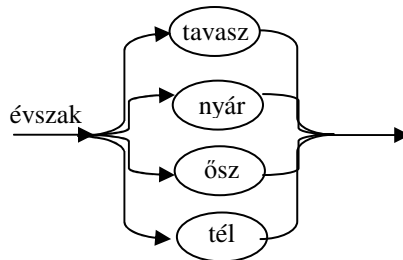
	X1	X2	X3
S1	S1/alfa	S2/	S1/beta
S2	S2/	S2/xxx	S1/alfa

A fenti állapottábla alapján specifikálja a

A "control state" lehetséges értékeit ! (2 pont): **[S1|S2]**

A "pseudocode" utasításait ! (4 pont): **S1/alfa, S2, S1/beta, S2/xxx**

11. Rajzolja fel az „évszak” enumeráció szintaxisgráfját ! (2 pont)



Minimálisan elérendő pontszám: 16/40

Vizsgára továbbvitt pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{8})$

PÓTZH FELADATSOR SZOFTVERTECHNOLÓGIA

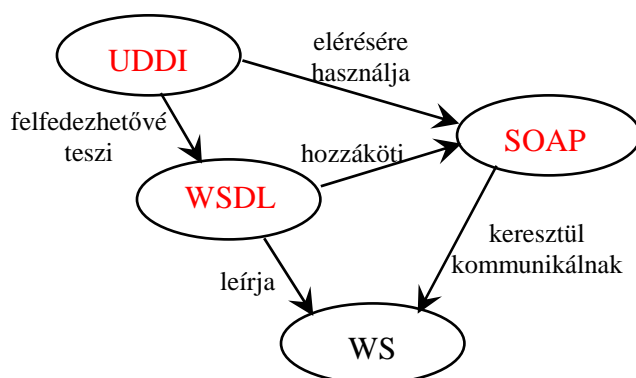
c. tárgyból

2007. november 9. 16:15

1. Minek a rövidítése a SOAP (a w3c ajánlás) ? (2 pont)

- Service Oriented Architecture Program
- Simple Object Architecture Profile
- Simple Object Access Protocol
- Service Object's Access Profile
- Service Oriented Access Protocol

2. A szolgáltatás-orientált architektúrát bemutató ábrán mik a név nélküli elemek ? (3 pont)



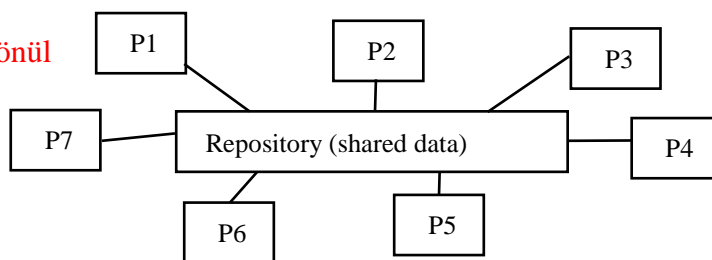
3. Jellemezze a Blackboard (tábla) szoftver architektúrát ! Adja meg a komponenseket és az architektúra alkalmazásának előnyeit ! (3 pont)

Komponensek:

- szeparált, független programok, alrendszerek, amelyek közös memórián keresztül működnek együtt
- központi adattár (adatbázis)

Előnyök:

- az adatok, a vezérlés és a feldolgozás jól elkülönül
- kisebb adathibák túrése és korrekciója

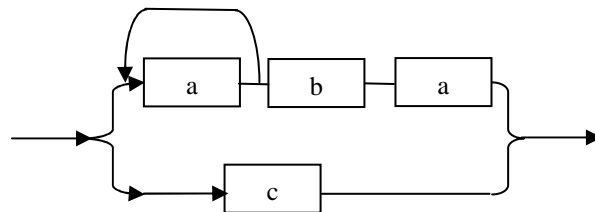


4. Az ISO 9126 szabvány szoftver termékre vonatkozó metrikákat definiál. Melyik CMM szint esetében meghatározó jelentőségű a termék minőségének mérése ? (2 pont)

Egyiken sem, mivel a CMM a FOLYAMATra fókuszál, nem a TERMÉKre.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

5. Egészítse ki az alábbi DTD vázlat úgy, hogy az a mellékelt szintaxisgráf szerinti adatszerkezetet definiálja ! A b elemnek legyen egy kötelező y attribútuma is ! (4 pont)



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
```

```
<!ELEMENT x ((a,a*,b,a)|c)>
<!ATTLIST b y CDATA #REQUIRED>
```

```
<!ELEMENT a (#PCDATA)>
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA)>
```


Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van b elem is ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <a></a>
  <b y="bbb"></b>
  <a></a>
</x>
```

Írja fel az adatszerkezet algebrai definícióját ! (2 pont)

$x = [(a + \{a\} + b + a) | b]$

6. Adatfolyam specifikációt készítünk. A megvalósítandó funkció kizárólag mozijegy eladása. A pénztárban ülő pénztáros mi lesz az ábrán ? (2 pont)

- terminátor
- processz
- entitás
- adatfolyam
- adattár
- egyik sem, hanem: **valamely processzek processzora**.....

7. Jellemezzünk egy stringet az alábbi műveletekkel ! Adja meg a PALIN műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja az egészek és a karakterek összehasonlítására szolgáló műveleteket. (4 pont)

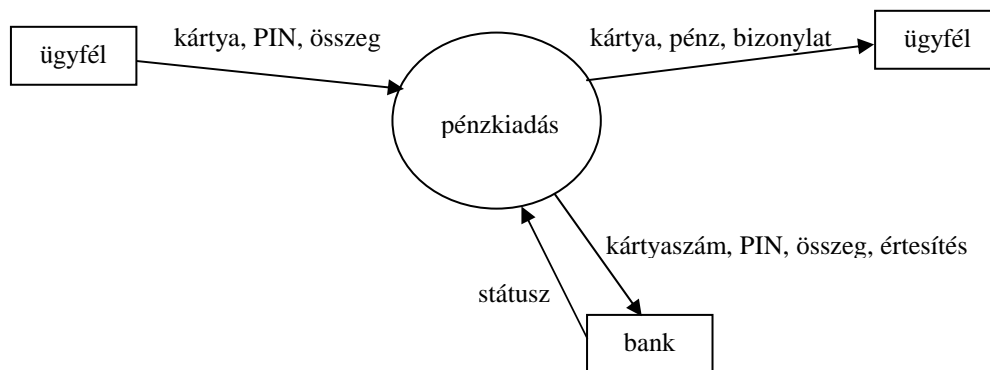
- CRT()** új (üres) stringet hoz létre
- LGTH(s)** az s string karaktereinek számát adja
- TAIL(s)** az s string első karakterének levágása után maradó stringet adja
- APPEND(s,x)** az s string végére rakja az x karaktert
- HEAD(s)** az s string első karakterét mutatja meg
- PALIN(s)** igaz, ha az s string palindróma

Egy string palindróma, ha az elejéről olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

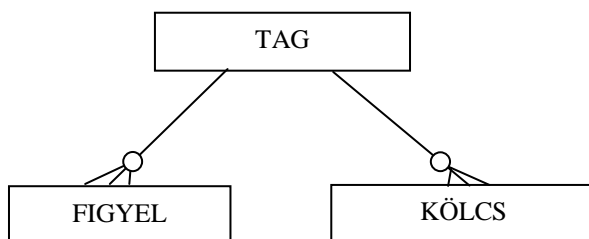
$$\text{PALIN}(\text{CRT}(\)) = \text{true}$$

$$\text{PALIN}(\text{APPEND}(s, x)) = (\text{LGTH}(s) == 0) \text{ or } (\text{PALIN}(\text{TAIL}(s)) \text{ and } (\text{HEAD}(s) == x))$$

8. Legyen egy bankjegykiadó automata (ATM), amelynek egyetlen funkciója van; az ügyfélnek pénzt ad ki. Az automata on-line kapcsolatban áll a bankkal, amelyen keresztül ellenőrzi az ügyfelet és a fedezetet, valamint értesíti a bankot a tranzakcióról. Rajzoljon context diagramot ! (4 pont)



9. A könyvtári tagokkal (TAG) kapcsolatosan tároljuk minden egyes kölcsönzési (KÖLCS), valamint a tagnak kéredelem esetén (SMS-ben, levélben, e-mailben) küldött figyelmeztetés (FIGYEL) adatát. A kölcsönzésre vonatkozó adatot a könyv visszavétele után 60 nappal töröljük, de a figyelmeztetési adatokat megőrizzük az ügyfél törléséig. Egészítse ki az alábbi ER diagramot ! (2 pont)



10. Az UML Superstructure dokumentációban az egyes fejezeteken belül előforduló UML osztályok abc sorrendben szerepelnek. Egy fejezeten belül milyen az elemek kohéziója ? (2 pont)

Logikai

11. Mi a “fan-in”? (2 pont)

- egy adott modult (objektumot) használó más modulok (objektumok) száma
- egy adott modul (objektum) döntési hatáskörébe tartozó modulok (objektumok) száma
- egy adott modul (objektum) által használt modulok (objektumok) száma
- egy adott modul (objektum) vezérlési hatáskörébe tartozó modulok (objektumok) száma

12. A szekvencia és az iteráció mellett van egy harmadik kompozíciós elv.

Mi ez az elv ? (1 pont) **szelekció**

A C nyelvben melyik ADATstruktúra implementálja ? (2 pont) **UNION**

13. Legyen egy ügyfél adattár, amelybe a felvesz processz vesz fel ügyfelet. A processz bemenete egy ügyfél adata (ügyféladat). Ha az ügyfél már szerepel az adattárban, akkor nincs teendő. Ha az ügyfél még nincs az adattárban, a processz azt beleírja. Rajzoljon DFD részletet a fentiek specifikálására ! (3 pont)



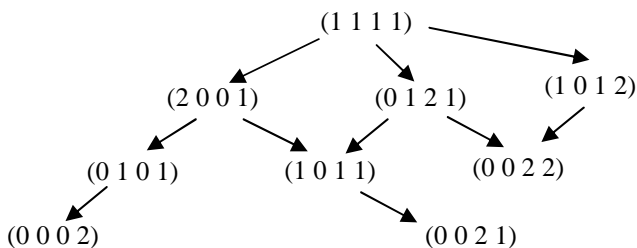
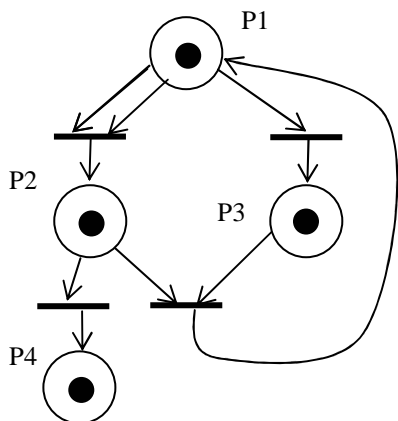
Minimálisan elérendő pontszám: 16/40

Vizsgára továbbvitt pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{4})$

ZH FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2007. január 9. 15:00

1. Rajzolja fel az alábbi Petri háló elérhetőségi gráfját (reachability tree) ! (3 pont)



2. Milyen típusú kohéziójú a C nyelv scanf () függvénye ? (2 pont)

- | | |
|--|--|
| <input type="checkbox"/> kommunikációs | <input type="checkbox"/> funkcionális |
| <input type="checkbox"/> logikai | <input checked="" type="checkbox"/> procedurális |

helyes válasz 2 pont, minden rossz -1 pont, de ≥ 0

Nevezzen meg legalább egy olyan kohéziós típust, amely a fentiek között nem szerepel ! (1 pont)

temporális vagy **esetleges** vagy **szekvenciális**

első helyes 1 pont, minden további helyes 0.5 pont, rossz -1 pont, de total ≥ 0

3. Írja le az alább definiált adatstruktúrát DTD-ben ! Legyen a, b, c és d parsed code data. (3 pont)

$$x = a + \{b + [a | c]\}$$

```
<!DOCTYPE x [
  <!ELEMENT x (a,(b,(a|c))*)>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
```

4. A kockázat tervezése során milyen stratégiákat választunk ? (3 pont)

A kockázat valószínűségének csökkentés

A bekövetkező káros hatások minimalizálása

A nem kívánatos eseményt követően felmerülő tevékenységek tervezése

5. Jellemezzünk egy stringet az alábbi műveletekkel! Adja meg az OCN műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két karakter összehasonlítására szolgáló műveleteket, és az integereken értelmezett összeadást és kivonást. (4 pont)

ADD(s,x) az s string végére rakja az x karaktert.

LGTH(s) az s string karaktereinek számát adja.

NEW() új (üres) stringet hoz létre.

OCN(s,x) az s stringben található x karakterek előfordulásának számát adja.

TAIL(s) az s string legrégebbi karakterének levágása után maradó stringet adja.

$OCN(NEW(), x) = 0$

$OCN(ADD(s, x), y) == \begin{cases} \text{if } (x == y) & OCN(s, y) + 1 \\ \text{else} & OCN(s, y) \end{cases}$

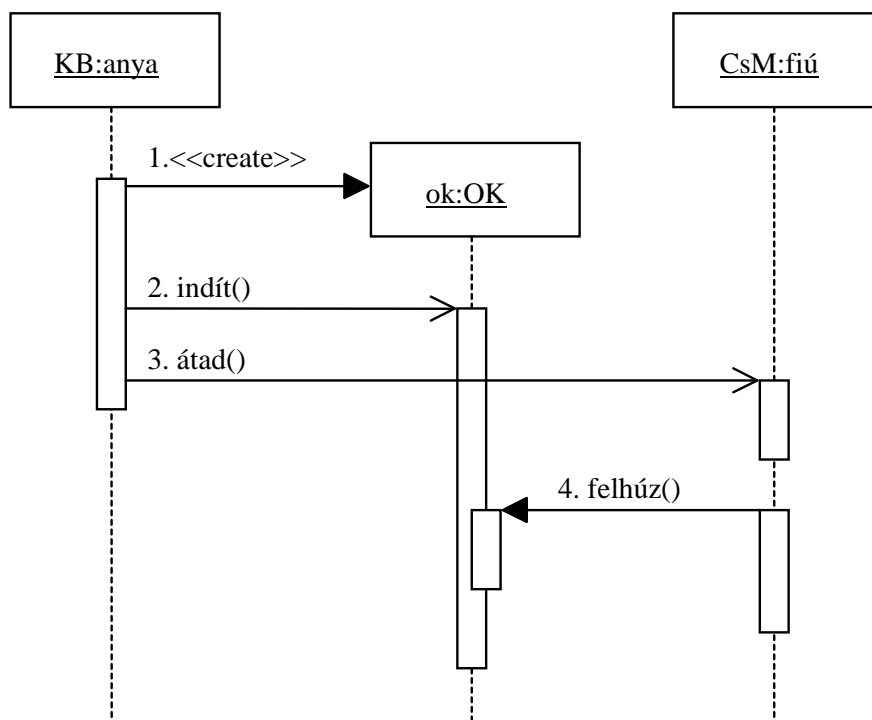
-1 pont, ha hiányzik

2 pont

2 pont

6. Készítsen UML szekvencia diagramot az alábbi történet alapján! (4 pont)

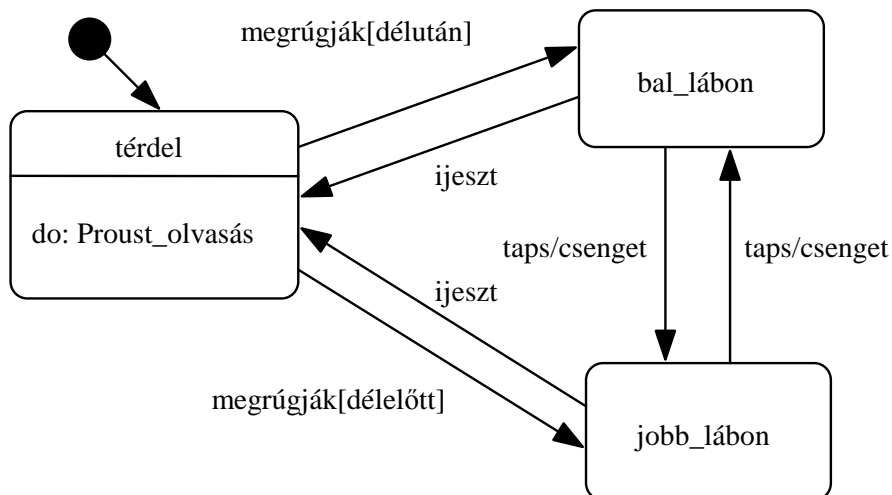
A híres mérnök, Kasza Blanka készít egy oszcillokulminátort, beindítja, majd átadja fiának, Csintalan Maximnak. A fiú másnap felhúzza a még mindig járó gépet.



7. Készítsen UML állapotdiagramot (state chart) az alábbi történet alapján! (4 pont)

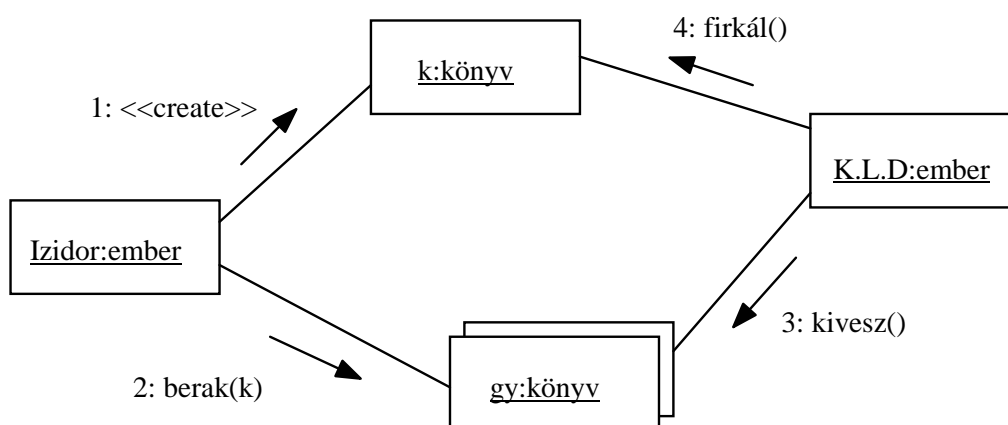
Harún ar-Rasíd kap Nagy Károlytól egy Heribert Illig nevű, 300 éves koboldot. A kobold alapból térdel. Ha ekkor megrúgják, attól függően, hogy délelőtt vagy délután van, a jobb illetve a bal lábára áll. Ha bármelyik lábán áll, és tapsot hall, akkor csenget egyet, és átáll a másik lábára.

Ha bármelyik lábán áll, és egy nullával ráijesztenek, térdre kényszerül. Mikor térdel, folyamatosan Proustot olvas.



8. Készítsen UML kollaborációs diagramot az alábbi történet alapján ! (4 pont)

Izidor ír egy könyvet, majd berakja a könyvgyűjteményébe. Éjjel K. Armandó Dzsoki betör hozzá, és pont ezt a könyvet lopja el a gyűjteményből, majd később bele is firkál.



9. Mi a lényege a JSD-ben az entitás-struktúra lépésnek (entity-structure step) ? (2 pont)

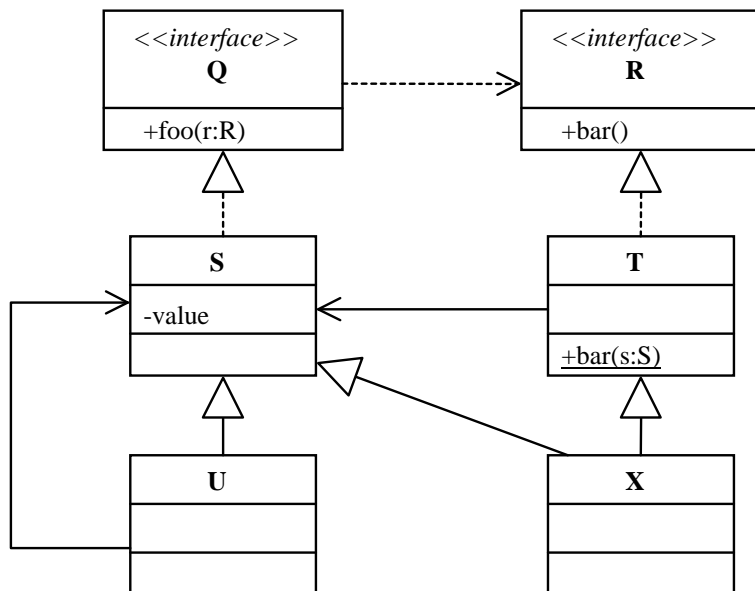
Az entitások *élettörténetének* specifikálása Jackson ábrával.....

Mi az előző és a következő lépés? (2 pont)

Előző: **entity-action**

Következő: **initial model**

10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)

- [B] X helyettesíthető U-val, mert mindketten megvalósítják a Q interfészt.
- [C] S foo(r:R) metódusa kaphat paraméterül X-et, mert X az S leszármazottja.
- [E] X módosíthatja egy S value attribútumát, mert az attribútum statikus.
- [B] S nem hívhatja meg egy T bar() metódusát, mert az asszociáció T-ből S-be irányul.
- [B] S foo(r:R) metódusa nem módosíthatja a value attribútumot, mert különböző a láthatóságuk.
- [B] U módosíthatja a vele asszociációban levő S objektum value attribútumát, mert S az U ősztyála.
- [B] T nem hívhatja meg S foo(r:R) metódusát, mert az R interfész nem függ Q-tól.
- [B] T bar(s:S) metódusa nem hívható meg egy X objektummal, mert X-nek nincs statikus attribútuma.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Minimálisan elérendő pontszám: 16/40

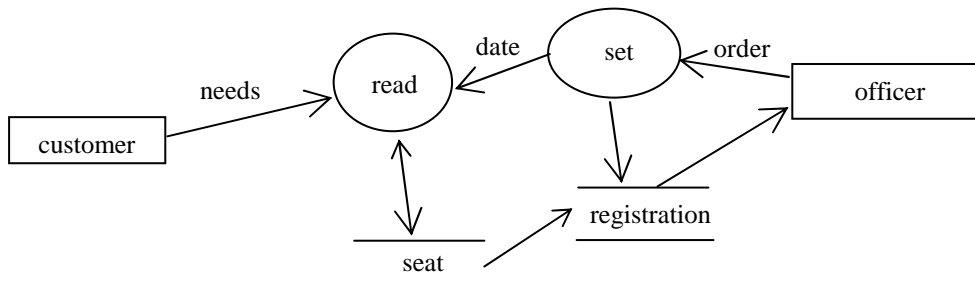
Vizsgára továbbvitt pontszám: $\text{int}(\frac{[\text{szerzett pont}] - 16}{8})$

ZH FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2006. december 8. 18:00

1. Milyen szintaktikai hibákat talál az alábbi adatfolyam ábrán (3 pont)?



- ha DFD, akkor nem lehetnek terminátorok, ha CD, akkor nem állhat több gombóc és store.
- a read processznek nincs kimenete, ezért a seat-ből a read-be mutató nyíl inkorrekt.
- registration és officer nem köthető.
- seat and registration nem köthető.

2. Mit jelent az objektum változókon bevezetett **tipizálás** és **kötés** fogalma? Milyen összefüggés van közöttük? (3 pont)

tipizálás: milyen objektumok tehetők a változóba (statikus – fordításkor, dinamikus – futáskor definit)(1 pont)

kötés: kihez kapcsolódnak a műveletek (statikus – a változóhoz, dinamikus – az értékhez)(1 pont)

kapcsolat (1 pont):

	<i>stat. tipizálás</i>	<i>din. tipizálás</i>
<i>stat.kötés</i>	nem OO	értelmetlen
<i>din. kötés</i>	normál OO (Java, C++)	Smalltalk, Excel cella

3. Írja le az alább definiált adatstruktúrát DTD-ben! Legyen a, b és c parsed code data. (3 pont)

xxx = [a | -] + b + {c}

```

<!DOCTYPE xxx [
  <!ELEMENT xxx (a?,b,c*)>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
  
```

4. Sorolja fel, hogy a minőség milyen nézőpontokból (views of quality) vizsgálható! Jelölje be, hogy a szoftver technológia jellemzően melyiket tekinti alapvetőnek! (4 pont)

- | | |
|--|---|
| <input type="checkbox"/> Transcendental (transzcendentális) | <input type="checkbox"/> User (felhasználói)..... |
| <input checked="" type="checkbox"/> Manufacturing (gyártási)..... | <input type="checkbox"/> Product (termék)..... |
| <input type="checkbox"/> Value-for-money (a pénzért kapható érték) | <input type="checkbox"/> |

minden helyes válasz 0,5 pont, a szoftver technológia 1,5 pont

5. Jellemezzünk egy (x,y) párokból álló listát – ahol x egy kulcs, y egy 0-nál nagyobb egész – az alábbi műveletekkel! Adja meg az algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket. (4 pont)

- NEW()** új (üres) listát hoz létre.
ADD(l,(x,y)) az l listához kapcsolja az x,y párt, ha x nem szerepelt a listán. Ha a listán már van x, akkor a hozzá tartozó y-t az új y-nal helyettesíti, ha az nagyobb a listán szereplőnél.
VALUE(l,x) megadja az l listán az x kulcshoz tartozó y-t. Ha a listán a megadott x nem szerepel, akkor az eredmény 0.
MIN(l) a listában szereplő legkisebb y. (üres lista esetén nulla)

VALUE(NEW()) = 0

-0.5 pont, ha hiányzik

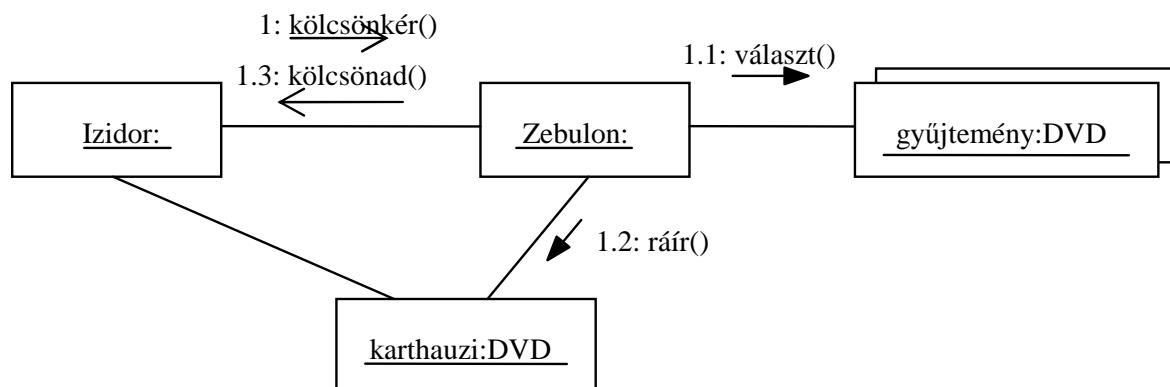
MIN(NEW()) = 0

-0.5 pont, ha hiányzik

VALUE(ADD(l, (x,y),z)) = if (x == z && VALUE(l,x) < y) y else VALUE(l, z) 2 pont

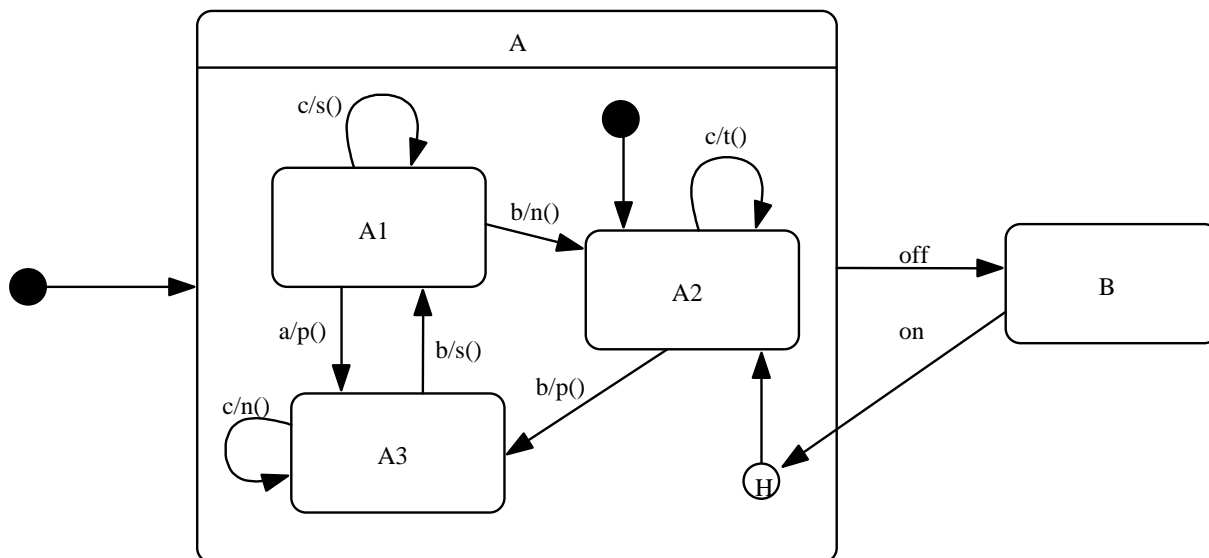
MIN(ADD(l, (x,y))) = if (MIN(l) == 0 || VALUE(l,x) == 0 && MIN(l) > y) y else MIN(l) 2 pont

6. Izidor kölcsönkéri Zebulon egyik DVD-jét, amin karthauzi szerzetesek olvassák fel a rend nagy szónokainak műveit. Zebulon kiválasztja gyűjteményéből a kért DVD-t, amire ráírja, hogy az övé, majd másnap odaadja Izidornak. Rajzoljon UML kollaborációs diagramot a fenti történetre! (4 pont)



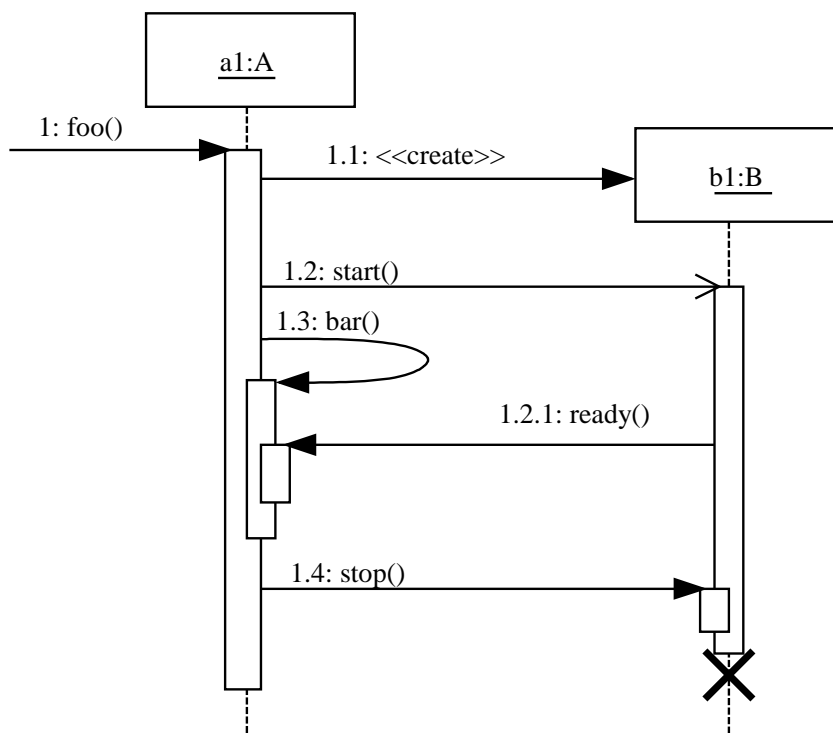
7. Legyen egy objektumunk, amelynek két fő állapota (**A,B**) van. Az **A** állapotban a táblázatban megadott állapotgép működik. Az **A** állapotból **B**-be az 'off' esemény hatására kerül. Visszatérni az 'on'-ra fog, és ekkor ott folytatja, ahol a kilépéskor abbahagyta. Kezdetben az **A** állapot aktivizálódik, az **A**-n belül pedig az **A2**. Rajzolja meg az objektum UML state-chart-ját! (4 pont)

	a	b	c
A1	A3/p()	A2/n()	A1/s()
A2	-	A3/p()	A2/t()
A3	-	A1/s()	A3/n()



8. Rajzoljon UML szekvencia diagramot ! (4 pont)

Az **A** osztály egy **a1** példányra a **foo()** metódus hatására létrehoz egy **B** osztályba tartozó **b1** objektumot, majd aszinkron módon meghívja annak **start()** metódusát. Ezután **a1** meghívja saját **bar()** metódusát. A **bar()** indulása után **b1** **a1**-nek szinkron módon a **ready()** üzenetet küldi. Ekkor a **bar()** véget ér, és **a1** meghívja **b1** **stop()** metódusát, mire **b1** megsemmisül.



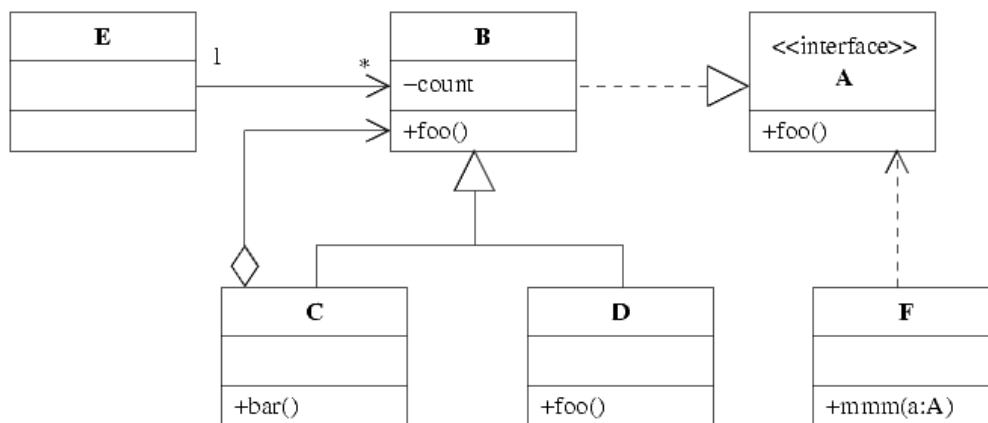
9. A három-rétegű szoftver architektúrában milyen logikai rétegeket definiáltunk ? (3 pont)

GUI – felhasználói felület

BOM – alkalmazás modell

DB – adatbázis.....

10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

[B] D osztályból meghívhatjuk a C bar () metódusát, mert a C osztályú objektumnak lehet komponense D osztályú objektum.

[B] D osztály nem definiálhatja felül a foo () metódust, mert B nem absztrakt.

[B] F osztályú objektum mmm (a : A) metódusából meghívhatjuk a C bar () metódusát, mert C lehet az mmm (a : A) paramétere.

[B] C osztály helyettesíthető D-vel, mert mindketten megvalósítják az A interfészt.

[D] F osztály mmm (a : A) metódusa paraméterül kaphat D osztályú objektumot, mert utóbbi megvalósítja az A interfészt.

[A] E nem ismeri az A interfészt, ezért E nem hívhatja meg B foo () metódusát.

[E] E osztály ismeri az A interfészt, mert B helyettesíthető E-vel.

[D] E osztályú objektum nem hívhatja meg C osztályú objektum bar () metódusát, mert csak a B osztályt ismeri.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Minimálisan elérendő pontszám: 16/40

Vizsgára továbbvitt pontszám: $\text{int}(\lceil \text{[szerzett pont]} - 16 \rceil / 4)$

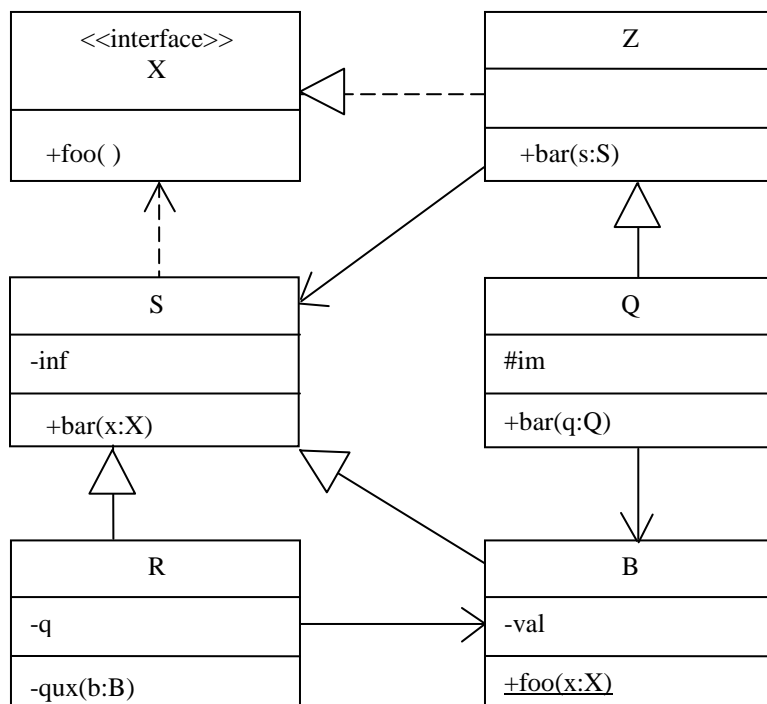
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2008. június 10. 16:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



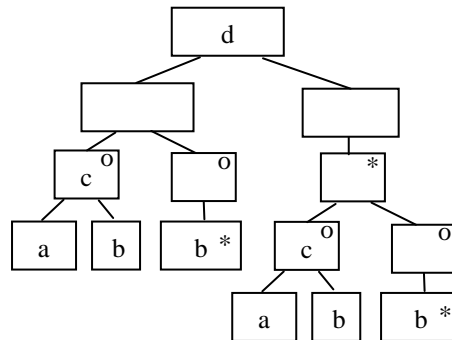
- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [E] R helyettesíthető S-sel, mert S az R leszármazottja
- [E] R helyettesíthető B-vel, mert mindketten megvalósítják az X interfészt
- [A] R átadható paraméterül Q bar(s:S) metódusának, mert Q és S interfésze megegyezik.
- [D] B foo(x:X) metódusa nem látja a val attribútum értékét, mert az attribútum nem statikus.
- [A] Q meghívhatja S bar(x:X) metódusát, mert mindketten megvalósítják az X interfészt.
- [B] B interfésze tartalmaz qux(b:B) metódust, mert B-nek van R-rel közös őse.
- [A] Q bar(s:S) metódusa nem módosíthatja az im attribútumot, mert az attribútum privát.
- [B] B meghívhatja R qux(b:B) metódusát, mert a metódus paramétere B osztályú.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Rajzolja fel az alábbi DTD-vel specifikált adatszerkezetet Jackson ábrával ! (4 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE d [
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (a, b)>
  <!ELEMENT d (c|b*)+>
]>
```



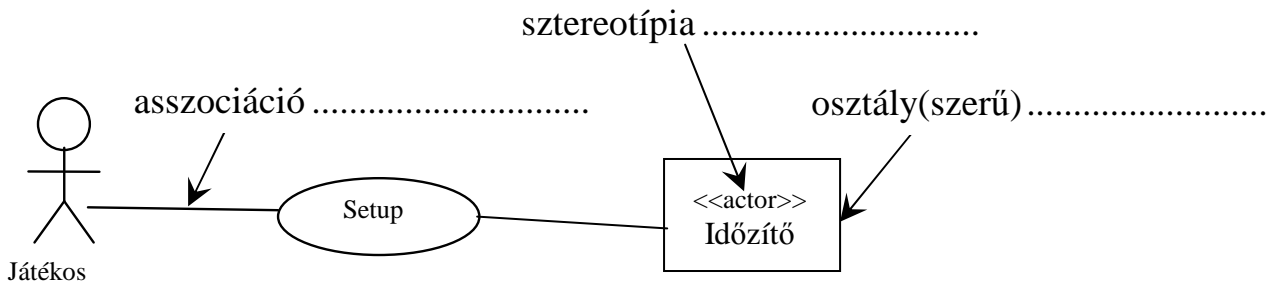
3. Mi a lényege a JSD-ben az “entitás-akció” lépésnek (entity-action step) ? (2 pont)

- Az entitások akcióinak ábrázolása Jackson-diagramon
- Entitás-relációs (akciós) diagram megrajzolása
- Az akciókban érintett entitásokra ER diagram rajzolása
- Az entitások processzekre fordításának akciója
- Az entitások és események (akciók) azonosítása
- Az entitás-élettörténet (ELH) azonosítása

Mi az **előző** lépés? (1 pont)
nincs

Mi a **következő** lépés? (1 pont)
entitás-struktúra...

4. Az alábbi ábrán három UML2 modell elemet megjelöltünk. Adja meg elemenként, hogy az melyik UML2 meta-modell elem példánya ! (3 pont)



5. Egy objektum metódusa procedurálisan kohézív, ha a metódus ... (1 pont)

- egy nagyon fontos procedúrát hajt végre
- kódjában további metódus és procedúra hívások vannak
- különböző adattípusra (osztályra) hajtja végre ugyanazt a műveletet
- nevében legalább két ige (pl. read_and_validate_input) áll
- nem tartalmaz adat-deklarációkat, csak kódot
- egy adattípusra (osztályra) különböző műveleteket (procedúrákat) hajt végre

6. Hogyan értelmezzük az UML2-ben a szekvenciális konkurenciát ? (2 pont) A magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

callers must coordinate outside the object so that only one flow is in the object at a time.

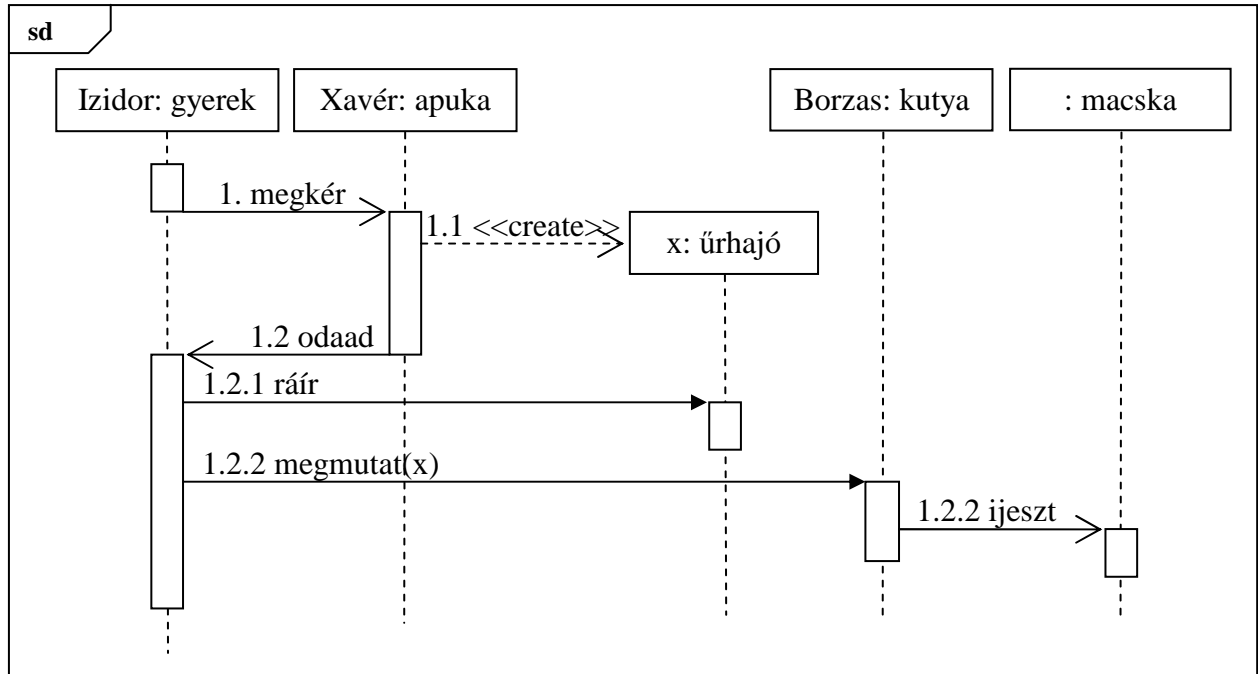
[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

Sorolja fel az UML által definiált egyéb konkurencia szemantikákat ! (2 pont)

guarded, concurrent

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

7. Rajzoljon UML2 **szekvenciadiagramot** ! Az üzeneteket hierarchikus számozással lássa el ! (7 pont)
 Izidor megkéri apukáját, Xavért, hogy készítse neki egy úrhajót. Mikor Xavér kész van, az úrhajót odaadja Izidornak, majd elsiet, mert sok a dolga. Izidor az úrhajóra ráírja a nevét, majd megmutatja a legjobb barátjának, Borzas kutyának, aki a mutogatás közben megijeszt egy macskát.



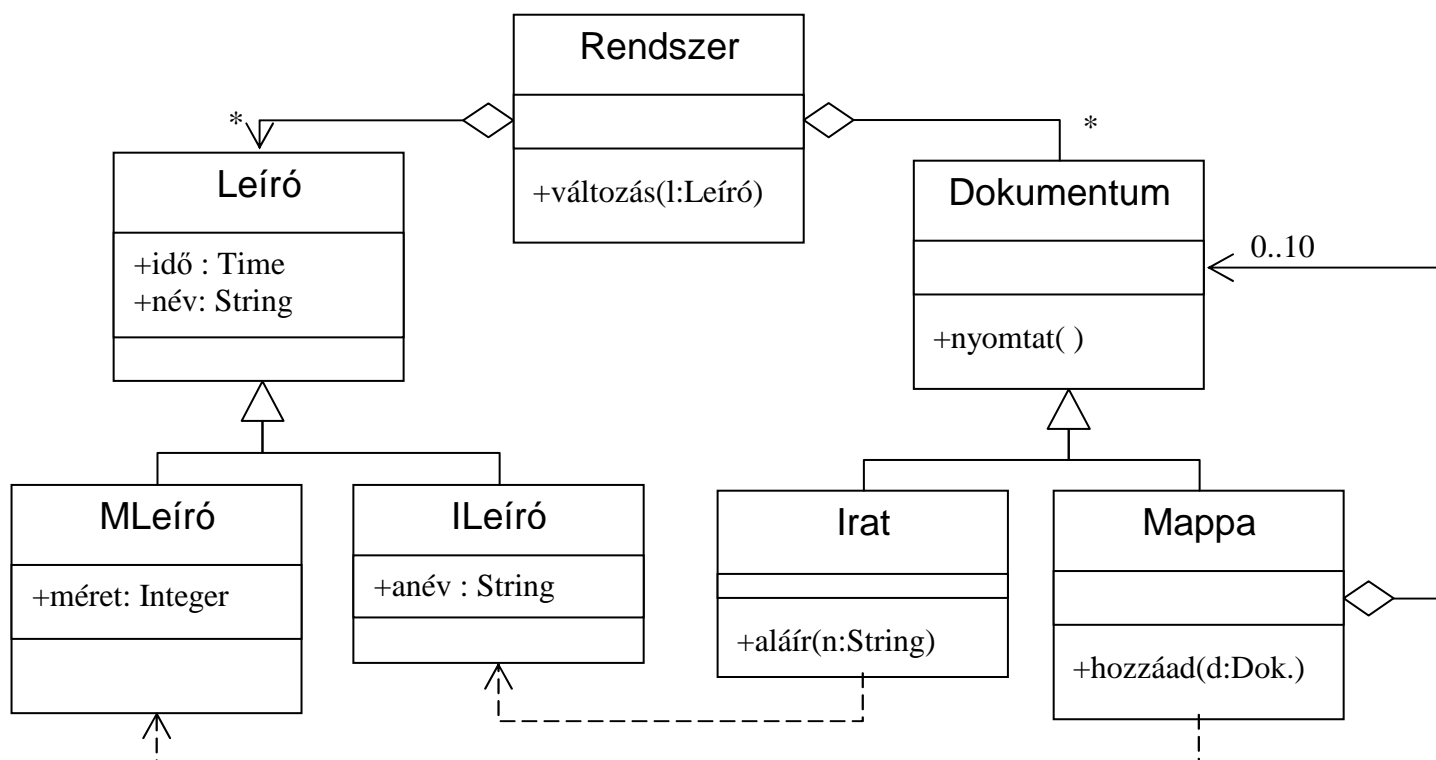
8 Jellemezzünk egy stringet az alábbi műveletekkel ! Adja meg az OCC műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két karakter összehasonlítására szolgáló műveleteket, és az integereken értelmezett alapműveleteket. (4 pont)

- ADD(s,x)** az s string végére rakja az x karaktert.
- TAIL(s, n)** az s string legrégebbi n karakterének levágása után maradó stringet adja.
- LGTH(s)** az s string karaktereinek számát adja.
- CRT()** új (üres) stringet hoz létre.
- OCC(s,x)** az s stringben található x karakterek előfordulásának számát adja.

$OCC(CRT(), x) = 0$		-1 pont, ha hiányzik
$OCC(ADD(s, x), y) ==$	if (x == y) OCC(s, y) + 1	2 pont
	else OCC(s, y)	2 pont

9. Készítsen UML2 **osztálydiagramot (class diagram)** az alábbi leírás alapján ! (9 pont)

Egy dokumentumkezelő rendszerben heterogén kollekciónként névvel ellátott dokumentumokat tárolunk. A dokumentum lehet irat vagy mappa. A mappa további dokumentumokat tárolhat, de maximum 10-et. A rendszernek képesnek kell lennie arra, hogy később további dokumentumfajtákkal (fotó, hangszalag stb. bővítsük). A dokumentumokat ki tudjuk nyomtatni (*nyomtat*), az iratokat alá lehet írni (*aláír*), a mappákba újabb dokumentum helyezhető (*hozzáad*). A rendszer nyilvántartja a dokumentum-módosításokat is. Minden dokumentum a módosításról értesíti a rendszert (*változás* metódus), és átadja a módosítás adatait leíró objektumot. Ebben a módosuló dokumentum megadja a nevét és a módosítás idejét, de ezen kívül a dokumentum típusától függő egyedi adatok is szerepelhetnek (irat esetén az aláíró neve, mappa esetén a mappa mérete).



10. Felsoroltunk szoftverrel kapcsolatos "hibákat". A hibák mellett jelölje be, hogy az melyik kategóriába tartozik! (4 pont) Rossz válasz esetén pontot veszít !

	bug	failure	error	fault
Hiányzik a "synchronized" kulcsszó	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A ciklusfeltétel hibás	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Elmaradt a kritikus kódok felülvizsgálata	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Alábecsültük a példányosítás erőforrás-igényét	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sok objektumnál nagyon lassan kapjuk az eredményt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Az első futásnál hibás eredményt kapunk	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

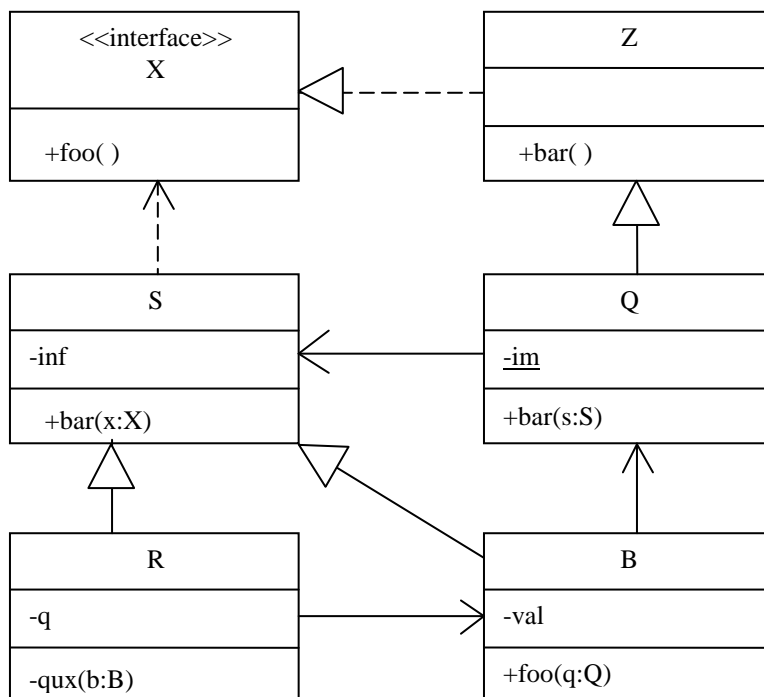
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2008. május 27. 16:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [E]** **S** helyettesíthető **Q**-val, mert **Q** az **S** leszármazottja
- [A]** **S** helyettesíthető **B**-vel, mert **B** megvalósítja az **X** interfészt
- [A]** **R** átadható paraméterül **Q** **bar(s:S)** metódusának, mert **Q** és **S** interfésze megegyezik.
- [B]** **B** **foo(q:Q)** metódusa nem látja saját **val** attribútumának értékét, mert az attribútum privát.
- [A]** **Q** meghívhatja **S** **bar(x:X)** metódusát, mert mindketten megvalósítják az **X** interfészt.
- [E]** **B** interfésze tartalmazza **bar(s:S)** metódust, mert a metódus statikus.
- [A]** **Q** **bar()** metódusa nem módosíthatja az **im** attribútumot, ezért az attribútum konstans.
- [E]** **B**-nek nincs **bar(x:X)** metódusa, ezért nem függ az **X** interfésztől.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Az alábbi XML leírás jól formált és érvényes-e ? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a,c*)|b)>
  <!ATTLIST a d CDATA #REQUIRED>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <a d></a>
  <c>zh</c>
</x>
```

- IGEN
 NEM

Hiba ?: A paraméterből hiányzik az egyenlőségjel és az érték

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van c elem ! (3 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

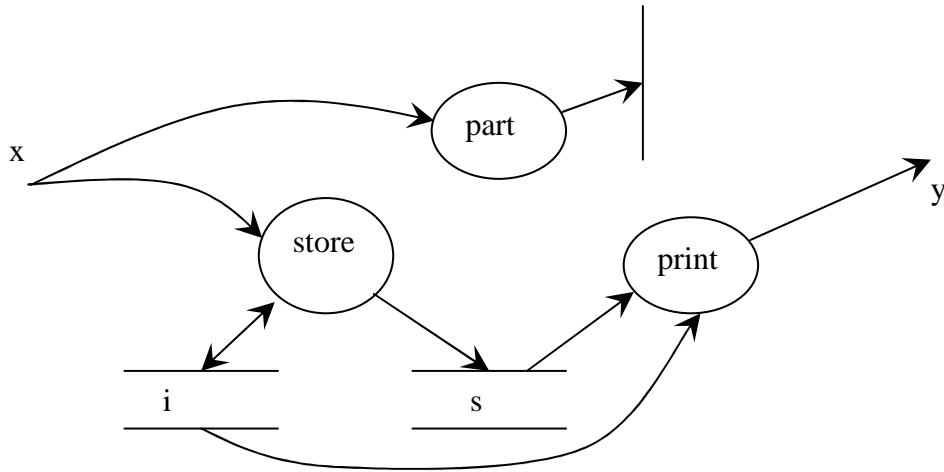
```
<x>
  <a d="x">aaa</a>
  <c>ccc</c>
</x>
```

3. Egy program (amelynek bemenete x, kimenete y) működését az alábbi állapottábla írja le:

	e1	e2	e3
A1	A1/-	A1/-	A2/s[i++]=x
A2	A3/-	A3/-	A2/s[i++]=x
A3	A3/s[i++]=x	A3/-	A1/y=print(s,i)

ahol e1, e2, e3 események, amelyek a part(x) függvény lehetséges értékei, s egy 1000 elemű x típusú értékek tárolására szolgáló tömb, i integer index.

Rajzolja fel a program **adatfolyamábráját** ! (7 pont)



Minden adat, process, store és a control 0,5 pontot ér

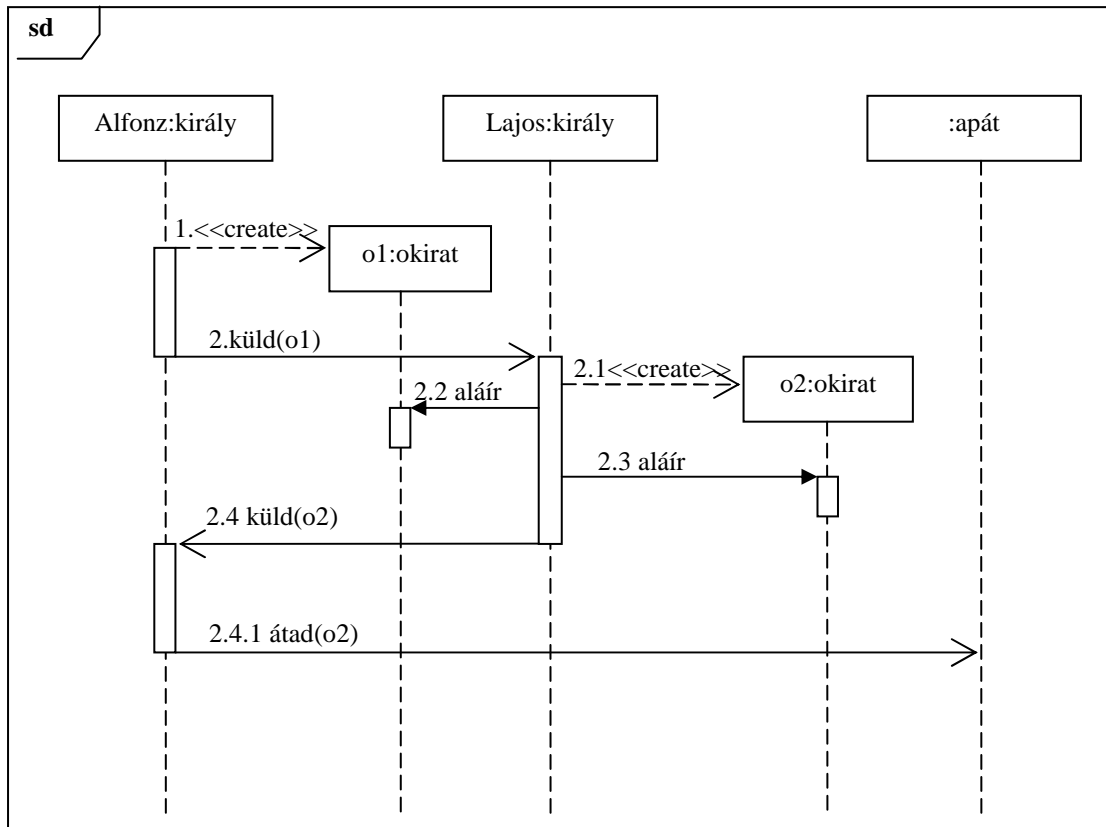
4. Melyek a kockázatkezelés folyamatai (risk management process) ? (4 pont)

- Kockázatok azonosítása (identification)*
- Kockázatok elemzése (analysis)*
- Kockázatok tervezése (planning)*
- Kockázat felügyelet (monitoring)*

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Rajzoljon UML 2 szekvenciadiagramot ! (7 pont)

X. (Bölcs) Alfonz, Kasztília királya szövetséget kíván kötni IX. (Szent) Lajossal, Franciaország királyával. Ezért elkészít egy okiratot, amit el is küld a francia uralkodónak. Lajos, amint megkapja a dokumentumot, készített egy másolatot, majd mindkettőt kézjegyével látja el, és a másolatot visszaküldi Alfonzknak, aki az okiratot a toledói apátnak adja át megőrzésre.



6. Definiálja a technológia (engineering, mérnökség) fogalmát öt komponensének megadásával (5 pont) !

gyakorlati problémák megoldására szolgáló **dolgok**

a **társadalom** által szabályozott **tudományos** alapokon álló

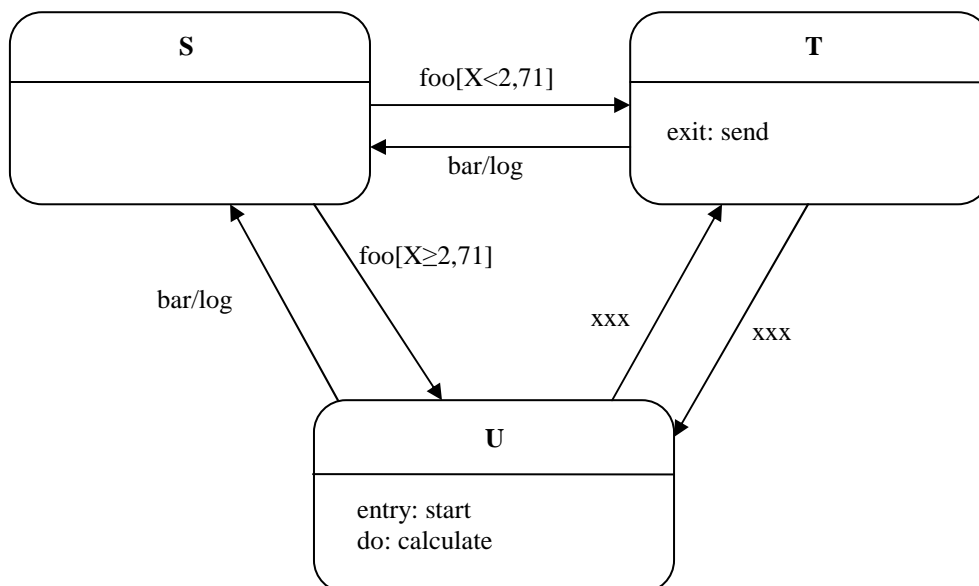
gazdaságos

előállításának mikéntje

mindegyik komponens 1 pont.

7. Egészítse ki az alábbi UML 2 állapotdiagramot (state chart) a következő leírás alapján! (7 pont)

Egy objektum három fő állapottal (**S**, **T**, **U**) rendelkezik. A kezdőállapot az **S**. Ha **S**-ben **foo** esemény éri, akkor attól függően, hogy **x** értéke kisebb, mint 2,71 vagy sem, rendre a **T** vagy az **U** állapotba kerül. Mindkét állapot a **bar** és az **xxx** események hatására hagyható el. Előbbi esemény esetén visszatér **S**-be, utóbbinál pedig **T**-ből **U**-ba, **U**-ból **T**-be kerül. **U**-ba lépéskor mindig lefut a **start** metódus, míg **T**-t elhagyva a **send**. A **bar** eseményre történő állapotváltás során a **log** metódus hívódik meg. Az **U** állapotban tartózkodás közben a **calculate** metódus fut.



8. Jellemezzünk egy stringet az alábbi műveletekkel ! A string karakterei előről 1-gyel kezdődően számozottak. Adja meg a **PAR** műveletre vonatkozó algebrai axiómákat ! Az axiómák felírásakor használhatja a két egész összehasonlítására és két karakter összehasonlítására szolgáló műveleteket. (5 pont)

- CRT()** új (üres) stringet hoz létre.
- SET(s,x)** az s string elejére – az 1. számú helyre – rakja az x karaktert.
- PAR(s)** igaz, ha bárhol a stringben egymás mellett legalább két egyforma karakter áll.
- IN(s,i)** eredményül adja az s string i-ik karakterét. Ha i nagyobb mint a string hossza, akkor az eredmény értelmetlen (nem definiált).
- LGTH(s)** az s string karaktereinek számát adja.

$PAR(CRT()) = false$ 0.5 pont

$PAR(SET(s, x)) == (LGTH(s) != 0) \ \&\&$ 1,5 pont

$(x == IN(s, 1) \ ||$ 1,5 pont

$PAR(s))$ 1,5 pont

ez utóbbival egyenértékű:

$if (LGTH(s) == 0) false \ else \ if (x == IN(s,1) \ || \ PAR(s)) \ true$

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

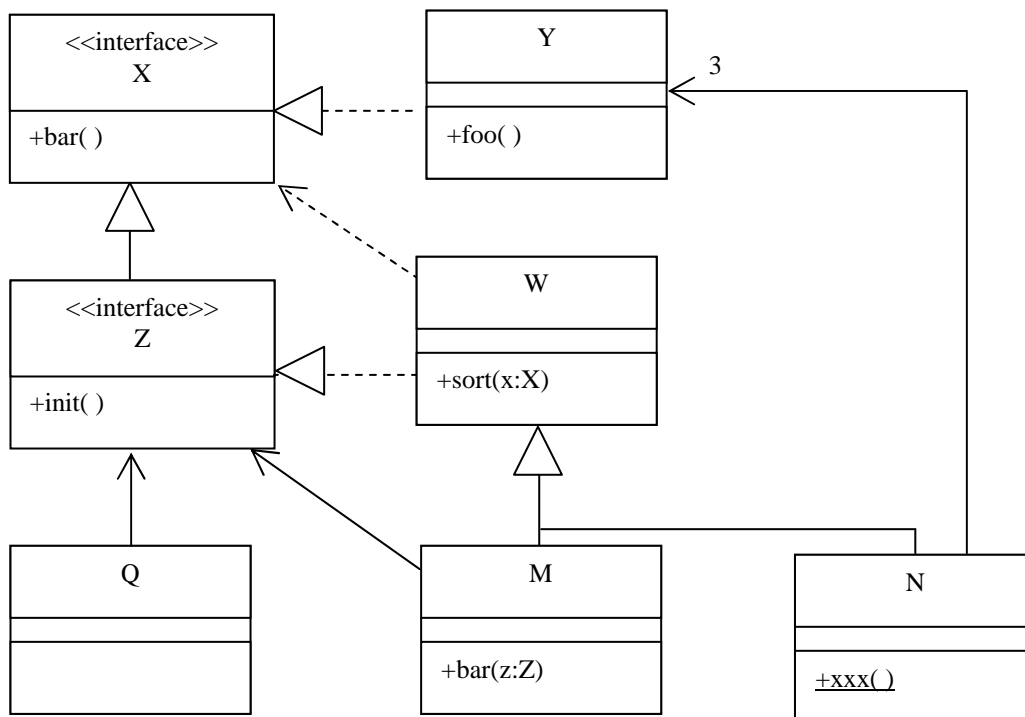
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2008. január 22. 14:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



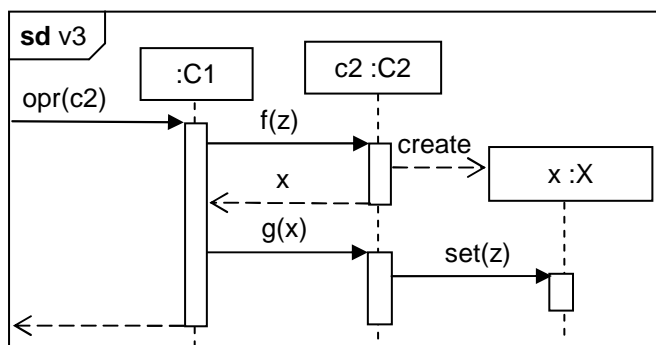
- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B]** Y helyettesíthető W-vel, mert mindketten megvalósítják az X interfészt.
- [C]** N meghívhatja Y foo() metódusát, mert N megvalósítja a Z interfészt.
- [C]** M bar(z : Z) metódusa kaphat paraméterül N objektumot, mert van közös ősük.
- [B]** W nem helyettesíthető M-mel, mert W-nek nincs bar(z : Z) szignatúrájú metódusa.
- [E]** Q helyettesíthető M-mel, mert mindkettő megvalósítja a Z interfészt.
- [B]** N xxx() metódusa meghívható a W osztály sort(x : X) metódusából, mert az N.xxx() statikus.
- [C]** W sort(x : X) metódusa meghívhatja egy paraméterül kapott Y objektum bar() metódusát, mert W-nek is van ugyanilyen szignatúrájú metódusa.
- [A]** M-nek és N-nek különböző az interfésze, mert N nem valósítja meg X-t.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Tételezze fel, hogy az alábbi UML2 szekvenciadiagramon szereplő objektumok osztályai között nincs más egyéb – a diagramból nem kiolvasható – kapcsolat (pl. öröklés) ! Mi a kapcsolat C2 és X között ? Válaszát egy, a magyar nyelv szabályainak megfelelő, olvasható MONDATtal indokolja ! Indoklás nélkül a választása nem érvényes. (4 pont)

- példányosítás (instantiation)
- asszociáció (association)
- kollaboráció (collaboration)
- függőség (dependency) C2 függ X-től
- függőség (dependency) X függ C2-től
- interakció (interaction)
- implementálás (implementation)



Indoklás:

C2 létrehoz, használ (de most nem kell emlékeznie)

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

3. Az oo világban szokásos típuskompatibilitást feltételezve jelölje be az IGAZ állításokat ! (4 pont)

- Egy objektum osztályában definiált statikus metódus dinamikusan kötődik egy dinamikusan (futási időben) létrehozott változóhoz.
- Egy statikusan (fordítási időben) létrehozott változóhoz csak statikusan kötődhet metódus.
- Egy objektum osztályában definiált statikus metódus statikusan kötődik egy statikusan létrehozott változóhoz.
- A dinamikusan létrehozott statikus típusú változóba csak a változó típusával pontosan egyező objektum tehető.
- Statikusan létrehozott statikus típusú változóba csak olyan objektum tehető, amelynek minden metódusa statikus.
- A statikusan létrehozott dinamikus típusú változóba helyezhető objektumokra nézve is kötelező a típuskompatibilitás.
- Ha egy változóhoz csak statikusan köthetők metódusok, akkor nem érvényes a típuskompatilitás.

4. A RUP a Use Case (használati eset) modellek milyen formáit alkalmazza ? (3 pont)

high level or expanded

essential (free of technology), real

primary, secondary, optional

Milyen UML2 metamodell elem az actor és a use case közötti “vonal” ? (1 pont)

asszociáció.....

5. A konfigurációs menedzsmentnek (CM) mik az alapvető folyamatai ? (4 pont)

storage conf. item

build mgmt

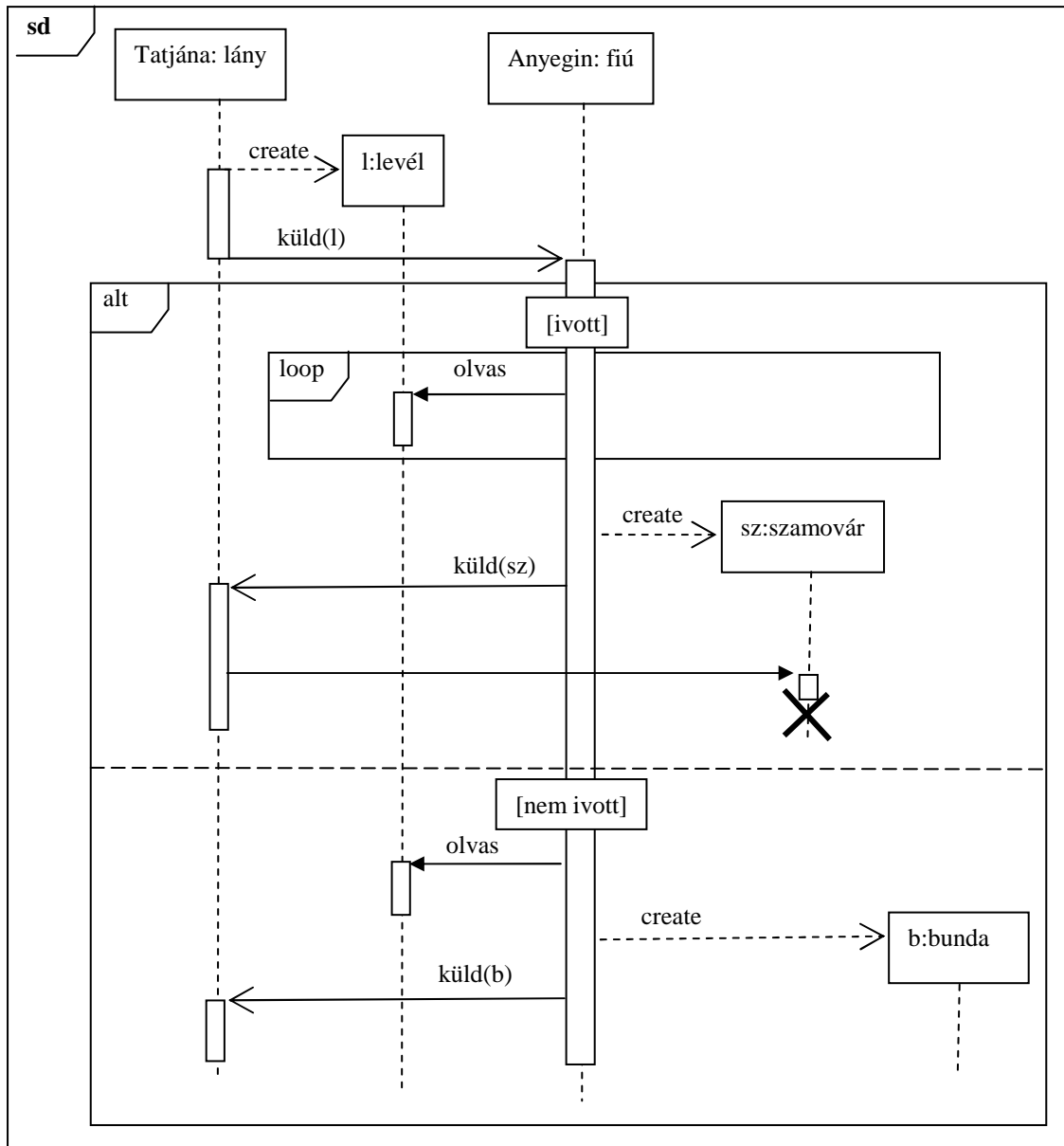
change mgmt

release mgmt.....

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Rajzoljon UML 2 szekvenciadiagramot ! (7 pont)

Tatjana levelet ír, majd elküldi Anyeginnek. Ha Anyegin ivott vodkát, akkor többször is elolvassa a levelet, majd készít egy gyönyörű, aranyozott számovárt, és elküldi Tatjánának, aki (mivel nem erre számított) megsemmisíti a számovárt. Ha azonban Anyegin nem ivott, akkor csak egyszer olvassa el a levelet, majd bundát csinál, és ezt a bundát küldi Tatjánának.

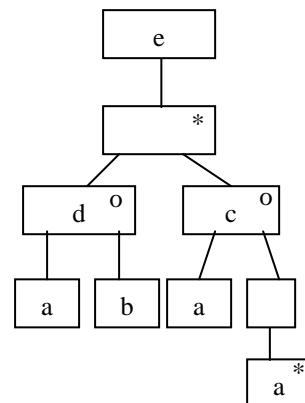


7. Rajzolja fel az alábbi DTD-vel specifikált adatszerkezetet Jackson ábrával ! (4 pont)

```

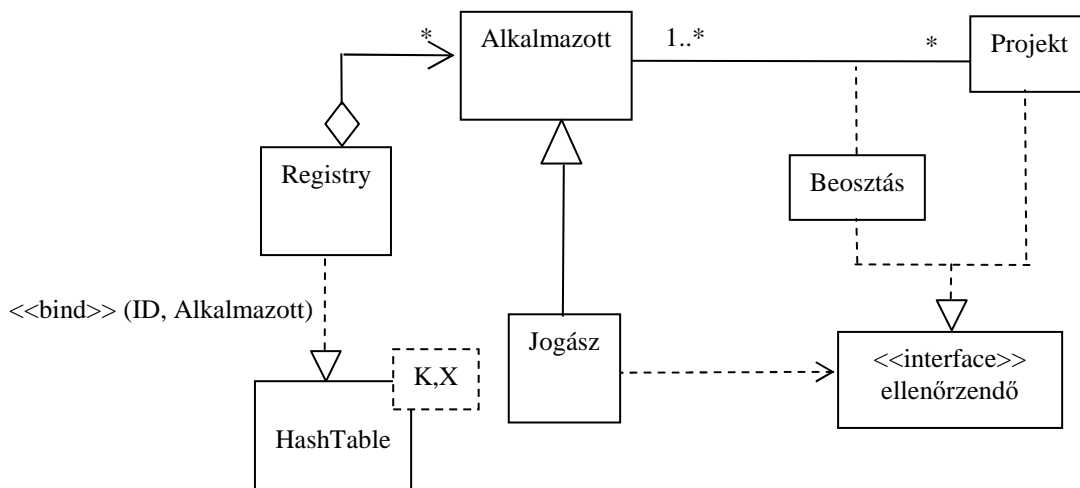
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE e [
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (a+)>
  <!ELEMENT d (a, b)>
  <!ELEMENT e (d|c)*>
]>

```



8. Rajzoljon UML 2 osztálydiagramot! A metódusokat és attribútumokat nem kell jelölje ! Csak a **vastagon** szedett classifier-ek szerepeljenek a diagramon ! (8 pont)

Egy cégnél nyilvántartják az **alkalmazottak** és a **projektek** adatait. Az alkalmazottak projektekhez vannak rendelve, minden projektnél különböző beosztásban. Minden projekten dolgozik legalább egy alkalmazott, de lehet olyan alkalmazott, aki éppen nincs projekthez rendelve. A **beosztás** határozza meg például, hogy mennyi bónuszt kap az alkalmazott az adott projekt sikere esetén. Mind a projekt, mind a beosztás jogilag **ellenőrzendő** (megvalósítják az **ellenőrzendő** interfészt). A **jogász** (aki persze a cég alkalmazottja is egyben) dolga, hogy az ellenőrzéseket elvégezze. A **registry**ben tárolják az alkalmazottak azonosítóinak és az alkalmazotti adatoknak az összerendelését. A registry az egyszerűség kedvéért a **HashTable** (K kulcs és X érték paraméterű) template osztályt példányosítja, ahol a kulcs az azonosító, az érték az alkalmazott adata.



9. Milyen minőségi jellemzőket tesztelünk a különböző FURPS típusú tesztekkel ? (5 pont)

Functionality, funkcionalitás	Usability, használhatóság.....
Reliability, megbízhatóság.....	Performance, teljesítőképesség
Supportability, támogatottság.....

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

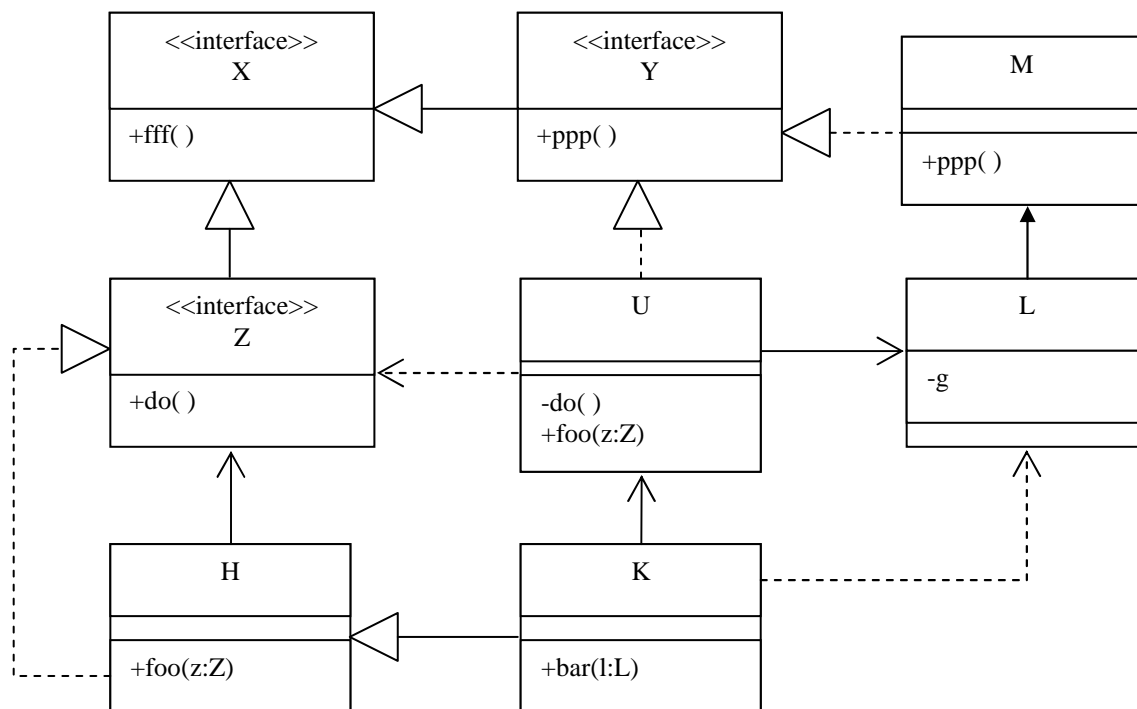
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2008. január 15. 14:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



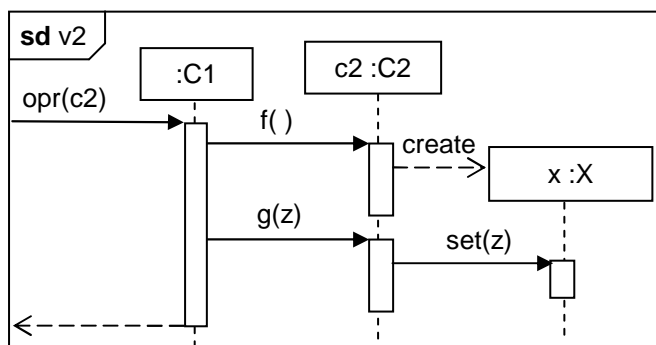
- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B]** H bárhol helyettesítheti U-t, mert mindketten megvalósítják az X interfészt.
- [B]** H foo(z:Z) metódusa meghívható egy U-val, mert U megvalósítja az Y interfészt.
- [A]** U nem hívhatja meg K bar(l:L) metódusát, mert K nem függ L-től.
- [A]** K implementálja a Z interfészt, ezért K meghívhatja U do() metódusát.
- [B]** K nem hozhat létre L objektumot, mert az L g attribútuma privát.
- [C]** U foo(z:Z) metódusa nem hívhatja meg egy paraméterül kapott H foo(z:Z) metódusát, mert U nem implementálja a Z interfészt.
- [E]** K bárhol helyettesíthető U-val, mert K az U leszármazottja.
- [A]** L nem ismeri az Y interfészt, ezért L nem hívhatja meg M ppp() metódusát.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Tételezze fel, hogy az alábbi UML2 szekvenciadiagramon szereplő objektumok osztályai között nincs más egyéb – a diagramból nem kiolvasható – kapcsolat (pl. öröklés) ! Mi a kapcsolat C2 és X között ? Válaszát egy, a magyar nyelv szabályainak megfelelő, olvasható MONDATtal indokolja ! Indoklás nélkül a választása nem érvényes. (4 pont)

- példányosítás (instantiation)
- asszociáció (association)
- kollaboráció (collaboration)
- függőség (dependency) C2 függ X-től
- függőség (dependency) X függ C2-től
- interakció (interaction)
- implementálás (implementation)



Indoklás:

C2-nak "emlékeznie kell" X-re

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

3. Mi a lényege a JSD-ben a "kezdeti modell" lépésnek (initial model step) ? (2 pont)

- A processzek Jackson-diagramjának felvétele
- ELH (entitás-élettörténet) diagram megrajzolása
- Az entitás-élettörténetek alapján processz-hálózat felvétele
- Az entitások halmazának kibővítése processzekkel
- Az entitások és események kezdeti leírása
- A prototípus entitásainak adatbázisstervei

Mi az **előző** lépés? (1 pont)
entity-structure
 Mi a **következő** lépés? (1 pont)
function

4. Jelölje meg, hogy a megadott rajzjelek minek az "ikon"-jai az UML2-ben ! (4 pont)

felsorolás (enumeration)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
termék (artifact)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
komponens (component)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
interfész (interface)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
diszk, fájl (disc, file)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
eszköz (device)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
nincs ilyen ikon az UML2-ben	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5. Milyen integrációs (vagy tesztelési) stratégia esetében használunk teszt betétet (test stub) ? (2 pont)

- top-down

Mi a funkciója a teszt betétnek (test stub) ? (2 pont)

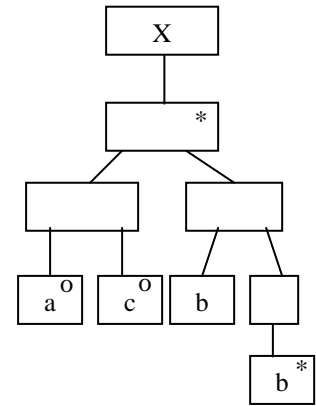
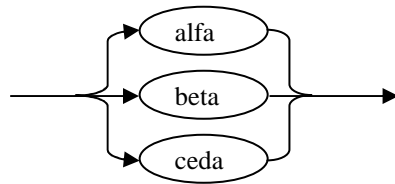
- a hiányzó programrészek (elemek, komponensek) helyettesítése

Kérdésenként egy, a magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Egészítse ki az alábbi DTD vázat úgy, hogy az a Jackson-ábra szerinti adatszerkezetet definiálja! A c elemnek legyen az alábbiak szerint definiált y nevű, predefinit alfa értékű attribútuma is! (4 pont)



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
```

```
<!ELEMENT x ((a|c), b+)*>
```

```
<!ATTLIST c y (alfa|beta|ceda) "alfa">
```

```
<!ELEMENT a (#PCDATA)>
```

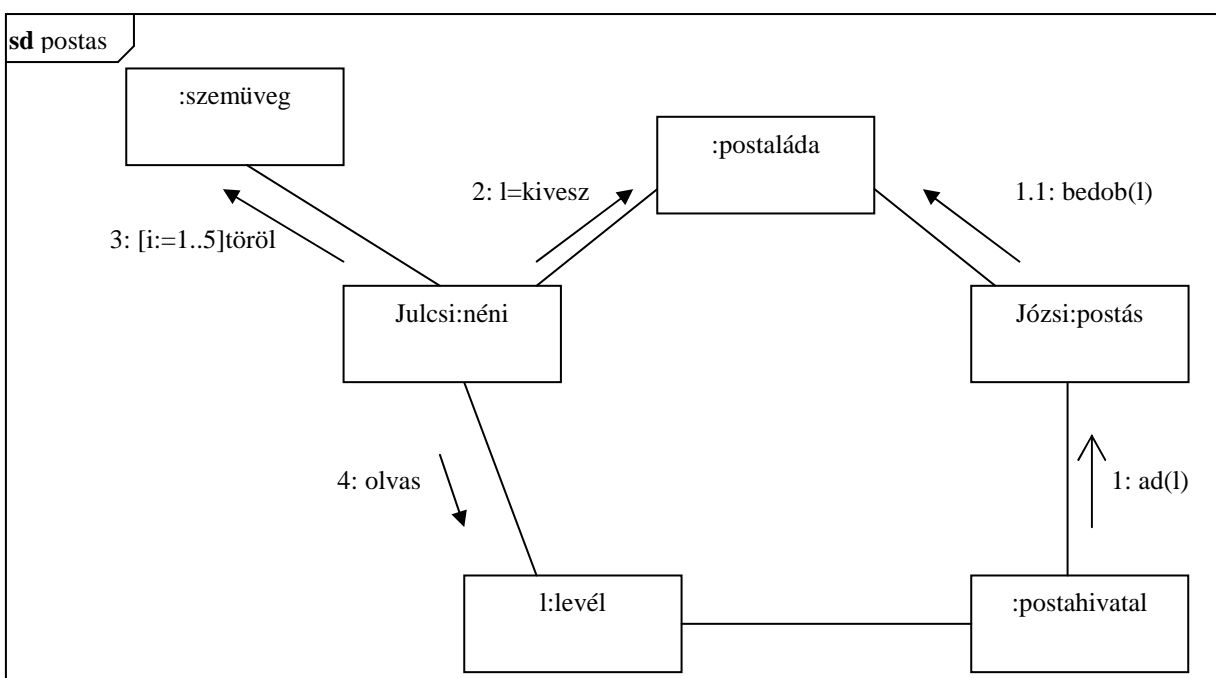
```
<!ELEMENT b (#PCDATA)>
```

```
<!ELEMENT c (#PCDATA)>
```

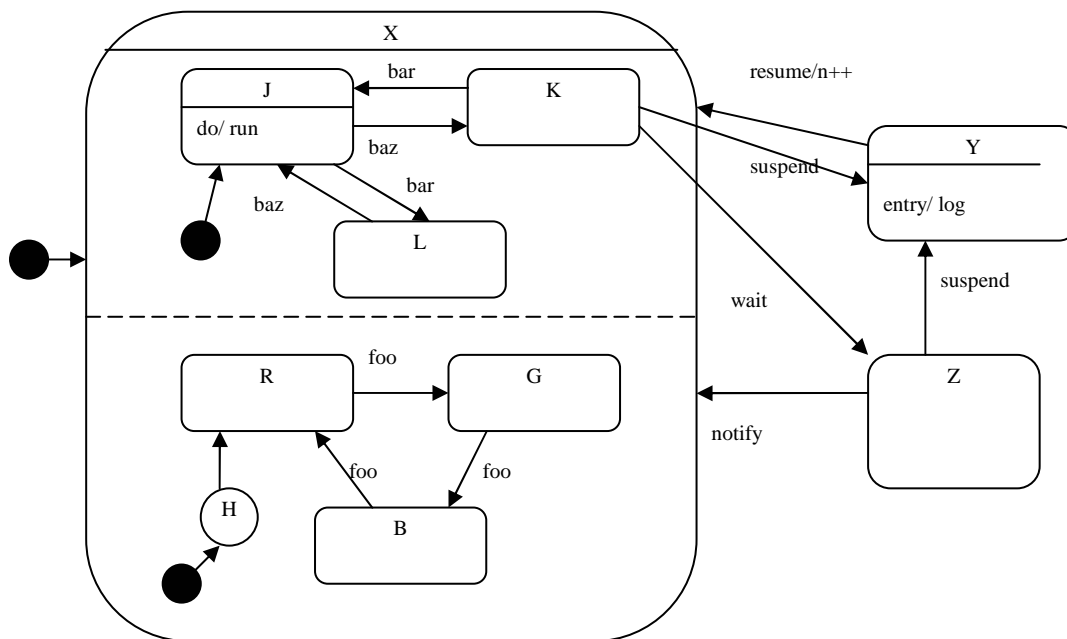

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van a, b és c elem, ez utóbbi beta attribútummal! (1 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <c y="beta"></c>
  <b></b>
  <a></a>
  <b></b>
</x>
```

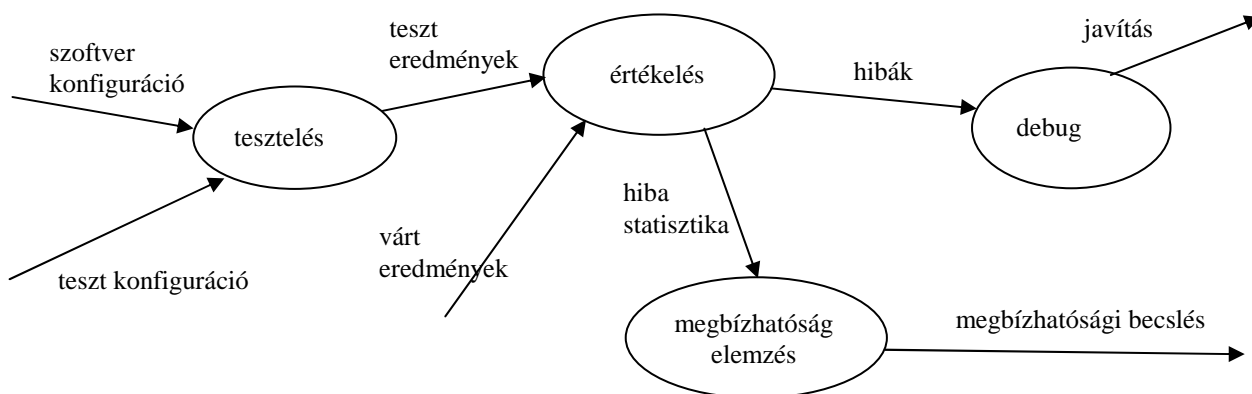
7. A postahivatal Józsinak, az énekes postásnak adja a Julcsi néninek szóló levelet. A postás a levelet bedobja a címzett postaládájába. Julcsi néni este, hazamenet, kiveszi a postaládából a levelet, majd ötször megtörli a szemüvegét (az utcán esett az eső), és elolvassa levelet. Rajzoljon UML2 kommunikációs diagramot! (7 pont)



8. Az **O** objektumnak három fő állapota van: **X**, **Y** és **Z**. A kezdő, **X** állapotban felveheti a **J**, **K** vagy **L** alállapot valamelyikét. Ezek közül **J** a kezdő, és ebben az alállapotban folyamatosan fut a **run** metódusa. **J**-ből **K**-ba a **baz**, **L**-be a **bar** eseményre kerül; **L**-ből **J**-be a **baz**, **K**-ból **J**-be a **bar** esemény viszi. Az **X** állapotban a **J-K-L** alállapotoktól függetlenül felveheti az **R**, **G** és **B** állapotok valamelyikét is. A három közül **R**-rel kezd, innen **foo**-ra a **G**-be, **G**-ből **foo**-ra a **B**-be, innen pedig **foo**-ra ismét az **R**-be kerül. Ha a **K** alállapotban van, a **suspend** és a **wait** eseményekre hagyja el az **X** főállapotot, előbbire az **Y**-ba, utóbbira a **Z**-be kerül. **Z**-ből vissza az **X**-be a **notify**, az **Y**-ba a **suspend** viszi. Az **Y** állapotba való minden belépéskor meghívódik a **log** metódus. Az **Y**-ből a **resume** viszi az **X**-be, ekkor az **n** változó értéke eggyel megnő. **X**-be minden belépéskor a **J-K-L** állapotok közül a **J**-vel kezd, az **R-G-B** alállapotok közül azzal, amelyikben az **X**-ből való legutolsó kilépésekor volt. Rajzoljon UML2 állapot diagramot ! (6 pont)



9. Rajzolja fel a tesztelés információs folyamatának adatfolyamábráját ! (6 pont)



mindegyik gombóc és él 0,5 pont.

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

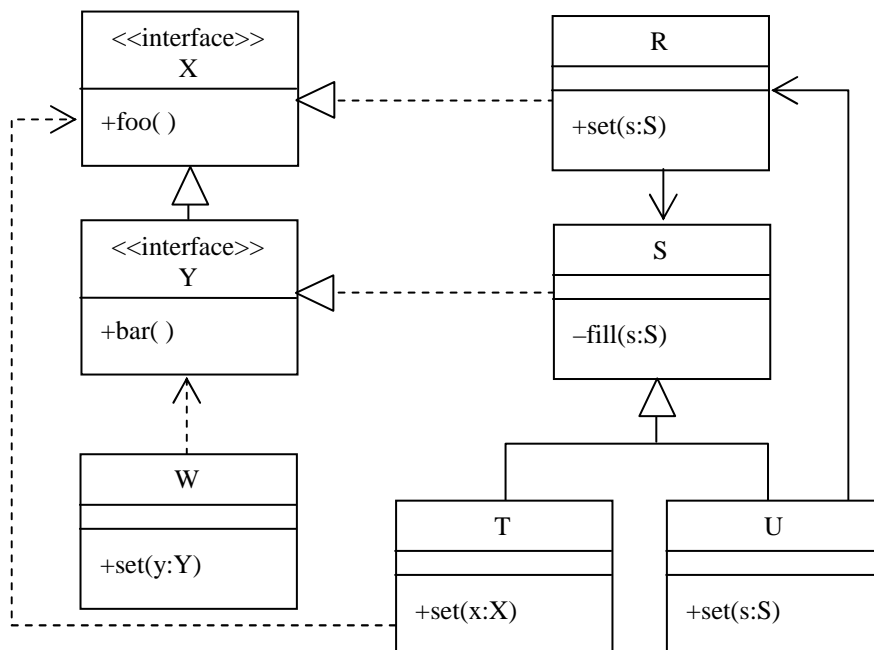
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2008. január 8. 14:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [E] Y bárhol helyettesíthető W-vel, mert W az Y leszármazottja.
- [D] U bármikor lehet T.set(x:X) paramétere, mert U megvalósítja az X interfészt.
- [B] R meghívhatja saját set(s:S) metódusából egy W set(y:Y) metódusát, mert S megvalósítja Y-t.
- [E] S fill(s:S) metódusa nem kaphat paraméterül T-t, mert a metódus protected.
- [A] T megvalósítja az X interfészt, mert T az R leszármazottja.
- [B] T pontosan egy U-t tartalmazhat, mert csak egy közvetlen ősük van.
- [E] T bárhol helyettesíthető U-val, mert egyforma az interfészük.
- [B] U meghívhatja S fill(s:S) metódusát, mert R asszociációban van S-sel.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Felsoroltunk szofverrel kapcsolatos "hibákat". A hibák mellett jelölje be, hogy az melyik kategóriába tartozik! (4 pont) Rossz válasz esetén pontot veszít !

	bug	fault	failure	error
A relációjel fordított	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hibás bemenetekre nem teszteltünk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Figyelman kívül hagytuk az objektum konstruálás időigényét	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A pointer értéke null	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A program lefut, de rossz eredményt kapunk	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
A reggeli órákban nem sikerül a szolgáltatóhoz kapcsolódn	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

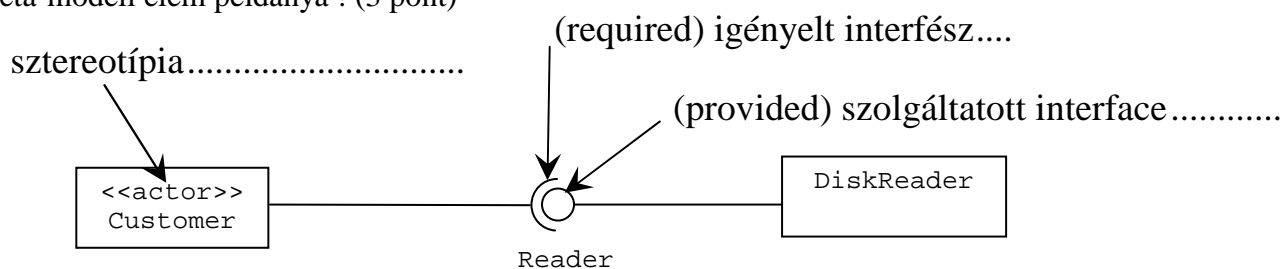
3. Mi a lényege a JSD-ben a "funkció" lépésnek (function step) ? (2 pont) Rossz válasz esetén pontot veszít !

- A funkcionális dekompozíció
- Az entitások funkcióinak specifikálása
- A processz-háló kibővítése funkció processzekkel
- A funkció processzek élettörténetének tervezése
- Az entitások halmazának kibővítése funkció-entitásokkal
- A funkció processzek ütemezése

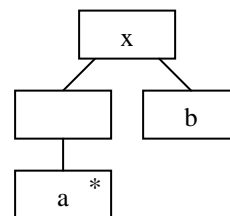
Mi az **előző** lépés? (1 pont)
initial model

Mi a **következő** lépés? (1 pont)
timing

4. Az alábbi ábrán három UML2 modell elemet megjelöltünk. Adja meg elemenként, hogy az melyik UML2 meta-modell elem példánya ! (3 pont)



5. Egészítse ki az alábbi DTD vázat úgy, hogy az a mellékelt JSP struktúra szerinti adatszerkezetet definiálja ! Az a elemnek legyen egy opcionális c attribútuma is ! (4 pont)



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
```

```
<!ELEMENT x (a*,b)>
```

```
<!ATTLIST a c CDATA #IMPLIED>
```

```
<!ELEMENT a (#PCDATA)>
<!ELEMENT b (#PCDATA)>
```

]>

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van c attribútumot tartalmazó a elem is ! (1 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <a c="CCC"></a>
  <b></b>
</x>
```

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Az alábbi DTD leírás helyes-e ? Ha nem, akkor mi(k) a hiba(k)? (3 pont)

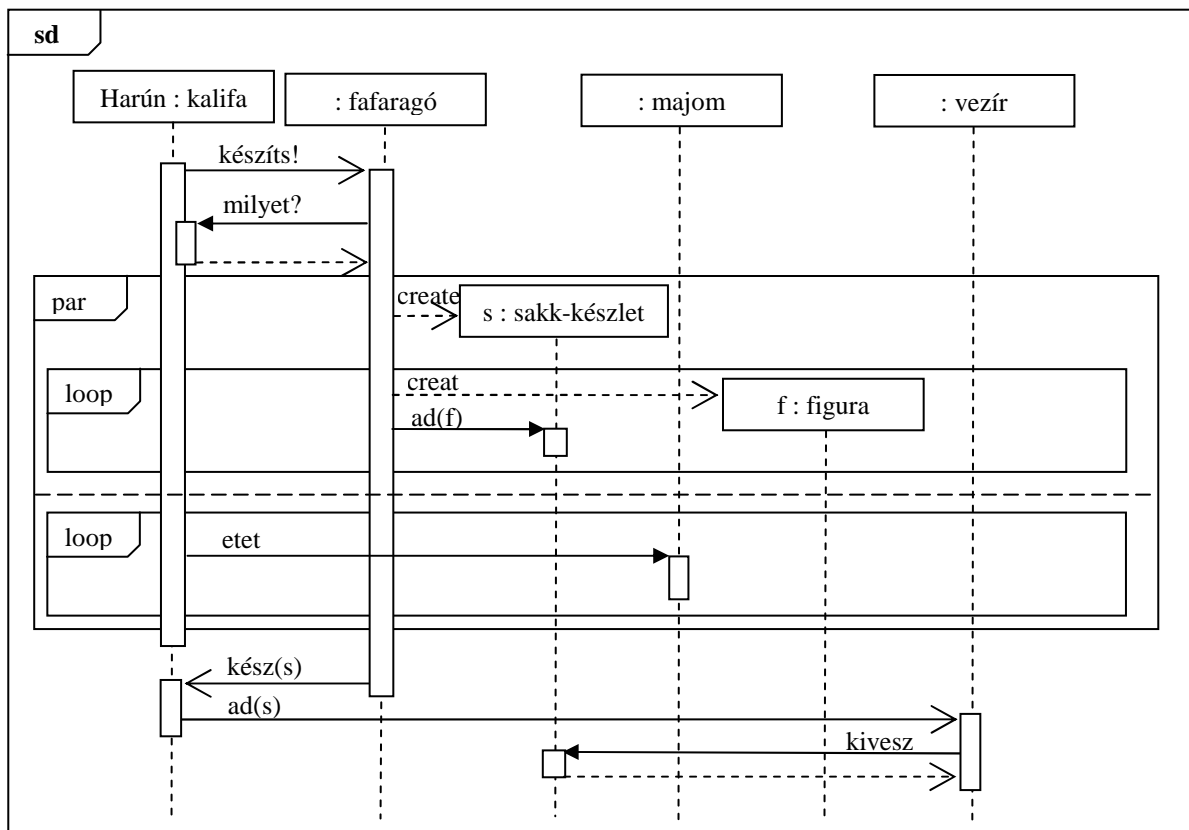
```
<!DOCTYPE JacksonDiag [
<!ELEMENT Entity (Event*)>
<!ELEMENT Event #EMPTY>
<!ATTLIST Event Optional CDATA #IMPLIED>
<!ATTLIST Entity Name CDATA> ] >
```

IGEN
 NEM

Hiba: JacksonDiag hiányzik, Name attrib sora hiányos, EMPTY elé nem kell #.....

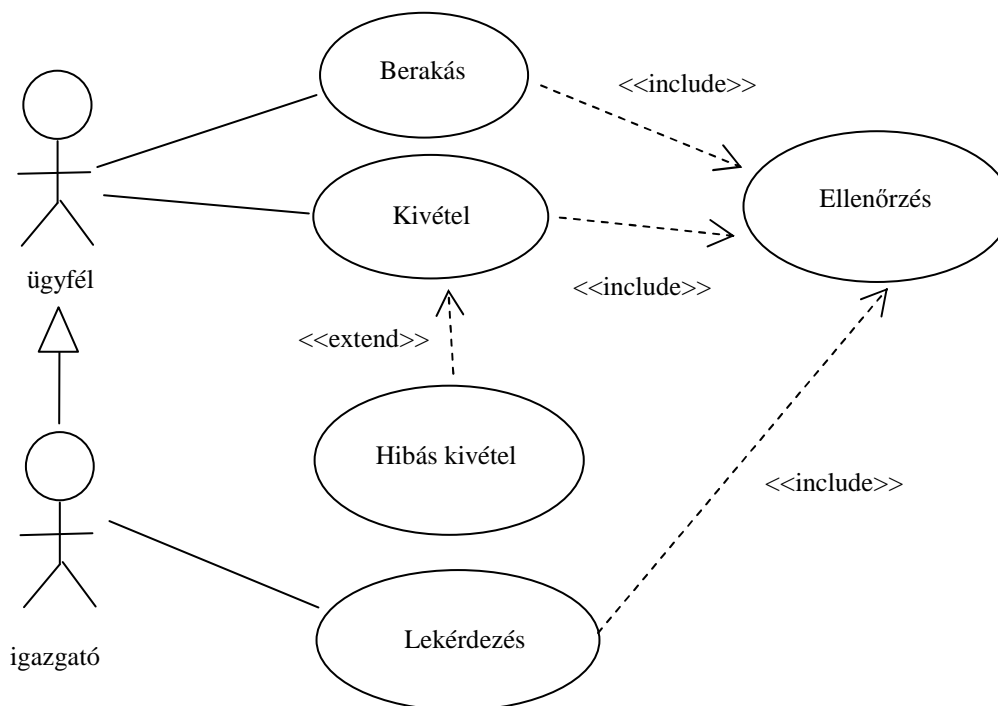
7. Rajzoljon UML2 szekvencia-diagramot az alábbi történet alapján ! (10 pont)

Harún ar-Rasíd, a bölcs kalifa meg kívánja jutalmazni vezírjét, ezért készített egy sakk-készletet udvari fafaragójával. A fafaragó, mielőtt nekilátna, megkérdi a kalifát, milyen fából készüljön, majd elkezd legyártani a remekművet. A készlethez egyenként faragja figurákat. Eközben a kalifa a kedvenc majmával szép sorban egyenként megeteti a kezében levő összes fügét. Mikor a sakk-készlet megvan, a faragó átadja a kalifának, aki továbbadja vezírjének. A vezír később, otthon kivesz a készletből egy fehér lovat.



8. Rajzoljon UML2 use-case diagramot az alábbi történet alapján! (7 pont)

A PickPack Raktárszolgáltató Kft bivalypasnádi telepén hatalmas polcrendszerekben tárolják a csomagokat. Ha egy regisztrált ügyfél csomagot hoz tárolási célból, ellenőrzik az azonosítóját, majd robotok egy üres helyre teszik a csomagot. Mikor az ügyfél elvinné egyik csomagját, ismét ellenőrzik az azonosítóját, majd előhozzák a kívánt csomagot. Előfordulhat, hogy a csomag nincs a helyén, mert már korábban elvitték. Ebben az esetben a rendszer figyelmezteti az ügyfelet a hibára. A raktár igazgatója mindazt meg tudja tenni, amit egy egyszerű ügyfél, de ő lekérdezheti a telepen tárolt csomagok számát is. Ehhez természetesen az ő azonosítóját is ellenőrzik.



9. A verziókezelő rendszerekben mit jelentenek az alábbi kifejezések (1-1 pont):

Check-out: **reserving an item for editing**

Check-in: **publish a change of an item**

A check-in esetében alkalmazott eljárásoknak (philosophies, policies) mi a lényege (2 pont)?

Reserved check-out

Modify – update – merge

A magyar nyelv szabályainak megfelelő MONDATokkal válaszoljon ! [A közreadott megoldás csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2007. január 23.

1. Mit jelent az “adatfluxus” ? A szoftver fejlesztésének melyik fázisában jelenik meg először ? (4 pont)
Kérdésenként legalább egy, a magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

- Az időegység alatt beérkező, feldolgozandó, kezelendő adatok, tranzakciók száma.
- A követelmény fázisban kell rögzíteni.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

2. Mi a lényege a JSD-ben a “kezdeti modell” lépésnek (initial model step) ? (2 pont) Legalább egy, a magyar nyelv szabályainak megfelelő, olvasható MONDATtal válaszoljon !

Az élettörténetekből processz-háló építése

Mi az előző és a következő lépés? (2 pont)

Előző: **entity-structure**

Következő: **function**

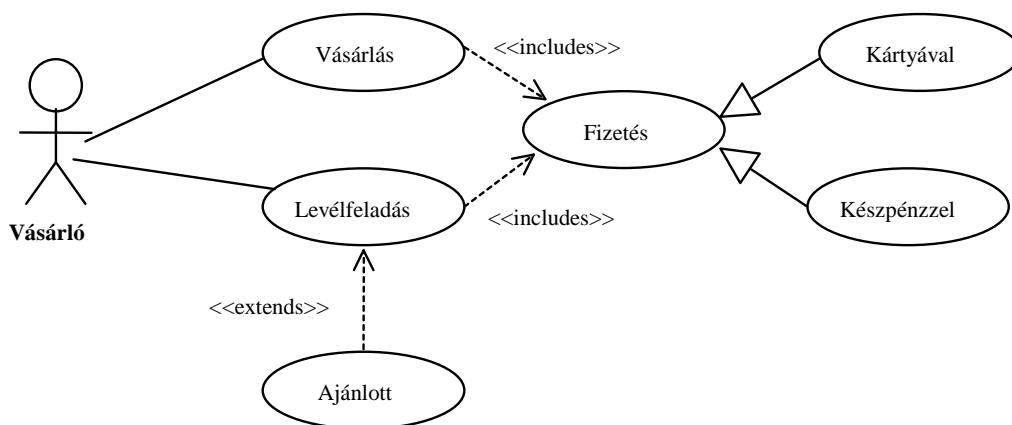
3. Definiálja az OO *operáció* és a *metódus* fogalmait ! (4 pont) Legalább egy-egy, a magyar nyelv szabályainak megfelelő, olvasható MONDATtal válaszoljon !

operáció: denotes the indivisible primitive of service provision that can be requested.

metódus: code that is executed to perform a service. A method is an immutable description of a computation that can be interpreted by an execution engine.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

4. Egy jobb angol boltocskában lehet árut (tejet, újságot, stb.) vásárolni és levelet föladni. Mindkettőnek elengedhetetlen része a fizetés, ami történhet készpénzzel vagy kártyával. Egyes boltokban lehetőség van ajánlott leveleket is föladni. Rajzoljon use-case diagramot ! (5 pont)



5. Készítsen kettő különböző, az alábbi DTD szerint érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet (3 pont) ! Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

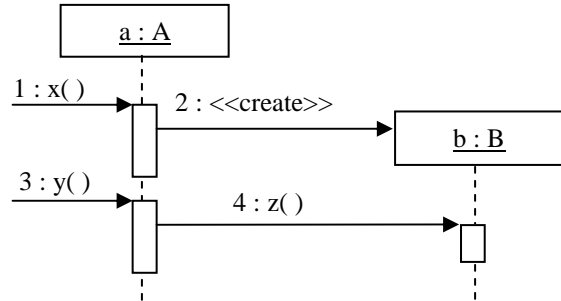
```
<!DOCTYPE x [
  <!ELEMENT x      (a? )>
  <!ELEMENT a      (b* , (c |d) )>
  <!ELEMENT b      (#PCDATA)>
  <!ELEMENT c      (#PCDATA)>
  <!ELEMENT d      (#PCDATA)>
]>
```

```
<x> </x>
```

```
<x>
  <a>
    <c>akarmi</c>
  </a>
</x>
```

6. Tételezze fel, hogy az alábbi szekvenciadiagramon szereplő objektumok osztályai között nincs más egyéb – a diagramból nem kiolvasható – kapcsolat (pl. öröklés) ! Mi a kapcsolat A és B között ? Válaszát indokolja ! Az indoklás legalább egy, a magyar nyelv szabályainak megfelelő, olvasható MONDAT legyen ! Indoklás nélkül a választása nem érvényes. (4 pont)

- függőség (dependency)
- asszociáció (association)
- kollaboráció (collaboration)
- példányosítás (instantiation)
- implementálás (implementation)



Indoklás:

A-nak "emlékeznie kell" B-re

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes értéken történő elfogadását.]

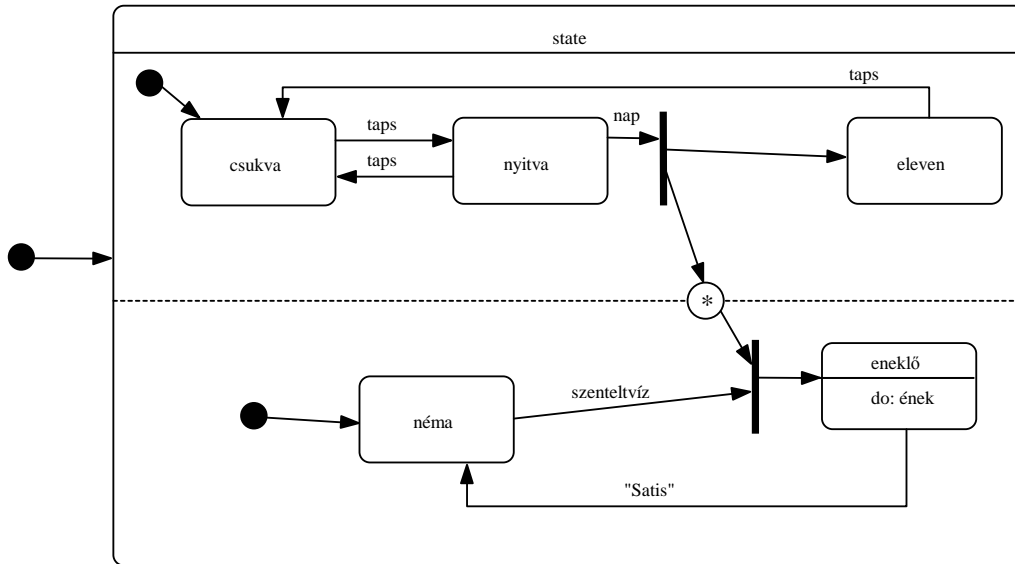
7. Adja meg az alábbi műveletekkel jellemzett stringet leíró algebrai axiómákat ! A string karakterei előről 1-gyel kezdődően számozottak. Az axiómák felírásakor használhatja a két egész összehasonlítására és két karakter összehasonlítására szolgáló műveleteket. (6 pont)

- NEW()** új (üres) stringet hoz létre.
- ADD(s,x)** az s string végére rakja az x karaktert.
- PAIR(s)** igaz, ha bárhol a stringben egymás mellett legalább két egyforma karakter áll.
- IN(s,i)** eredményül adja az s string i-ik karakterét. Ha i nagyobb mint a string hossza, akkor az eredmény értelmetlen (nem definiált).
- LGTH(s)** az s string karaktereinek számát adja.

- LGTH(NEW()) = 0 -0.5 pont, ha hiányzik
- LGTH(ADD(s, x)) = LGTH(s) + 1 -0.5 pont, ha hiányzik
- PAIR(NEW()) = false -0.5 pont, ha hiányzik
- IN(NEW(), i) = értelmetlen -0.5 pont, ha hiányzik
- PAIR(ADD(s, x)) == (LGTH(s) != 0) && (x == IN(s, LGTH(s)) || PAIR(s)) 3 pont
- IN(ADD(s, x), i) = 3 pont
 - if (i > LGTH(s) + 1) értelmetlen
 - if (i == LGTH(s) + 1) x
 - if (i < LGTH(s) + 1) IN(s, i)

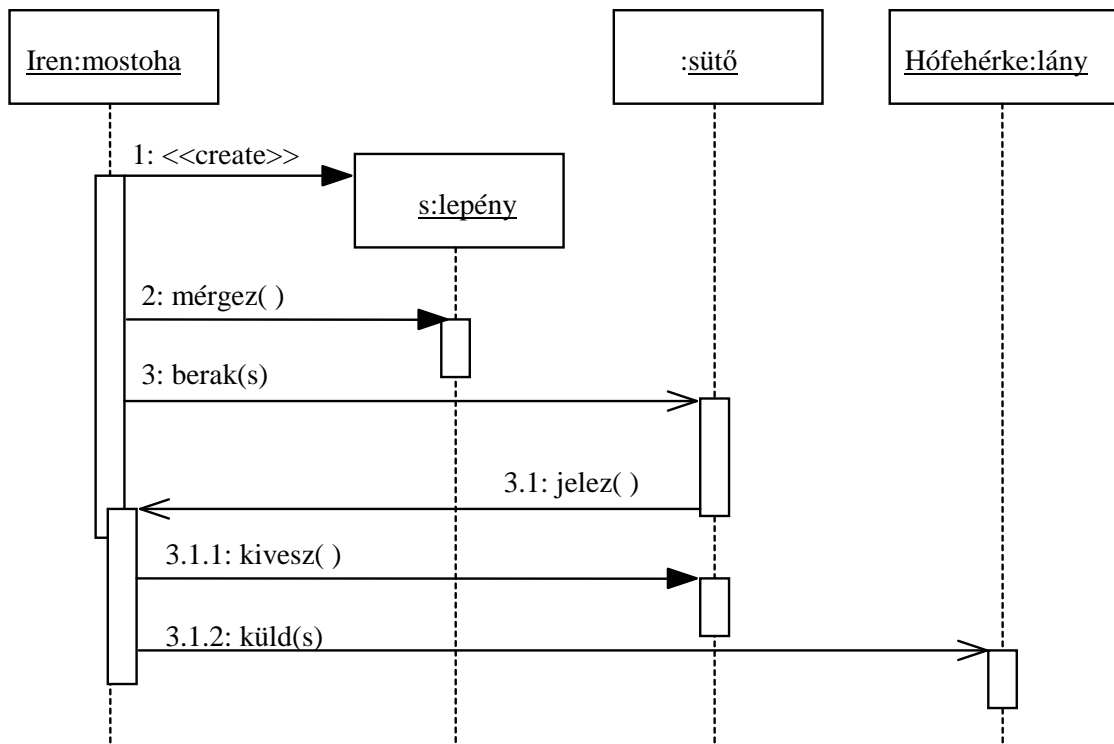
8. Készítsen UML állapotdiagramot (state chart) az alábbi leírás alapján ! (5 pont)

II Frigyes, német-római császár érdekes kódexet kapott a monte cassinoi apátságából. A kódex kezdetben csukva volt. Tapsra kinyílt, újabb tapsra becsukódott. Ha nyitott állapotában ráesett a napsugár, megelevenedtek a képei. Tapsra azonban ilyenkor is becsukódott, és a képek is visszamerevedtek. Mindezek közben a könyv néma volt. Ha ugyanezen napsugár ráesése után valamikor szenteltvízzel meghintették, beneventán antifónákat énekelt, függetlenül attól, hogy éppen nyitva vagy csukva volt. Egészen addig énekelt, amíg azt nem mondták neki, hogy „Satis!”, mikor is ismét néma lett.

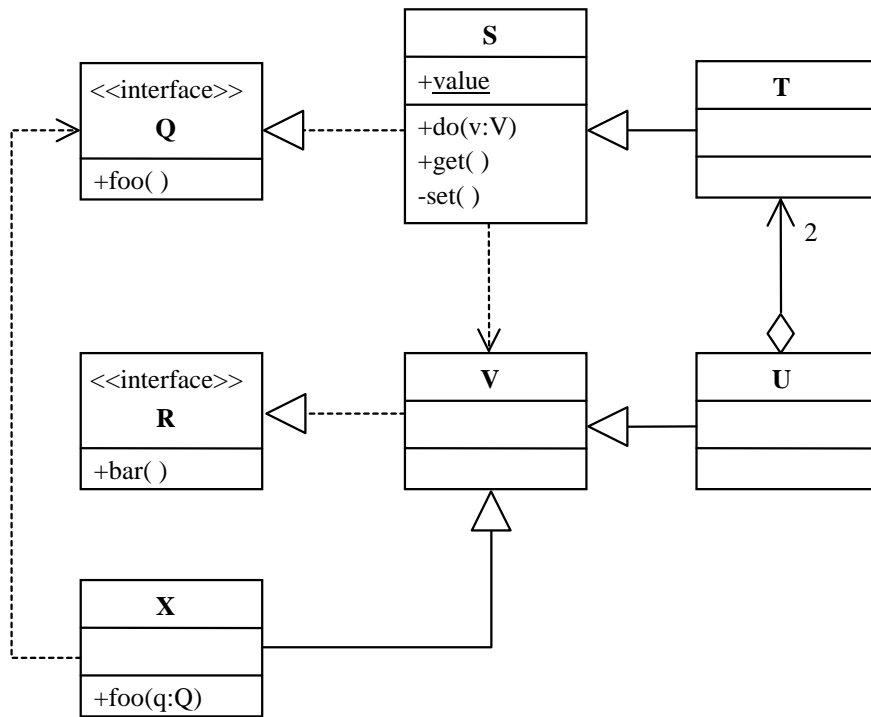


9. Készítsen UML szekvencia diagramot az alábbi leírás alapján ! (5 pont)

Irén, a gonosz mostoha meg akarja mérgezni Hófehérkét. Ehhez készít egy lepényt, mérget tesz bele, majd berakja a sütőbe sülni. Míg a lepény sül, gonosz gondolatokon jár az esze. Fél óra múlva a sütő jelez, hogy kész van. Erre Irén kiveszi a lepényt, és elküldi Hófehérkének.



10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)

- [E]** X foo(q:Q) metódusa kaphat paraméterül U-t, mert U megvalósítja a Q interfészt.
[B] T value attribútuma nem érhető el U-ból, mert az attribútum statikus.
[E] V helyettesíthető S-sel, mert mindketten megvalósítják a Q interfészt.
[A] T meghívhatja egy X bar() metódusát, mert X függ az S osztálytól.
[C] S nem hozhat létre U osztályú objektumot, mert nem ismeri a T osztályt.
[A] X meghívhatja egy Q interfészű objektum foo() metódusát, mert X-nek is van azonos szignatúrájú metódusa.
[E] U megszűntekor 2 T is megsemmisül, mert U kompozíciós kapcsolatban áll 2 T-vel.
[B] S set() metódusa nem módosíthatja a value attribútumot, mert a metódus nem publikus.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

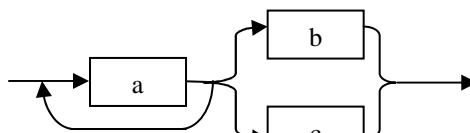
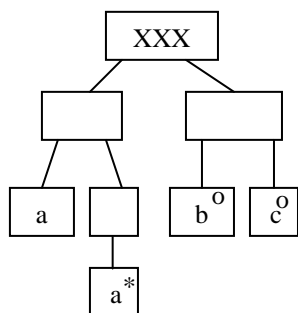
c. tárgyból
2007. január 16.

1. Mit nevezünk “legacy software”-nek ? Miért fontos a szoftver technológia szempontjából ? (2 pont)
Kérdésenként legalább egy, a magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

- has exceeded its economic life time - but cannot be abandoned.
- nagy karbantartási igény, refactoring, re-engineering.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megismétlése nem szükségképp vonja maga után annak teljes érteken történő elfogadását.]

2. Az XXX entitás élettörténetét az alábbi JSD ábra definiálja. Írja le ugyanezen élettörténetet szintaxis gráffal !
(Az élettörténetben szereplő eseménynek feleljen meg az azonos nevű szimbólum !) (4 pont)



3. Készítsen kettő különböző struktúrájú, az alábbi DTD szerint érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet (6 pont) ! Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

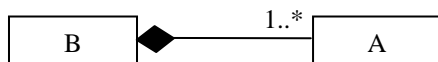
```
<!DOCTYPE x [  
  <!ELEMENT x (d|b)>  
  <!ELEMENT d (a,c*)>  
  <!ELEMENT a (#PCDATA)>  
  <!ELEMENT b (#PCDATA)>  
  <!ELEMENT c (#PCDATA)>  

```

```
<x>  
  <b>barmi</b>  
</x>
```

```
<x>  
  <d>  
    <a>akarmi</a>  
  </d>  
</x>
```

4. Az A osztály példányai a B osztály egy példányával együtt keletkeznek és pusztulnak el. Rajzoljon olyan UML diagramot, amely leírja a két osztálynak ezt a tulajdonságát ! (4 pont)

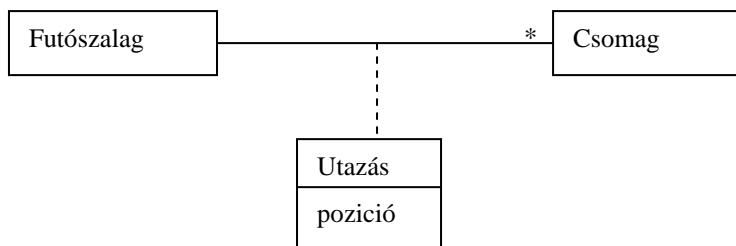


5. Milyen integrációs (vagy tesztelési) stratégia esetében használunk teszt ágyakat (test bed) ? Mi a funkciója a teszt ágyaknak ? (4 pont) Kérdésenként legalább egy, a magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

- **bottom-up**
- **környezet, amelyben az integrált unitok együttes működése tesztelhető.**

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes érteken történő elfogadását.]

6. Egy futószalagon csomagok utaznak. A szalagot elhagyó csomagot a későbbiekben is használni akarjuk. Ki tudja (kinek a felelőssége), hogy mi a csomag pozíciója a futószalagon ? Legyenek az osztályok kohézívek és lazán csatoltak ! A probléma modellezésére rajzoljon UML osztálydiagramot a pozíció feltüntetésével ! (4 pont)



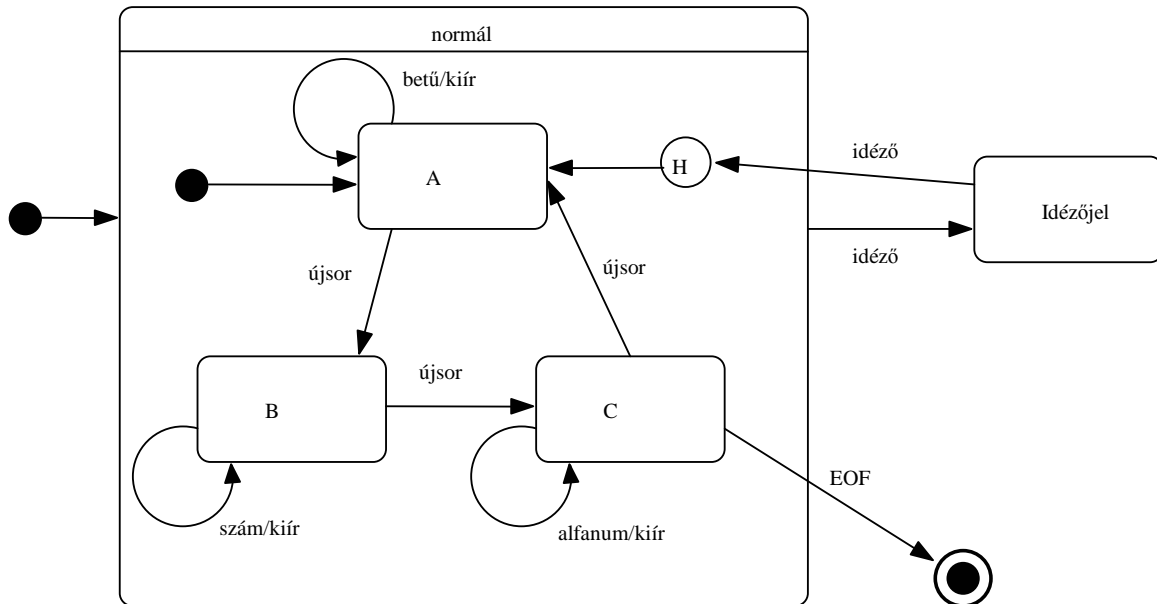
7. Mit jelent a „szoftver konfigurációs menedzsment (CM)” ? Mi a „konfiguráció” ? (4 pont) Kérdésenként legalább egy, a magyar nyelv szabályainak megfelelő MONDATtal válaszoljon !

- CM is the discipline of *controlling the evolution* of complex systems; software CM is its specialization for computer programs and associated documents
- A configuration is a selection of configuration items designating a state of the project.

[Ez csak vázlat és útmutató a javító számára, tartalmazza a válasz lényegét. Szó szerinti megisméltése nem szükségképp vonja maga után annak teljes érteken történő elfogadását.]

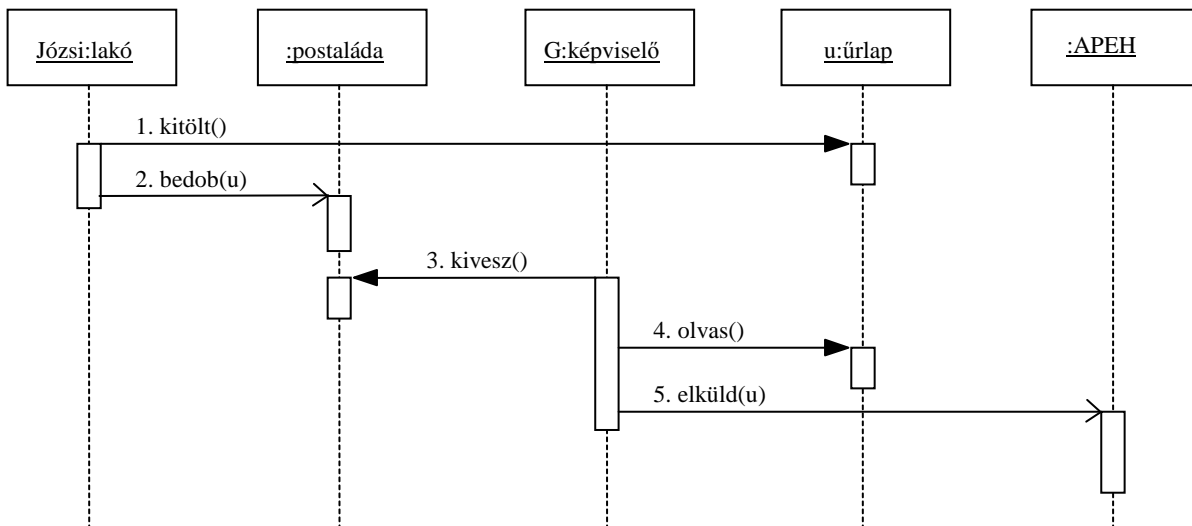
8. Készítsen UML állapotdiagramot (state chart) az alábbi leírás alapján ! (6 pont)

A parser normál állapotban három (A,B,C) műveletet végez. Kezdetben A-ban van. Ha ilyenkor betűt kap, kiírja. Sorvégejelkor átvált B-re. Ekkor csak a számjegyeket írja ki. Sorvégejelkor C-re vált: ilyenkor minden betűt és számot kiír. Újabb sorvégre ismét A-ba kerül. C-ben fájlvégejelre befejezi a működést. Ha bármely állapotában idézőjel jön, onnantól nem ír ki semmit, csak ha újabb idézőjel jön, ekkor az utoljára otthagyt módot folytatja.

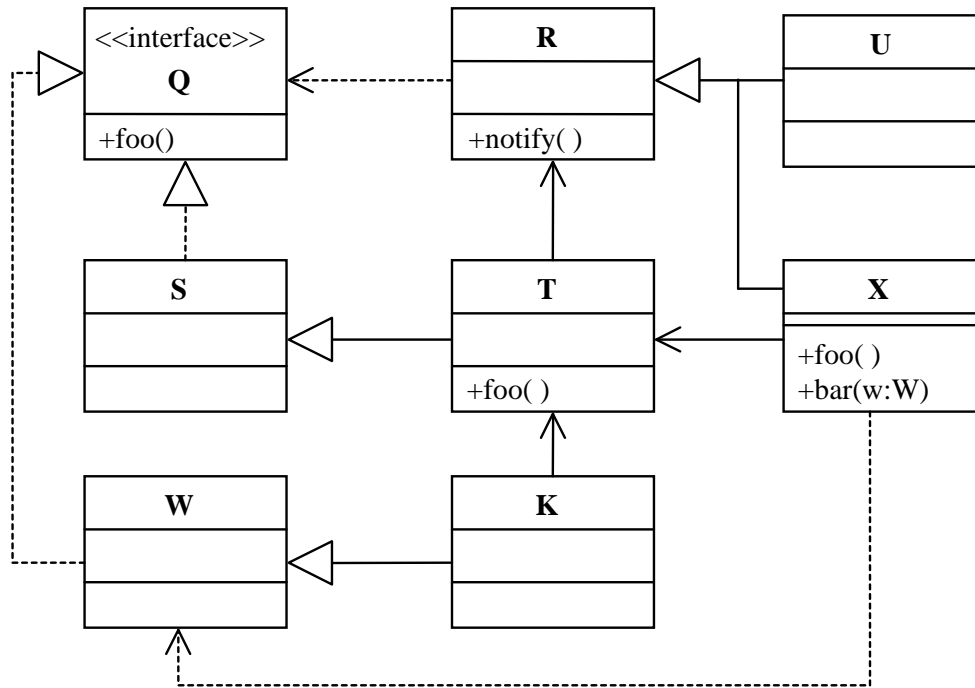


9. Készítsen UML szekvencia diagramot az alábbi leírás alapján ! (6 pont)

Józsi, a bimbó úti társasház egyetlen minimálbéres vállalkozója kitölti a gázártámogatási űrlapot, és bedobja a közös képviselő postaládájába. A közös képviselő, Géza, kiveszi az űrlapot, elolvassa, és mivel nem szereti Józsit, elküldi az űrlapot az APEH-nek.



10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)

- [B]** *W* és *S* bárhol felcserélhetők, mert közös interfészt implementálnak.
[C] *T* meghívhatja egy *X* osztályú objektum *foo()* metódusát, mert *X* függ a *Q* interfésztől.
[A] *X* létrehozhat egy *T* osztályú objektumot, mert van közös ősük.
[E] *R* helyettesíthető *T*-vel, mert *T* az *R* leszármazottja.
[B] *K* meghívhatja egy *R* osztályú objektum *notify()* metódusát, mert *K* megvalósítja a *Q* interfészt.
[B] *K* nem hívhatja meg egy *T* *foo()* metódusát, mert *K*-nak is van ugyanilyen szignatúrájú metódusa.
[A] *X* *bar(w:W)* metódusa kaphat paraméterül *K* osztályú objektumot, mert mindketten *S* leszármazottai.
[E] *U* meghívhatja egy *X* *foo()* metódusát, mert van közöttük asszociáció.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

VIZSGA FELADATSOR
SZOFTVERTECHNOLÓGIA

c. tárgyból

2007. január 9. 13:00

1. Definiálja a szoftver hibával kapcsolatos alábbi fogalmakat ! (4 pont)

Bug is a software defect (incorrect step, process, or data definition) that causes a failure

Error human action or omission that results in a fault.....

Failure the inability of a software to perform its required functions within specified performance requirements

Fault same as Bug

2. A kockázat tervezése során milyen stratégiákat választunk ? (6 pont)

A kockázat valószínűségének csökkentése

A bekövetkező káros hatások minimalizálása

A nem kívánatos eseményt követően felmerülő tevékenységek tervezése

3. Írja le az alább definiált adatstruktúrát DTD-ben ! Legyen a, b, c és d parsed code data. (3 pont)

x = a + [{b + [c | d]} | a]

```
<!DOCTYPE x [
  <!ELEMENT x (a, ((b, (c | d)) * | a))>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
  <!ELEMENT d (#PCDATA)>
]>
```

4. Mit jelent a "baseline" ? (4 pont)

A baseline is a 'snapshot' in time of one version of each artifact in the project repository. It provides an official standard on which subsequent work is to be based, and to which only authorized changes can be made. After a baseline is set every subsequent change to a baseline is recorded as a delta until the next baseline is set.

5. A tesztelés információs folyamatában álló „értékelés” (evaluation) processznek mi(k) a be- és kimenete(i) ? (6 pont)

Bemenet

Kimenet

teszt eredmények (test results)

hibák (errors)

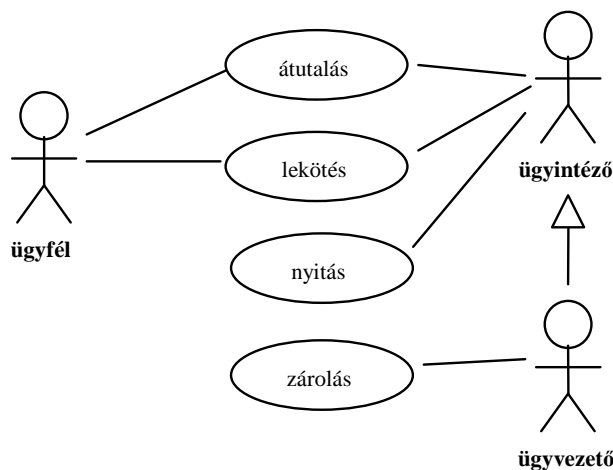
várt eredmények (expected results)

hiba statisztika (error statistics).....

.....
 mindegyik jó 1,5 pont, mindegyik hiba -1, de total >= 0

6. Rajzoljon UML use-case diagramot a következő leírás alapján ! (4 pont)

A *MoneyOrLife* bankban az ügyfelek a weben átutalást és lekötést kezdeményezhetnek. A fiókokban az ügyintézők átutalás és lekötés mellett új számlát is nyithatnak. A fiók ügyvezetője az ügyintézői jogosultságokon kívül a számlákat zárolhatja is.

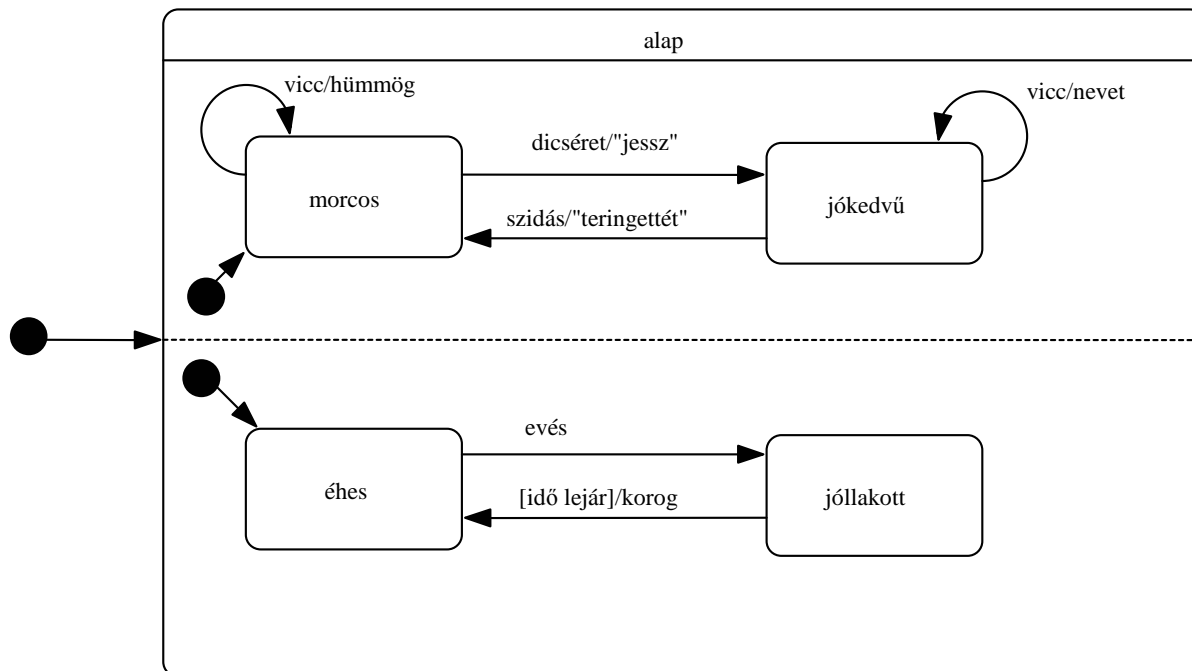


7. Mi a lényege a RUP (Rational Unified Process) életciklus „átmeneti” (transition) fázisának ? (3 pont)

supplying the product to the user community (manufacturing, delivering, and training)

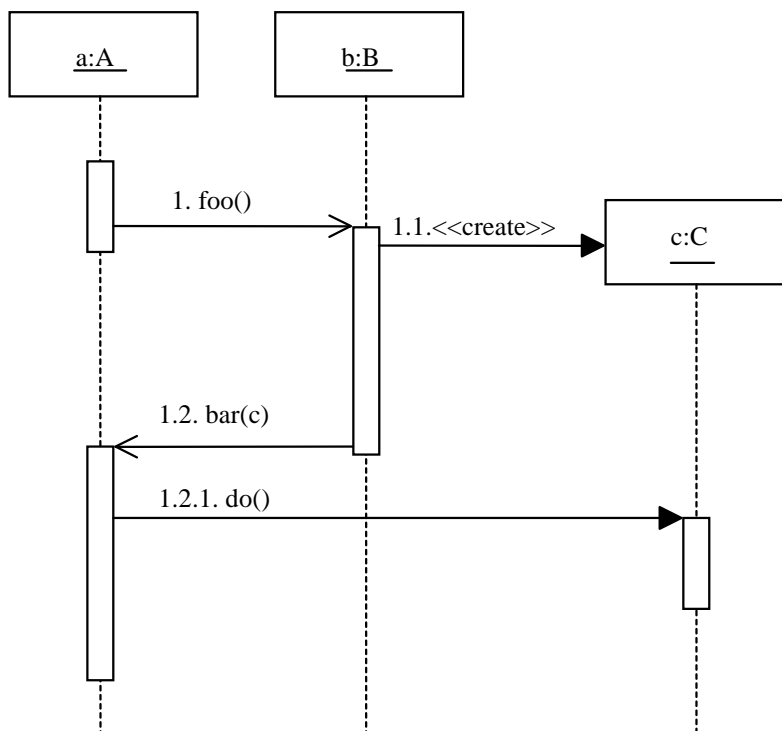
8. Készítsen UML állapotdiagramot (state chart) az alábbi leírás alapján ! (5 pont)

Izidor alaptól morcos kisfiú. Ha ekkor viccet hall, csak hümmög. Ha megdicsérik, azt mondja, „jessz”, és jókedvű lesz. Ekkor a vicceken nevet, de, ha megdorgálják, azt mondja, hogy “teringgettét”, és ismét morcos lesz. Mindeközben többnyire éhes (már éhesen született). Ha kap enni, akkor jóllakott. Rövid idő elteltével azonban újra éhes lesz, és ekkor kordul egyet a gyomra.

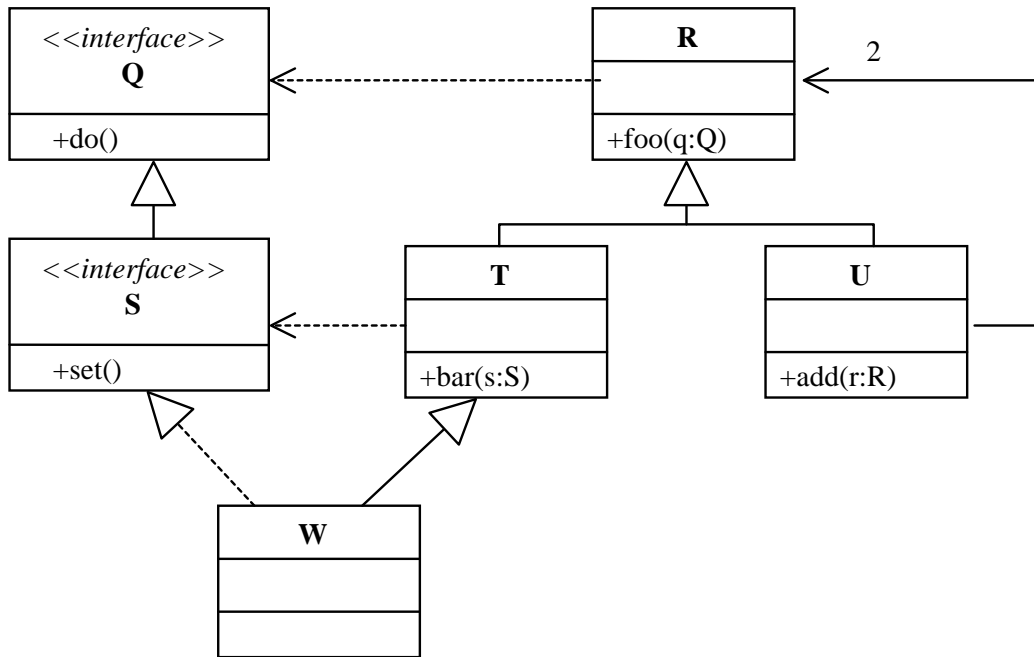


9. Készítsen UML szekvencia diagramot az alábbi leírás alapján ! (5 pont)

Az **A** osztályú **a** objektum meghívja aszinkron módon a **B** osztályú **b** objektum **foo()** metódusát és passzív állapotba kerül. Erre **b** létrehoz egy **C** osztályú objektumot (**c**), és meghívja **a** objektum **bar()** metódusát **c** paraméterrel, szintén aszinkron módon. A **bar()** hatására **a** meghívja paraméterül kapott objektum szinkron **do()** metódusát.



10. Az alábbi UML diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)

- [B] T helyettesíthető U-val, mert T és U is az R-nek leszármazottja.
 [D] R helyettesíthető W-vel, mert W az R leszármazottja.
 [E] T bar(s:S) metódusa kaphat paraméterül T-t, mert T megvalósítja az S interfészt.
 [A] T bar(s:S) metódusából meghívhatjuk egy paraméterül kapott W do() metódusát, mert W nem valósítja meg a Q interfészt.
 [C] U nem hívhatja meg T bar(s:S) metódusát, mert a metódus nem statikus.
 [A] R meghívhatja egy Q interfészű objektum do() metódusát, mert R megvalósítja a Q interfészt.
 [C] U pontosan 2 R-t ismer, mert R-nek 2 közvetlen leszármazottja van.
 [B] U add(r:R) metódusa nem kaphat W-t paraméterül, mert U nem ismeri W-t.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

Eredmények értékelése:

Pontszám	Osztályzat
25 -	2
34 -	3
43 -	4
52 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

2000. május 24.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet! Az axiómák felírásakor megengedett a két karaktert összehasonlító művelet használata. (6 pont)

CRT()	új (üres) stringet hoz létre.
ADD(s,x)	az s string végére rakja az x karaktert.
LAST(s)	az s string végén álló karaktert adja.
END(s1,s2)	igaz, ha az s2 string az s1 string végén áll.
DUPLO(s)	igaz, ha az s stringben legalább egyszer dupla karakter fordul elő. Például: aggódik

2. Készítsen entitás-relációs diagramot az alábbi problémára. (6 pont)

A Lynkenstein kereskedőház kiterjedt üzlethálózata segítségével kedvező áru termékekkel látja el vásárlóit. A boltokban sokféle áru (pilótadzseki, bűváróra, szabókréta, tollseprű, stb.) kapható. Az akciós árukat kivéve ugyanazon árut az üzlethálózat minden boltjában egyforma áron adnak. Az árak naponta változnak. Időnként egy-egy boltban bizonyos termékeket akciós áron árulnak. Ugyanazon termék akciós ára boltonként különböző lehet. A vállalat a törzsvevőinek kártyát ad. A kártyás vásárlásokat tételesen nyilvántartják. Ennek alapján a cég a nagy vásárlókat alkalmanként megjutalmazza (és vásárlói profilt készít). Jutalomra számíthat a legnagyobb forgalmú bolt személyzete is.

3. Mi a history indicator? Hogyan jelöljük és mire használjuk? (3 pont)

4. Készítse el egy mozi jegyárusításának adatfolyamábráját! Adja meg a context diagramot és annak egy mélyezési felbontását! (6 pont)

A mozi előadásait az előadás jellemzőinek (film címe, előadás időpontja, helyárak) megadásával a mozi vezetője definiálja. A pénztáros a definiált és még be nem fejezett előadásokra készpénz fizetése ellenében jegyet ad, amely egyben számla is. A jegyen rajta van a mozinak az adózással kapcsolatos adatain kívül a film címe, az előadás időpontja, az elfoglalt székek azonosítója (erkély, bal, 2. sor 2-5 szék) és a jegy ára. Egy jeggyel több szék is foglalhatunk. A pénztárosnál telefonon előjegyezhetünk jegyet.

5. Mi a multiobject? Hogyan jelöljük és milyen diagramon fordul elő? (3 pont)

6. Adjon JSD alapú megoldást az alábbi problémára! (10 pont)

A Pílisi Természetbarátok egyesület évente levelező feladatmegoldó versenyt hirdet meg az interneten általános iskolai csapatok számára. A versenyre a PT honlapján, egy kérdőív kitéltésével kell jelentkezni. A jelentkezéssel egyidőben kell megadni annak a két tanárnak a nevét is, akik a megoldásokat fogják majd javítani. A jelentkezés elfogadásakor minden csapat egyéni azonosítót és jelszót kap, amelyet e-mail-ben küldenek el neki. Ezt követően kéthetente csapatonként legfeljebb 1-1 feladat hívható le az azonosító és jelszó segítségével.

A feladatokra a megoldásokat elektronikusan kell elküldeni az egyesület e-mail címére. A beadás sorrendje tetszőleges, de csak azokat a megoldásokat értékelik, amelyek a verseny utolsó napjáig beérkeznek. A beküldött megoldásokat a jelentkezéskor megadott tanároknak küldik tovább, akiknek két héten belül kell azokat kijavítani. A verseny korrektsége miatt egy megoldást többen is (legalább hárman és nem abból az iskolából, ahonnan a megoldás jött) javítanak. A verseny lezárásakor a csapatnak küldött e-mailben és az interneten minden csapatról statisztikát tesznek közzé, amely tartalmazza a csapat nevét, azonosítóját, a lekért feladatokat, a lekérések időpontját és a csapat eredményét.

Azonnal jelezzen, ha valamelyik csapat a harmadik feladatát kéri le. Kérésre adja meg, hogy mely feladatok vannak még javító tanároknál. Kérésre listázza ki a már kiadott azonosítókat és jelszavakat. Kérésre adja meg azokat a tanárokat, akik az átlagosnál kevesebb javítási feladatot kaptak.

Készítse el az entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modellt, funkció processzeket. kapcsolja a modellhez a

7. Rajzolja le egy grafikus editor class diagramját. Egy grafikus dokumentum lapokból áll. A lapokon grafikus elemek találhatóak, ezek közé tartozik a szöveg, a 2D alakzat és a csoport. Egy csoport legalább két grafikus elemből áll. Csoport eleme lehet csoport is. Egy grafikus elem közvetlenül csak maximum egyetlen csoportba tartozhat. A 2D objektumok: kör, ellipszis, négyzet, vonal, téglalap. (5 pont)

8. Legyenek olyan objektumok (Originátorok), amelyek állapotát időközönként menteni kell - a Memento interface-t megvalósító objektumba - hogy az később esetleg undo-val visszaállítható legyen. A Memento realizáló objektumokat egy Caretaker objektum kezeli, ő utasítja az Originátorokat az állapot elmentésére Memento createMemento(), és visszaállítására setMemento(Memento) operációkkal. A Memento interface tartalmazza a setState() és getState() operációkat.

Hogyan oldható meg, hogy egy konkrét Mementóra vonatkozóan a setState() operáció csak egyszer hajtódjon végre? (2 pont)

Adja meg a class diagramot, az operációkkal együtt. (5 pont)

Rajzoljon szekvencia diagramot arra az esetre, amikor a Caretaker egy Originátornál az állapot elmentését kezdeményezi, majd később visszaállítja az elmentett állapotot. (4 pont)

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

2000. május 31.

1. Alábbi axiómák segítségével specifikálja az alábbi műveletekkel jellemzett stringet! Az axiómák felírásakor megengedett a két karaktert összehasonlító művelet használata. (6 pont)

CRI()	új (üres) stringet hoz létre.
ADD(s,x)	az s string végére rakja az x karaktert
LAST(s)	az s string végén álló karaktert adja
END(s1,s2)	igaz, ha az s2 string az s1 string végén áll.
ISIN(s1,s2)	igaz, ha az s2 string az s1 stringben bárhol előfordul.

2. Készítsen entitás-relációs diagramot az alábbi problémára! (6 pont)

Egy vasúttársaság több mozdonytípust is használ. A típusokra jellemző a fajta (gőz, diesel, villamos), a tergelnyomás és a vonóerő. A típus kódja egy háromjegyű szám vagy egy betű és egy kétdjegyű szám. Dönteselőkészítési és összehasonlítási célból olyan típusok adatait is tárolják, melyek mozdonyval nem rendelkezők. A mozdonyokat a típus és egy három vagy negyedik sorozat azonosítja. Jellemzője meg a gyártás éve. A mozdonyok vonatokat továbbítanak, minden vonat pontosan egy mozdony. A vonat jellemzője az osztotomege, a dátum, a késés és a vonatszám. Ugyanazzal a vonatszámmal nem indulhat két vonat ugyanazon a napon. A vonatszámától függ a vonat fajtája (személy, tehers), az induló- és célállomás és a közbúlsó állomások, valamint az indulás és az érkezés időpontja az állomásokon (a közbúlsó állomásokon is). Az állomások jellemzője a név, a vágányok száma és a szolgálat létszáma.

3. Mi a kapcsolat a funkcionális és dinamikus modell között? (3 pont)

4. Készítse el az alábbi parkolást felügyelő rendszer adatfolyamábráját! Adja meg a context diagramot és annak egy mélységi felbontását! (6 pont)

A parkoló autó vezetője a bejáratnál elhelyezett nyomógombot megnyomva kap egy kártyát, majd a felnyíló sorompón keresztül behajt a parkolóba. Kilépéskor az autó a kijárathoz áll, ahol a vezető a belépéskor kapott kártyát egy olvasóba helyezi. A rendszer a kártyáról leolvassa a belépési időt, majd az aktuális idő és a tarifabizlat alapján kiszámítja a parkolási díjat, amelyet kijelez az autós számára egy kijelzőn, valamint a kezelő emelőjén. A vezető átadja a kezelőnek a parkolási díjat, mire a kezelő számlát nyomtat és kinyitja a sorompót. A kártya nélkül autót a kezelő kiléptetheti a maximális tarifá megfizetése esetén. A bejáratnál egy kijelzőről leolvasható a szabad helyek száma. A parkoló működése folyamatos, a kezelők váltják egymást. A szolgálat választásokról a forgalomról és a bevetelről a rendszer jegyzőkönyvet készít. A kezelőnek lehetősége van lekérni a szolgálati időjele vonatkozó forgalmi adatokat

5. A tervezés során bevezetjük a kohézió és csatlós fogalmait. Mit jelentenek? Milyen fő típusaik vannak? (4 pont)

6. Adjon JSD alapú megoldást az alábbi problémára! (10 pont)

A Kistéleki Általános iskolában minden év május 31.-én diákigazgatót választanak. A diákigazgatókat az osztályok javasolják, a választási bizottsághoz beküldött adatlappal, melyen megadják a jelölt nevét, osztályát, és a választási bizottsághoz delegált ellenőrök nevét (2 fő). Ha a bizottság a jelöltet elfogadta, a jelölt számára egy LOGO-t ad, melyet jogában áll a választási kampányában a plakátjain szerepeltetni. A választási hajrákat

szórán a minden jelölt nagygyűléseket tarthat, de hetente legfeljebb csak egyet. A nagygyűlések programját a tervezett időpont előtt legalább 1 héttel a bizottságnak le kell adni (de természetesen lehetőség van akár a teljes kampányprogram egyidejű benyújtására is). A bejelentett programot a bizottság más jelöltek ellenőrével (legalább 3-mal) megvizsgálhatja. Egy ellenőr egyszerre csak egy programot bírálhat. Az ellenőröknek 3 napjuk van, hogy véleményezzék a programot. A választási kampány lezárása után a bizottság összefoglaló plakátot készít a jelöltek kampányolásáról, melyen szerepelteti a jelölt nevét, LOGO-ját, a nagygyűlések számát, időpontját és látogatottságát, valamint az ellenőrök által adott véleményeket.

Azonnal jelezzék, ha valamelyik jelölt már az ötödik nagygyűlést jelenti be. Kérésre adja meg, hogy mely nagygyűlést programját véleményezik meg ellenőrök. Kérésre listázza ki a már kiadott LOGO-kat. Kérésre adja meg azokat az ellenőröket, akik az átlagosnál több kifogást emeltek.

Készítse el az entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az inicial modellt, kapcsolja a modellhez a funkció processzektet.

7. Rajzoljon use-case diagramot és magas szinten (high-level) definiálja a use-case-eket! (6 pont)

A mozi előadásait az előadás jellemzőinek (film címe, előadás időpontja, helyárak) megadásával a mozi vezetője definiálja. A pénztáros a definiált előadásokra készpénz fizetése ellenében jegyet ad. A pénztárosnál telefonon előjegyezhetünk jegyet.

8. Készítsen olyan objektum modellt, amely alkalmas a LÉTRA játék leírására. (9 pont)

A LÉTRA játék a hagyományos asztali társasjáték egy változata. A pálya a starttól a célig tart, közben 100 darab számozott mező van, amelyek közül bizonyosok kitüntetettek.

A játékosokat reprezentáló bábók a starthelyről indulnak és a kockadobás értékének megfelelően lépnek előre.

A játékok célja a célmező elérése.

A kitüntetett mezők szerepe az, hogy - létrékatn szolgálva - a mezőre lépő bábút egy másik mezőbe (általában hátrébb) dobja át.

Függős szabály, hogy egy mezőn csak egyetlen bábu állhat. A később érkező bábu a már ott állót kiüti, ami azt jelenti, hogy a bábu a start mezőből indulhat újra.

Rajzoljon class diagramot!

Adja meg az operációs szignatúráit!

Rajzoljon kollaborációs diagramot arra az esetre, ha egy bábnak Lép j (lépésszám) üzenetet küldünk. A lépés hatására a bábu lépjen egy kitüntetett mezőre, majd ezen átszussza össze ki az éppen ott álló bábút.

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
LÁSZLÓ ZOLTÁN
2000. január 11.

1.A és C csoport Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett listát, amelynek elemei olyan (x,y) párok, ahol x egy kulcs, y egy 0-nál nagyobb egész ! (6 pont)

NEW()	új (üres) listát hoz létre
ADD(I,(x,y))	az I listához kapcsolja az x,y párt, ha x nem szerepelt a listán. Ha a listán már van x , - akkor a hozzá tartozó y -t az új y -nal helyettesíti, ha az nagyobb a listán szereplőnél.
VALUE(I,x)	megadja az I listán az x kulcshoz tartozó y -t. Ha a listán a megadott x nem szerepel, akkor az eredmény 0.
SUM(I)	képezi a listán szereplő elemek y értékeinek összegét.
SMALL(I)	a listában szereplő legkisebb y . (üres lista esetén nulla)

1.B és D csoport Algebrai axiómák segítségével definiálja a páratlan számok korlátos - az alábbi műveletekkel jellemezett - halmazát. A halmaz csak páratlan számot őriz meg, a páros számok beillesztése hatástalan. A halmaz legfeljebb tíz elemet tartalmazhat, a további elemek beillesztése hatástalan. A hatástalan művelet nem hiba, az eredmény az eredeti halmaz! Az axiómák felírásánál használhatja az **ODD(n)** logikai értékű függvényt. (6 pont)

CRT()	új (üres) halmazt hoz létre
ADD(s,n)	a fent definiált módon beszúr egy elemet a halmazba
ISIN(s,n)	megadja, hogy a eleme-e az s halmaznak (logikai érték)
SIZE(s)	a halmaz elemeinek száma
SMALL(s)	a halmaz legkisebb eleme (üres halmaz esetén nulla)
BIGS(s)	a halmaz a legkisebb elem elhagyása után

2.B és C csoport Készítsen entitás-relációs diagramot a következő szöveges specifikáció alapján. (6 pont)
Harácsfalva/Motorino nagyközség önkormányzata bevezette a motorkerékpár-adót. Ennek értelmében a tulajdonos a motor típusától, korától és műszaki állapotától függően napi X forint adót köteles fizetni. Ennek kapcsán nyilvántartják a motorokat és azok gazdáit. A motorokat rendszeresen műszaki vizsgálatra kell vinni. Előírás, hogy a vizsgáztató két egymást követő vizsgán nem lehet ugyanaz a személy, valamint nem lakhat a motor éppen aktuális tulajdonosával egy utcában. Motori no lakossága a motorokat rendszeresen adja-veszi, cseréli, kidobja, ezért az idüarányos adózás miatt a tulajdonlásban beálló változást és annak dátumát be kell jelenteni. Mivel az emberek nem üzérkednek, feltételezhető, hogy egy motor egy napon egyszer cserél gazdát. Egy gazdának több motorja is (ehet, de közös tulajdonú motor nincs. Az önkormányzat az adó mértékét évente rendeletben szabályozza.

2.A és D csoport Készítsen entitás-relációs diagramot a következő szöveges specifikáció alapján. (6 pont)
Meseváros/New York City útfelügyeletének feladata a közutak hibáinak javítása. A bejelentett úthibákat

nyilvántartásba veszik, megadva a bejelentés idejét, a gödör pontos helyét (utca, házszám, az út széléhez képesti helyzet), a luk méretét egy 10 fokozatú skálán, valamint a javítás sürgősségét. Ez utóbbit meghatározza a gödör mérete, és az utca forgalma. Egy hiba javítására az útfelügyelet munkalapot állít ki. Ezen szerepelnek a gödör azonosító adatai, a javítást végző brigád azonosítója, a javításhoz szükséges eszközök, a felhasznált anyagok mennyisége, a javításhoz szükséges munkaidő, a gödör állapota a javítást követően (kijavított, ideiglenesen javított, nem javított), valamint a javítás költsége, amit az eszközökből, az elhasznált anyagokból, a javítóbrigád létszámából és a javítási időből lehet kiszámítani. Az útfelügyelet nyilvántartja az egyes úthibák okozta károkat, rögzítve a károsult autó rendszámát, a tulajdonos nevét, címét és a kárösszeget.

3.A és C csoport Mi a kapcsolat a funkcionális és dinamikus modellek között ? (3 pont)

3.B és D csoport Hogyan jelölik az interfészt az UML-ben ? Példaként ábrázolja azt az esetet, amikor az Ember osztály megvalósítja a Kártyajátékos interfészt. (3 pont)

4.B és C csoport Készítse el egy jegyárúsító automata adatfolyamábráját! Adja meg a context diagramot és annak 0 szintű felbontását ! (6 pont)

A jegyárúsító automatát metróállomásokon szerelik fel, céljuk metrójegyek eladása. Az automatán található billentyűzettel az utazónak meg kell adni a jegy típusát (gyerek, felnőtt, nyugdíjas, katona illetve egy útra szóló vagy retúrjegy) és a célállomás nevét. Ezen adatok alapján az automata kijelzi a jegy árát. Az utazó 10, 20, 50 és 100 forintos pénzérmeget dobálhat be a résen, mialatt a hátralévő összeget jelzi ki a gép. Amíg a kívánt összeget az utazó még nem dobta be, a Cancel gomb megnyomásával a vásárlástól elállhat és visszakapja a pénzét. Ha a bedobott pénz mennyisége eléri vagy meghaladja a jegy árát, az automata egy jegyet nyomtat feltüntetve a kiadó és cél állomás kódját, a jegy típusát, árát és a vásárlás időpontját, és visszaadja az esetleg visszajárót. Amennyiben az automata elromlik, a gép hibát jelez és nem reagál a gombnyomásokra. Az eladott jegyekre vonatkozó statisztika és a pénzforgalom gyűjtött adatai vonalon keresztül a központból lekérdezhetőek.

4.A és D csoport Készítse el egy italárúsító automata adatfolyamábráját! Adja meg a context diagramot és annak 0 szintű felbontását ! (6 pont)

Az automatán található billentyűzettel a vásárlónak meg kell adni az ital fajtáját (cola, light, juice, kakaó, leves) és a kért mennyiséget. Ezen adatok alapján az automata kijelzi az árát. A vásárló 10, 20, 50 és 100 forintos pénzérmeget dobálhat be a résen, mialatt a hátralévő összeget jelzi ki a gép. Amíg a kívánt összeget a vásárló még nem dobta be, a Cancel gomb megnyomásával a vásárlástól elállhat és visszakapja a pénzét. Ha a bedobott pénz mennyisége eléri vagy meghaladja az ital árát, az automata kiadja az italt és egy nyugtát nyomtat feltüntetve a kiadott ital fajtáját, mennyiségét, árát, a vásárlás időpontját, és visszaadja az esetleg visszajárót. Amennyiben az automata elromlik, a gép hibát jelez és nem reagál a gombnyomásokra. Az eladásokra vonatkozó statisztika és a pénzforgalom gyűjtött adatai vonalon keresztül a központból lekérdezhetőek.

5.B és C csoport Mi a jelentése és hogyan jelöljük az UML-ben a korlátozást ? Mutasson példát rá. (3 pont)

5.A és D csoport Mi a history indicator? Hogyan jelöljük és mikor használjuk? (3 pont)

6.A és C csoport Adjon JSD alapú megoldást az alábbi problémára ! (10 pont)

A Távoktatási Főiskola (TF) minden félévben kurzusokat hirdet meg hallgatói számára. Egy hallgató több kurzusra is beiratkozhat. Csak az a kurzus indul, amelyre 10-en befizették a tandíjat. A hallgatóknak a félév során hetente feladatot kell megoldaniuk, és a megoldást e-mail-ben a kurzusvezetőnek elküldeniük. A feladatra kapott pontszámról ugyancsak e-mailben 3 napon belül értesítést kapnak. A félév végén vizsgát kell tenni. Csak az vizsgázhat, aki a feladatok legalább 80 %-át beküldte és legalább 10 pontos átlagot ért el. Minden kurzuson legalább 3 vizsgát tartanak, de egy tárgyból egy hallgató legfeljebb kétszer vizsgázhat. Azok a diákok, akik egy

kurzusra az összes feladatot beküldték, és maximális pontot értek el, valamint két felvett tárgyból jelesre vizsgáztak, visszakapják az egyik kurzusra befizetett tandíjukat.

- Kérésre adja meg azom vizsgák listáját, amelyen senki nem vett részt
- Kérésre adja meg azok listáját, akik valamely tárgyból nem vizsgázhatnak
- Azonnal jelezzen, ha valakinek vissza kell fizetni a tandíjat

Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modelt,**
 kapcsolja a modellhez **funkció processzeket.**
 a

6.B és D csoport Adjon JSD alapú megoldást az alábbi problémára ! (10 pont)

A Középiskolások Tanulmányi Egyesülete (KTE) évente levelező tanulmányi versenyeket hirdet meg különböző témakörökben. Csak az a verseny indul, amelyre legalább 50 diák jelentkezik. (Természetesen egy diák akárhány témakörben versenyezhet.) A résztvevőknek 3 hetente kell egy postán kapott feladatlapot megoldani és a szervezőknek visszaküldeni. Az előző forduló kijavított feladatát a következő forduló feladatával együtt mindenki megkapja. Év végén a legjobb feladatmegoldóknak versenyeket rendeznek. Azokat hívják meg, akik a feladatok legalább 70 %-át visszaküldték és összesen több mint 100 pontot ének el. Egy témakörben legalább 3 versenyt rendeznek, egy diák azonban legfeljebb csak kettőn indulhat, és a jobb eredményét veszik figyelembe. Az a diák, aki egy témában az összes feladatot visszaküldte, és még legalább két témában a záró-versenyen az első 10 hely valamelyikén végzett, az egyesület elnökétől különdíjat kap.

- Azonnal jelezzen, ha valakit elnöki különdíjra kell felterjeszteni
- Kérésre adja meg azon záró-versenyek listáját, ahol senki nem jelent meg
- Kérésre adja meg egy záró-versenyre meghívandók listáját

Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modelt,**
 kapcsolja a modellhez **funkció processzeket.**
 a

7. (mindenhol ugyanaz, csak a pótórák száma más) Rajzoljon konkurens állapotokat is tartalmazó UML statechart-ot, amely leírja az autóvezetői tanfolyam hallgatójának dinamikus viselkedését. (8 pont)

Az autóvezetői iskolában párhuzamosan tartják a KRESZ, a műszaki és az elsősegélynyújtó tanfolyamokat. Vezetési gyakorlatra csak az a tanuló mehet, aki letette a KRESZ-ből a vizsgát. Vezetésből csak az kísérhet meg vizsgát, aki valamennyi elméleti vizsgáját letette és legalább 30 óra vezetési gyakorlata van. Minden sikertelen vizsga ismételhető. Ismétlő vizsgát vezetésből csak az tehet, aki az utolsó eredménytelen vizsgáját követően legalább 10 pótórát vett gyakorlatból.

7.D csoport Rajzoljon konkurens állapotokat is tartalmazó UML statechart-ot, amely leírja az autóvezetői tanfolyam hallgatójának dinamikus viselkedését. (8 pont)

Az autóvezetői iskolában párhuzamosan tartják a KRESZ, a műszaki és az elsősegélynyújtó tanfolyamokat. Vezetési gyakorlatra csak az a tanuló mehet, aki letette a KRESZ-ből a vizsgát. Vezetésből csak az kísérhet meg vizsgát, aki

valamennyi elméleti vizsgáját letette és legalább 30 óra vezetési gyakorlata van. Aki bizonytalannak érzi magát, vehet még pótórát. Minden sikertelen vizsga ismételhető. Ismétlő vizsgát vezetésből csak az tehet, aki az utolsó eredménytelen vizsgáját követően legalább 5 pótórát vett gyakorlatból.

8. (mindenhol ugyanaz, csak a metódusok neve más) Legyen egy tetszőleges O osztály. Ennek egy olyan O_x változatát kell elkészíteni, amelyik rendelkezik O valamennyi tulajdonságával + egy p példány a `create_a_copy ()` metódusának meghívásakor készít magáról egy m másolatot, amely folyamatosan követi p állapotának minden változását. p -ről több másolat is készíthető, de m már nem másolható tovább. Egy m másolaton egy kliens részéről kezdeményezett olyan műveletet, amely m állapotának megváltozását okozná, m továbbítja p -nek, aki a műveletet saját magát hajtja végre, hogy valamennyi másolata kövesse a változtatást.

Adja meg O_x attribútumait.

Rajzoljon szekvencia diagramot arra az esetre, amikor p -ről egy kliens készít két másolatot (m_1 , m_2), majd az m_1 -nek meghívja egy olyan metódusát (`modif ()`), amely m_1 -nek módosulását okozná. (8 pont)

Eredmények értékelése:

Pontszám	Osztályzat
21-	2
28-	3
35-	4
42-	5

*lista, elemei olyan (x,y) párok, ahol
x egy kulcs, y egy 0-nál nagyobb egész !*

NEW() új (üres) listát hoz létre.

ADD(L,(x,y)) az L listához adja az x,y párt, ha x nem szerepel a listán. Ha a listán már van x, akkor a hozzá tartozó y-t az új y-nal helyettesíti, ha az nagyobb a listán szereplőnél.

IN(L,x) megadja, hogy x kulcsu elem szerepel-e az L listában.

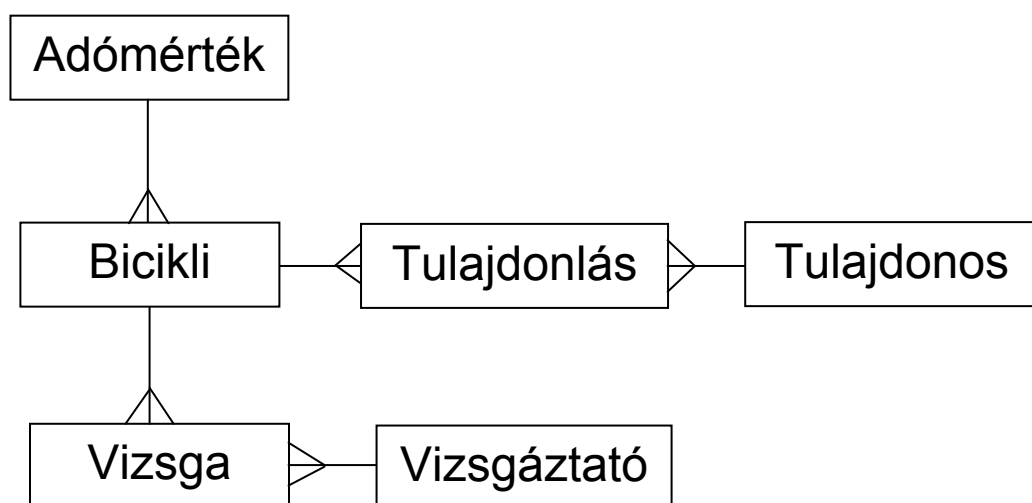
VALUE(L,x) megadja az L listán az x kulcshoz tartozó y-t. Ha a listán a megadott x nem szerepel, akkor az eredmény 0.

SUM(L) az elemek y értékeinek összege.

- **IN(NEW(),x) = false**
- **VALUE(NEW(), x) = 0**
- **SUM(NEW()) = 0**
- **IN(ADD(L,(x1,y)), x2) = (x1 == x2) or IN(L,x2)**
- **VALUE(ADD(L,(x1,y)),x2) = if (IN(L,x2)) then
if (x1 == x2) and (y > VALUE(L,x1)) then y
else VALUE(L,x2)
else if (x1 == x2) then y else 0**
- **SUM(ADD(L,(x,y))) == if (IN(L,x)) then
if (y > VALUE(L,x))
then SUM(L) - VALUE(L,x) + y
else SUM(L)
else SUM(L) + y**

Készítsen entitás-relációs diagramot a következő szöveges specifikáció alapján.

Harácsfalva nagyközség önkormányzata bevezette a bicikliadót. Ennek értelmében a bicikli tulajdonosa a bicikli típusától, korától és műszaki állapotától függően napi X forint adót köteles fizetni. Ennek kapcsán nyilvántartják a bicikliket és azok gazdáit. A bicikliket rendszeresen műszaki vizsgálatra kell vinni. Előírás, hogy a vizsgáztató két egymást követő vizsgán nem lehet ugyanaz a személy, valamint nem lakhat a bicikli éppen aktuális tulajdonosával egy utcában. Harácsfalva lakossága a bicikliket rendszeresen adja-veszi, cseréli, kidobja, ezért az időarányos adózás miatt a tulajdonlásban beálló változást és annak dátumát be kell jelenteni. Mivel a harácsfalviak nem üzérkednek, feltételezhető, hogy egy bicikli egy napon egyszer cserél gazdát. Egy gazdának több biciklije is lehet, de közös tulajdonú bicikli nincs. Az önkormányzat az adó mértékét évente rendeletben szabályozza.



Készítse el egy italárusító automata adatfolyamábráját !
Adja meg a context diagramot és annak 0 szintű felbontását

Az automatán található billentyűzettel a vásárlónak meg kell adni az ital fajtáját (cola, light, juice, kakaó, leves) és a kért mennyiséget. Ezen adatok alapján az automata kijelzi az árat. A vásárló 10, 20, 50 és 100 forintos pénzérméket dobálhat be a résen, mialatt a hátralévő összeget jelzi ki a gép. Amíg a kívánt összeget a vásárló még nem dobta be, a Cancel gomb megnyomásával a vásárlástól elállhat és visszakapja a pénzét. Ha a bedobott pénz mennyisége eléri vagy meghaladja az ital árát, az automata kiadja az italt és egy nyugtát nyomtat feltüntetve a kiadott ital fajtáját, mennyiségét, árát, a vásárlás időpontját, és visszaadja az esetleg visszajárót. Amennyiben az automata elromlik, a gép hibát jelez és nem reagál a gombnyomásokra. Az eladásokra vonatkozó statisztika és a pénzforgalom gyűjtött adatai vonalon keresztül a központból lekérdezhetőek.

Adjon JSD alapú megoldást az alábbi problémára ! (12 pont, 99. június 3.)

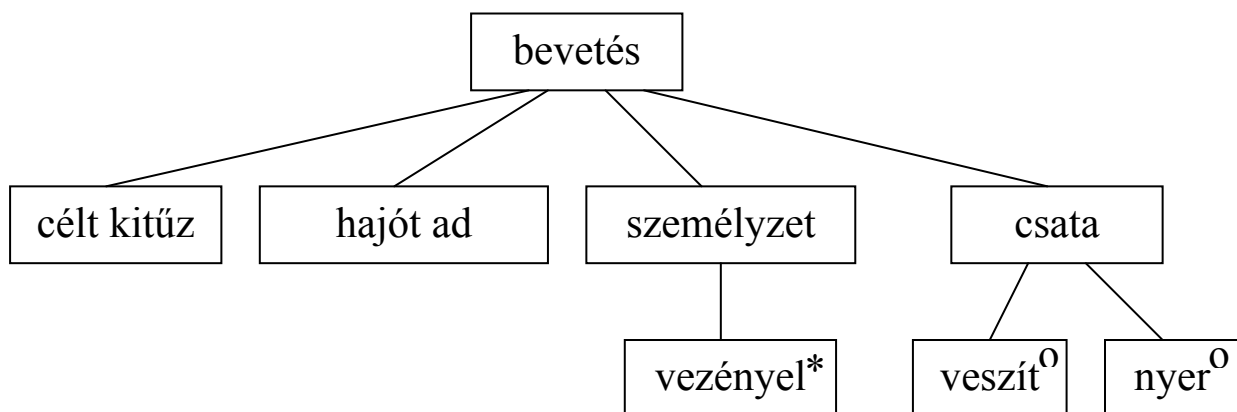
Az MZ/X űrparancsnokság K2 osztályú csillaghajókat vet be azzal a céllal, hogy a folyamatosan terjeszkedni akaró ellenséges zebereket visszaszorítsák, űrbázisaikat elpusztítsák. A parancsnokság meghatározza a bevetés célját, majd csillaghajót és személyzetet rendel az akcióhoz. A csillaghajó a megadott célt általában sikeresen támadja, ám a zeber űrelhárítás sem tétlenkedik és bizony előfordul, hogy egy-egy hajó személyzettel együtt odavész.

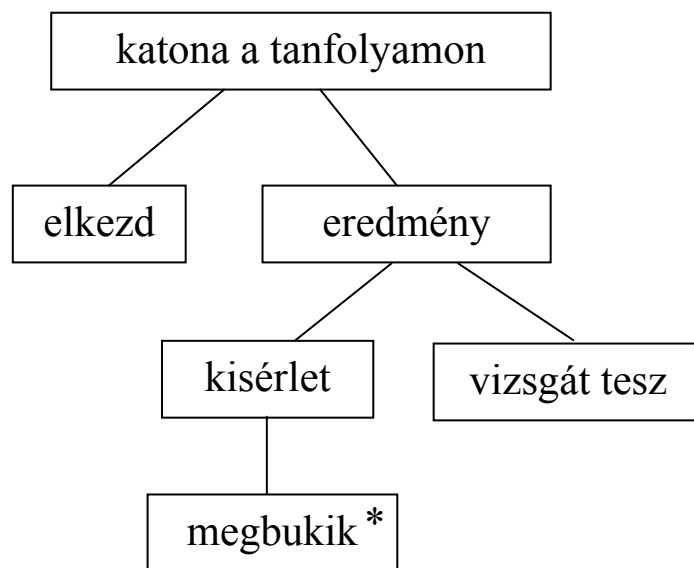
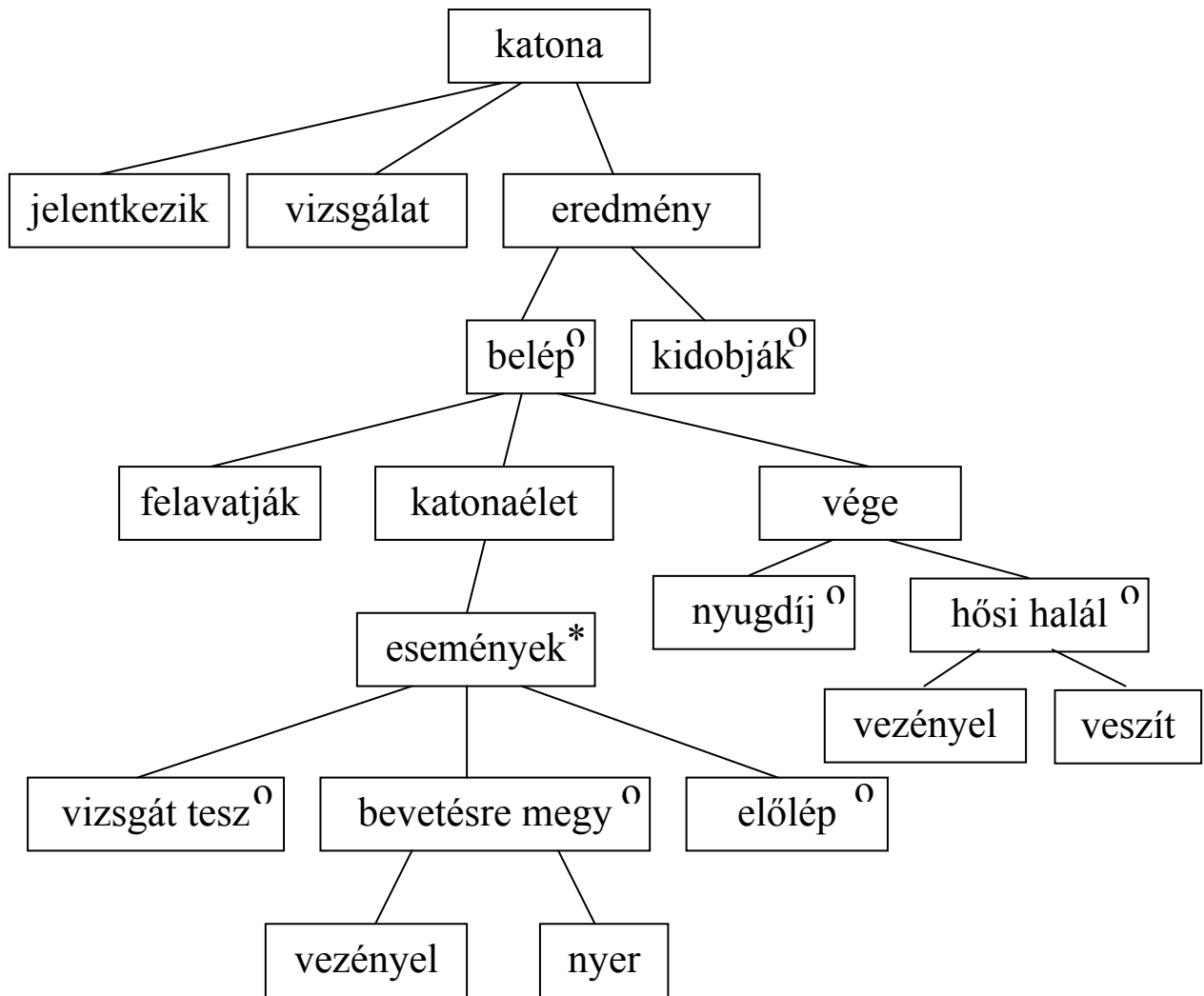
A nehéz szolgálat és veszteség indokolja, hogy a parancsnokság folyamatosan személyzetet toboroz. Ha a jelentkező az intelligencia és orvosi vizsgálaton megfelel, akkor a legközelebbi kiképző kurzusba bekerül. A kiképzést vizsga zárja. Aki a vizsgán nem megy át, azt alkalmatlanság címen azonnal leszerelik. Akit egyszer leszereltek, az ismételten nem jelentkezhets. A sikeres vizsgázókat felavatják, és ezzel nyugdíjazásukig vagy halálukig a parancsnokság kötelékébe tartoznak. A felavatottak bármikor bevetésre vezényelhetők. A parancsnokság nagy gondot fordít a személyzet folyamatos továbbképzésére az űrharcászat területén. A továbbképző tanfolyamokat a normális szolgálati rend megtartása mellett, a bevetésekkel párhuzamosan tartják. Egy ember egyidejűleg több tanfolyamra is járhat. A tanfolyam végén vizsgát kell tenni. A sikertelen vizsga korlátlanul ismételhető. Aki három tanfolyamot sikeresen elvégez és a felavatás, illetve a legutóbbi előléptetés óta volt legalább 20 bevetése, azt ismételten előléptetik.

Definiálja az alábbi funkciókat ellátó processzeket:

- Kérésre adja meg azon bevetéseket, amelyekhez még nem rendeltek személyzetet.
- Kérésre léptesse elő a feltételeknek megfelelőket.
- Kérésre adja meg a továbbképzésre járó, de éppen bevetésre vezényeltek listáját.
- Kérésre adja meg azok listáját, akik valamely tanfolyamon második kísérletre sem tudtak sikeresen levizsgázni.

Készítse el az **entitás/akció leírást, az entitások attribútumaival, az entitás struktúra diagramot, az initial modelt,** kapcsolja a modellhez a **funkció processzeket.**





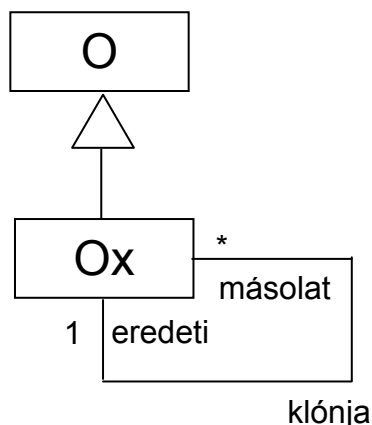
Legyen egy tetszőleges O osztály. Ennek egy olyan Ox változatát kell elkészíteni, amelyik rendelkezik O valamennyi tulajdonságával + egy p példánya a $clone()$ metódusának meghívásakor készít magáról egy m másolatot, amely folyamatosan követi p állapotának minden változását. p -ről több másolat is készíthető, de m már nem másolható tovább. Egy m másolaton egy kliens részéről kezdeményezett olyan műveletet, amely m állapotának megváltozását okozná, m továbbítja p -nek, aki a műveletet saját magán hajtja végre, hogy valamennyi másolata kövesse a változtatást.

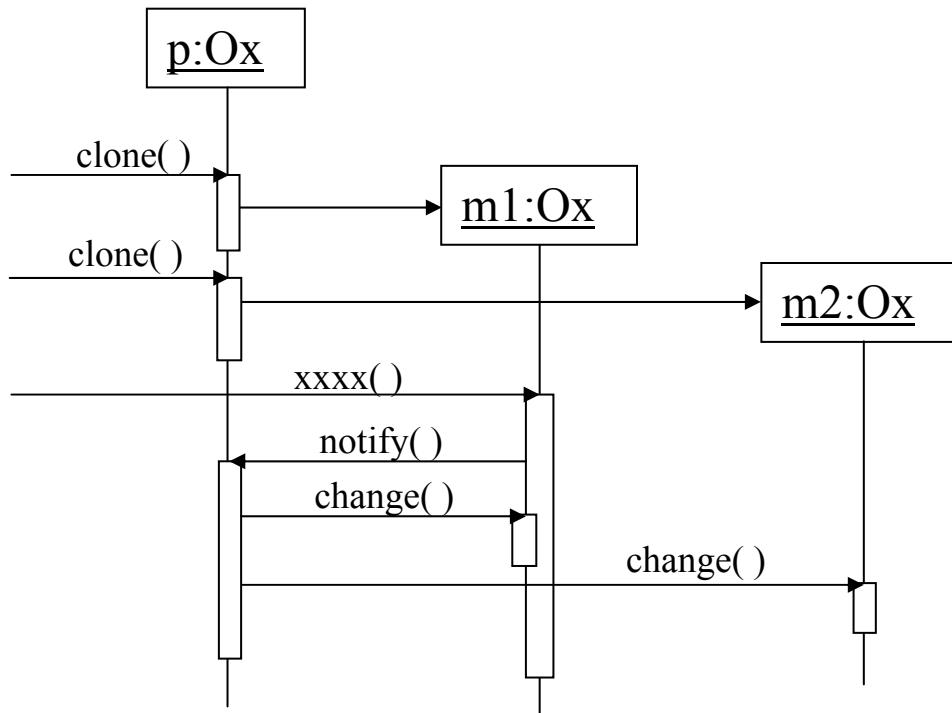
Adja meg Ox attribútumait.

Rajzoljon szekvencia diagramot arra az esetre, amikor p -ről egy kliens készít két másolatot ($m1$, $m2$), majd az $m1$ -nek meghívja egy olyan metódusát ($xxxx()$), amely $m1$ -nek módosulását okozná.

Ox attribútumai:

klónozzható





PT vizsga 1995.01.05. megoldásai --részpontokkal--

Ezt az egészet a Hotta papírja alapján írtuk be változtatás nélkül, őt verjétek....

1.

```

lgth crt()=0
lgth add(s,ch)=lgth(s)+1
lgth head(s,i)=if i>lgth(s) then lgth(s) else i
in crt(),i= értelmetlen
in add(s,ch),i= if i=lgth(s)+1 then ch           --0.5--
                  if i<lgth(s)+1 then in(s,i)    --1--
                  if i>lgth(s)+1 then értelmetlen --0.5--
in head(s,i1),i2)=if i2>i1 then értelmetlen    --0.5--
                    else in(s,i2)                --1--

```

```

head crt(),i)=crt()
head add(s,ch),i)=if lgth(s)+1 > i then head (s,i)
                  if lgth(s)+1 <=i then add(s,ch)

```

2.

	NEM LÁTHATÓ	MONDATVÉG	SORVÉG	EGYÉB
SZÜNET	SZÜNET	<u>(ROSSZ)</u>	SZÜNET++	MONDAT
MONDAT	MONDAT	MUTÁN	ROSSZ++	MONDAT
MUTÁN	MUTÁN	<u>(ROSSZ)</u>	SZÜNET++, ÍR	ROSSZ
ROSSZ	ROSSZ	KILÉP	ROSSZ++	ROSSZ
KILÉP	KILÉP	ROSSZ	SZÜNET++	ROSSZ

--oszlop 1p, minden jó állapot 1p, nincs redundáns állapot 1p--

3. ???

4.

1: WRITE SOR

2: SOR++

3: SOR = 0

4: READCH

Jó rész: 4 pont

Rossz rész/eleje: 1 pont

Rossz rész/többi: 2 pont

Output: 1 pont

Utasítások: 1 pont, elhelyezésük: 1 pont

5.



--2PONT--



KÁRTYAOLV.

KÁRTYA KEZELŐ

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

1995. január 12.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett zsákot (bag, multiset)! (6 pont)

NEW()	új zsákot hoz létre.
STO(b, item)	item -et a b zsákba rakja.
ERA(b, item)	item egy példányát kiveszi b -ből. Ha b -ben nem volt item , b változatlan marad.
DUP(b, item)	item előfordulásainak száma b -ben.
DIF(b)	az egymástól különböző értékek száma b -ben.

2. Specifikálja kiterjesztett véges állapotú automatával (EFSM) azt a programot, amely egy szövegfájlban megszámolja az **elválasztott szavakat** ! Szónak nevezzük a betűkből és kötőjelekből álló karaktorsorozatot. Minden más karakter szóhatároló. Kötőjelet csak összetett szó képzéséhez és elválasztáshoz használunk, ezen kívül kötőjel a szövegben nem áll. Összetett szavakban a kötőjel csak betűk között állhat. Elválasztott a szó, ha a kötőjelet közvetlenül sorvége követi, majd a következő sorban betűvel folytatódik. Lehetnek többszörösen összetett szavak is, de nincs több soron áthúzódó elválasztott szó. (7 pont)

Például : egymillió-kétszázi-

zenhármezer-negyven

egymillió-kétszázi-

zenhá-

romezer-negyven

elválasztott többszörösen összetett szó, 1 találat

nem fordul elő

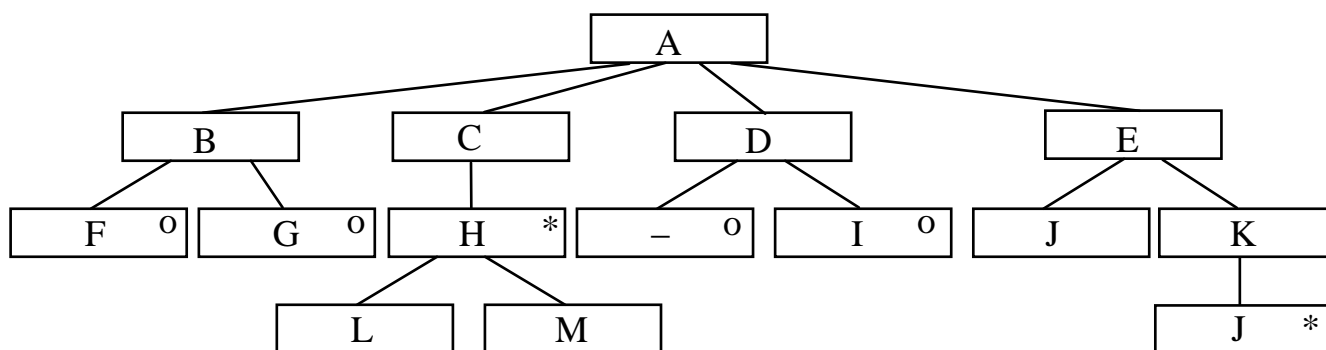
3. A COCOMO költségbecslési módszer milyen jellemzők között definiál kapcsolatokat ? Hogyan terjesztik ki a modellt a karbantartási költségek becslésére ?(3 pont)

4. JSP szerint tervezze meg a 2. pontban specifikált programot ! (10 pont)

5. Készítse el egy pénzjegykiadó automata adatfolyamábráját ! (5 pont)

Szerte a világon egyre több helyen üzemel olyan automata, amelyből a VISA-kártya tulajdonosok az adott ország pénznemében pénzt vehetnek fel az otthoni bankszámlájukról. A VISA-kártya egy mágnescsíkkal ellátott bankkártya (hasonló az AGROBANK kártyához). A pénzjegykiadó automaták (ATM) ún. OFF-LINE üzemmódban működnek, azaz nincs állandó kapcsolatuk a kártyatulajdonosok számlavezető bankjainak számítógépeivel. A kártyatulajdonos a kártyát az ATM-be helyezve és a titkos PIN-kódot begépelve pénzt vehet fel. A tranzakcióról az ATM nyomtatott bizonylatot ad a kártyatulajdonosnak. Ugyanazon kártyával a VISA számláról egy napon felvehető maximális összeg limitálva van. Az ATM a tranzakciók adatait tárolja és naponta egyszer elküldi az ATM-et üzemeltető helyi banknak. A bank naponta egyszer az érvénytelen/letiltott kártyák adatait elküldi az ATM-nek. Az érvénytelen kártyákat az ATM felismeri és bevonja. Az ATM rendelkezik egy szervíz felülettel amelyen keresztül a helyszínen a karbantartók kiolvashatják és beállíthatják az adatokat.

6. Rajzolja meg az alábbi JSD entitás-struktúra diagram alapján a neki megfelelő állapotgráfot ! (2 pont)



7. Adjon JSD alapú megoldást az alábbi problémára ! (13 pont)

A GÁZ GÉZA tudományos egyesület középiskolai tanulók számára különféle tanulmányi versenyeket szervez, matematika, fizika, informatika tárgyban, több korcsoportban. Az egyesület által meghirdetett versenyekre az iskolák a várhatóan induló diákok számának megadásával jelentkeznek. Az egyesület az egyes versenyek előtt 12 nappal a sokszorosítótól kapott feladatsorokat postázzák az iskoláknak, *ajánlott* és *bizalmas* megjelöléssel. A csomagban a feladatsorok mellett dolgozatkísérő lapokat is küldenek. Az iskolákban a csomagokat csak a verseny kezdetekor szabad felbontani. A dolgozatkísérő lapok egyes rovatait (név, osztály, laccím stb.) a diákok töltik ki, más rovatokat (iskola, pontszámok stb.) az iskolai javító tanárok. A legalább 40 pontos dolgozatokat, a kísérő lappal együtt a megadott határidőre az iskolák elküldik az egyesületnek. Az egyesületben a javításokat átnézik, szükség szerint módosítják és a dolgozatok eredményéről április közepéig írásban értesítik az iskolákat. Ha az iskola vitatja az egyesület által módosított pontszámok jogosságát, az értesítés kézhezvétele után 5 napon belül óvást emelhet. A határidőig beérkezett óváásokat az egyesületben egy héten belül elbírálják és május 15.-én a szakma neves képviselőinek jelenlétében eredményt hirdetnek, melyen csak az első 10 helyezettet nevezik meg. A dolgozatokat és a kísérőlapokat az eredményhirdetést követően ünnepélyesen elégetik.

Definiálja az alábbi funkciókat ellátó processzeket:

- Azonnal jelezzen, ha az egyesület olyan iskolától kap dolgozatot, amelyik nem jelentkezett a versenyre.
- Kérésre adja meg azoknak az iskoláknak a nevét, amelyek diákjai az utolsó három évben legalább kétszer értek el dobogós helyezést.
- Kérésre adja meg, hogy egy folyamatban levő versenyre eddig hány iskolából hány dolgozat érkezett.

Készítse el az

**entitás/akció leírást, az entitások attribútumaival,
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a funkció processzeket.**

8. Készítse el az alábbi feladat objektum modelljét ! Az objektum diagram a szövegben közöltekén kívül egyéb attribútumot és relációt ne tartalmazzon ! Az attribútumok jelölésére a zárójelbe tett megnevezéseket használja! Az adatmodellek feleljenek meg a 3NF szabályainak. (4 pont)

Az XYZ egyetem hallgatói a félév során egyetemi oktatók által tanított tárgyakat vesznek fel, tanulnak és tesznek belőle vizsgákat. A hallgatót jellemzi egyedi azonosító kódja (HKOD), a hallgató neve (HNÉV), és a hallgató által az előző félévben elért tanulmányi átlag (ÁTLAG). Egy hallgató több tárgyat vehet fel, és természetesen egy tárgyat több hallgató is látogathat. A tantárgynak van megnevezése (TNÉV) és száma (TSZÁM), ez utóbbi a tárgyat egyértelműen azonosítja. Egy tárgyat egyetlen oktató (ONÉV) tanít, akit személyi azonosítója (OAZON) jellemez. Egy oktató több tárgyat is taníthat. A hallgató az általa felvett tárgyakból tárgyanként egy alkalommal vizsgát tesz, amelynek eredménye (VJEGY) beszámít az átlagba.

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

1995. január 5.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett karakterstringet (6 pont)

CRT()	új (üres) stringet hoz létre.
ADD(s,ch)	az s string végére ch karaktert tesz.
HEAD(s,i)	az s stringet i hosszúra csonkolja (az első i karakterét veszi).
IN(s,i)	megadja az s string i -ik karakterét.
LGTH(s)	egész, az s string hossza.

2. Specifikálja kiterjesztett véges állapotú automatával (EFSM) azt a programot, amely egy szövegfájlt beolvassa visszaírja mindazon sorok sorszámát, amely sorokban pontosan egyetlen mondat áll. A mondat látható karakterrel kezdődik és mondat végjellel (pont, kérdő- vagy felkiáltó jel) zárul. A mondatok között tetszőleges számú nem látható (szóköz és tabulátor) karakter állhat, amelyeket a sor tartalmának vizsgálatakor figyelmen kívül kell hagyni. Mondat végjel csakis és kizárólag mondat végén áll. A sort sorvégjel zárja és a fájl tekintse végtelen hosszúnak. (7 pont)

3. Milyen modelleket, és milyen diagramokat használ az OMT módszertan ? (3 pont)

4. JSP szerint tervezze meg a 2. pontban specifikált programot ! (10 pont)

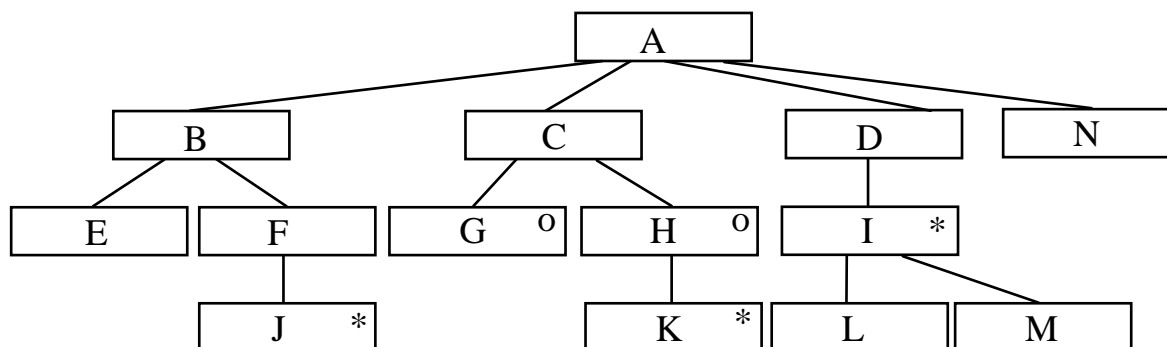
5. Készítse el egy automatikus metró kijárat automata kapu vezérlőjének adatfolyamábráját ! (6 pont)

Egy nagyváros metróállomásai automatikus jegy/bérletkezelő és ellenőrző kapukkal vannak felszerelve. A jegyeket és a bérleteket mind a kijáratnál mind a bejáratnál az ellenőrző kapun való áthaladáskor egy olvasóba kell dugni amely leolvassa a jegyen/bérleten lévő mágnescsíkon rögzített információkat. A bejárat automata, amennyiben a jegy/bérlet érvényes felírja a mágnescsíkra a belépés időpontját és az állomás kódját. A kijárat automata ellenőrzi a jegy/bérlet érvényességét, és ha mindent rendben talál kinyitja a kaput. Ha bármilyen rendellenességet észlel, a kapu zárva marad, és a készülék szirénázni kezd. A készülék a következő esetekben jelez:

- A jegy/bérlet érvénytelen.
- A jeggyel/bérlettel a megengedettnél többet utaztak (tarifa határ átlépés)
- Ugyanezzel a bérlettel 10 percen belül másodszor kísérelnek meg kilépni az állomásról.

Kilépés után az automata a jegyet nem adja vissza. Az érvénytelen vagy elhasznált jegyeket és az érvénytelen bérleteket az automata egy belső tartályba gyűjti, melyet a kezelőszemélyzet naponta ürít. A szerelvények közlekedtetésének optimalizálásához minden automata statisztikát készít a rajta áthaladt utasokról az utazási paraméterek (idő, időtartam, kiindulási állomás) nyilvántartásával. Ezeket az adatokat óránként egy interfészen keresztül a forgalomirányító központba továbbítják.

6. Rajzolja meg az alábbi JSD entitás-struktúra diagram alapján a neki megfelelő állapotgráfot ! (2 pont)



7. Adjon JSD alapú megoldást a tankönyvkiadók adminisztrációs rendszerének kialakítására ! (12 pont)

Az iskolák a tanulók részére a tankönyveket kiadóktól vásárolják meg. Az iskolák a kiadóknak megrendeléseket küldenek. Egy iskola több kiadótól rendel, egyetlen kiadótól akár többször is. A kiadók a megrendelés elfogadását vagy elutasítását az iskoláknak visszajelzik. A kiadó az iskolának szánt könyveket - azonnal, amint azok a nyomdából megérkeznek, mert nem tud raktározni - maximum 20 kg-os csomagokra bontva postán küldi el. Minden egyes csomaggal, tartalmának megfelelően számlát küldenek. Az iskolák az egyes számlákra akár részletekben is fizethetnek. De megvárhatják az egy megrendeléshez tartozó **összes** számla beérkezését, azokat kifizetetlenül visszaküldve a kiadónak kérhetik, hogy a csomagonkénti számlákat sztornózzák és küldjenek egy, a megrendelés egészére vonatkozó számlát. Ezt a számlát az iskolának már egyösszegben ki kell fizetnie.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha olyan iskolától érkezik megrendelés, amelyeknek két hónapnál régebbi kiegyenlített számlája van !

Kérésre adja meg azon iskolákat, amelyekhez küldött mindegyik csomag értéke 5000 Ft felett volt !

Kérésre sorolja fel azon elfogadott megrendeléseket, amelyekre még nem küldtek csomagot !

Készítse el az **entitás/akció leírást, az entitások attribútumaival,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

8. Készítse el az alábbi feladat objektum modelljét ! Az objektum diagram a szövegben közöltekén kívül egyéb attribútumot és relációt ne tartalmazzon ! Az attribútumok jelölésére a zárójelbe tett megnevezéseket használja ! Az adatmodellek feleljenek meg a TNF szabályainak. (4 pont)

Harácsfalva nagyközség önkormányzata úgy határozott, hogy az autókra kivetett adót kiterjesztik a biciklikre is. Ennek kapcsán nyilvántartják a bicikliket és azok gazdáit. A biciklit egyértelműen azonosítja az alvázszáma (BALSZ), de szükséges ismerni a bicikli típusát (BIT), mivel az adó a típustól függ. A faluban sok azonos típusú bicikli van, ami időnként konfliktusok forrását jelenti. A biciklik gazdáinak jellemzői a gazda neve (GAZDA), címe (CÍM), személyi igazolvány száma (SZEM). Harácsfalva lakossága a bicikliket rendszeresen adja-veszi, cseréli, kidobja, ezért az időarányos adózás miatt a tulajdonlásban beálló változást és annak dátumát (DÁT) be kell jelenteni. Mivel a harácsfalviak nem üzérkednek, feltételezhető, hogy egy bicikli egy napon egyszer cserél gazdát. Egy gazdának több biciklije is lehet, de közös tulajdonú bicikli nincs. Az önkormányzat az adó mértékét (ADÓ) bicikli típusonként évente rendeletben szabályozza.

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

1994. január 26.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett konténert (5 pont)

CRT	új, üres konténert hoz létre.
PUT(k,item)	k -ba item -et tesz.
HEAD(k)	megmutatja a k -ban található legkisebb item -et.
TAIL(k)	k -ból eltávolítja annak legkisebb item-jét.

2. Specifikálja kiterjesztett véges állapotú automatával (EFSM) azt a programot, amely egy szövegfájlból kiírja és megszámlolja azon szavakat, amelyek a sorban utolsók, magánhangzóval kezdődnek és több bennük a mássalhangzó, mint a magánhangzó! A fájl beolvasását követően írja ki a feltételeknek megfelelő szavak számát! Szónak tekintendő a betű karakterekből álló minimum egy hosszú sorozat. Két szó között tetszőleges számú és minőségű szeparátor állhat. Szeparátor minden nem betű karakter. Nem üres sor vége előtt biztosan áll legalább egy szeparátor. (7 pont)

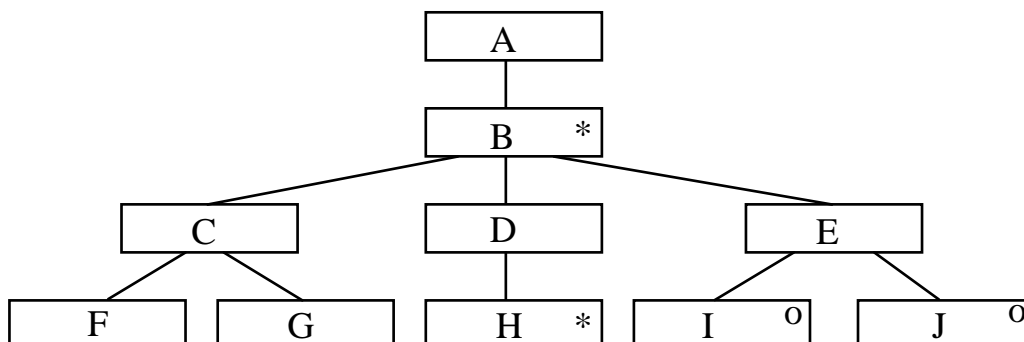
3. Adja meg, hogy egy szoftver létrehozása során mikor, milyen dokumentációt kell elkészíteni ! (4 pont)

4. JSP szerint tervezze meg 2. pont szerinti programot (7 pont)

5. Készítse el egy számlaadó taxióra adatfolyamábráját ! (7 pont)

A taxióra a taxi gépkocsi használatával összefüggő adatokat gyűjti, tárolja, jelzi és nyomtatja. A taxióra bemenetei a kerék egy-egy fordulatát jelentő impulzusok és az órán található kezelő szervek. Ez utóbbiak egyikével lehet a négy különböző tarifa közül választani. Egy tarifa tartalmazza az alapdíjat, a kilométerenkénti díjat és az egyperces várakozás díját. A legfontosabb kezelőszerv a szabad-foglalt kapcsoló. Foglaltra kapcsoláskor a beállított tarifának megfelelő alapdíjat mutatja a készülék, majd folyamatosan kijelzi a díjat a megtett útnak illetve várakozási időnek megfelelően. Szabadra kapcsoláskor a taxióra számlát nyomtat, amely tartalmazza a tarifa számát, a megtett kilométert, a várakozást, a szolgáltatás díját ÁFÁ-val és anélkül, valamint az olyan alapadatokat, mint a dátum, a kocsi rendszáma, a szolgáltató neve, adószáma. Szabad módban a készülék az üresen megtett kilométereket jelzi. Pontban éjfélkor az elmúlt nap integrált adatai letárolódnak Szabad módban egy gomb megnyomásával lehetséges az aktuális és az előző három nap integrált adatainak kinyomtatása. A leplombált készülék belsejében elhelyezett csatlakozón keresztül - az arra jogosítottak - valamennyi tárolt adatot lekérhetik, törölhetik és az alapadatokat beállíthatják.

6. Rajzolja meg az alábbi JSD entitás-struktúra diagram alapján a neki megfelelő állapotgráfot ! (2 pont)



7. Adjon JSD alapú megoldást az alábbi problémára ! (14 pont)

A könyvtár tagjai részére könyveket kölcsönöz. A belépési nyilatkozat kitöltésével bárki tag lehet. A tag akkor jogosult kölcsönözni, ha befizeti a tagsági díjat. Ennek fejében a díj befizetésének napjától számított egy évig igényelheti a könyvtár szolgáltatásait. A tagsági díj ismételt befizetésével a jogosultság fenntartható vagy ismét megszerezhető. Egyidőben egy tag maximum 5 könyvet kölcsönözhet.

A könyvtár a könyveket a kiadóktól vásárolja; egy műből esetenként több példányt is. A könyveket életük során legföljebb egyetlen alkalommal újra kötik a könyvkötészetben. A tönkrement vagy érdektelen könyveket leselejtezik.

A könyvtárban lehetséges az előjegyzés. Ez azt jelenti, hogy a tag bejelentheti kölcsönzési szándékát olyan műre, amelyikből a szándékolt kölcsönzés idejében egyetlen példány sincs bent a könyvtárban (még nincs ilyen, olvasónál vagy a kötetben vannak). Ha a könyv az előjegyző számára rendelkezésre áll, akkor arról levélben értesítik. Ha megadott ideig a tag nem jelentkezik, akkor a könyvet mások számára felszabadítják. Verseny esetén a döntés a könyvtáros kezében van.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha olyan tag kíván kölcsönözni, akinél esedékes a tagsági díj befizetése.
Kérésre adja meg az adott című műre fennálló előjegyzések számát és a legrégebbi előjegyzés adatait.
Kérésre adja meg azokat a könyveket, amelyeket az elmúlt fél évben egyszer sem kölcsönöztek.

Készítse el az **entitás/akció leírást, az entitások attribútumaival,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

8. Mik a CASE rendszerek fontosabb komponensei ? (4 pont)

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

1994. január 12.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett karakterstringet ! (4 pont)

CRT()	új (üres) stringet hoz létre.
ADD(s,ch)	az s string végére ch karaktert tesz.
TAIL(s)	az s string első karakterének levágása után maradó stringet adja meg.
IN(s,i)	megadja az s string i -ik karakterét.
LGTH(s)	egész, az s string hossza.

2. Specifikálja kiterjesztett véges állapotú automatával (EFSM) azt a programot, amely egy szövegfájlból kiír minden olyan mondatot, amelynek második szava két mássalhangzóval kezdődik ! A fájl vége csak mondat végjel után következhet. Mondatot határol a pont, a felkiáltójel és a kérdőjel. Szónak tekintendő a betű karakterekből álló minimum egy hosszú sorozat. Két szó között tetszőleges számú és minőségű szeparátor állhat. Szeparátor minden nem betű és nem mondat végjel karakter. A mondat végjel előtt biztosan áll egy szeparátor.

(8 pont)

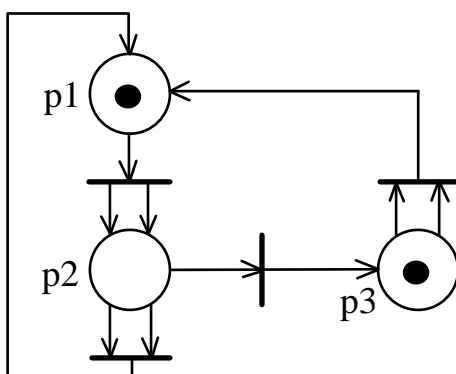
3. Adja meg, hogy egy szoftver létrehozása során mikor, milyen dokumentációt kell elkészíteni ! (4 pont)

4. JSP szerint tervezze meg 2. pont szerinti programot (8 pont)

5. Készítse el egy automatikus mérleg adatfolyamábráját ! (8 pont)

A mérleget a zöldségesnél használják az alábbi módon. A vásárló az általa kiválogatott - egyfajta - gyümölcsöt vagy zöldséget egy átlátszó zacskóba szedi. A zacskót a mérlegre teszi, ahol leolvashatja a súlyt. A mérleghez kapcsolt vezérlő táblán megnyomja a zacskóba szedett áru képével ellátott gombot, mire a mérleg egy címkét nyomtat, amelyen feltünteti a vásárolt áru megnevezését, súlyát és árát, valamint a vásárlás idejét. Az öntapadós címkét a zacskóra kell ragasztani. A mérleg belsejében elhelyezett - csak a személyzet számára hozzáférhető - billentyűzet és kijelző segítségével a vezérlő tábla gombjaihoz hozzárendelhetők az alapadatok, (az áru megnevezése, egységára), beállítható a dátum és az idő valamint a címkén szereplő állandó szövegek. Ugyanitt lekérdezhető az egyes árufajtákból kimért napi összes mennyiség.

6. Kiindulva az ábrán szereplő markerezésből adja meg az alább definiált Petri-háló elérhetőségi gráfját! (3 pont)



7. Adjon JSD alapú megoldást az alábbi problémára ! (12 pont)

A Zeneakadémia végzős növendékei szeretnének elhelyezkedni. A végzettek közül vagy zenetanár vagy zenekari tag lehet. A növendékek több állásra is pályázhatnak. A zenekari állásokat zenekarok, a zenetanári állásokat zeneiskolák hirdetik meg. Mindkét esetben a jelölt egy részletes önéletrajz benyújtásával pályázik. A felvételt szervező bizottságok a nem megfelelő pályázatokat " *az állást már betöltöttük* " címen visszautasítják. Az elfogadott pályázatokkal kapcsolatos felvételi eljárás különbözik zenetanári és zenekari állás esetén. A zeneiskola beszélgetésre, majd egy alkalommal próbatanításra hívja be a jelentkezőket. A zenekarban két fordulóban próbajátékot hirdetnek. Az első forduló után a felvételi bizottság a gyengéket a rostan kiszűri (elutasítja), második próbajátékra már nem is hívja. Egy állás betöltésére végül a bizottságok egyetlen jelöltnek tesznek ajánlatot. A növendékek a kapott ajánlatok közül egyéni szempontjaik szerint választanak.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha ugyanaz a zenekar a növendékek közül már legalább egynek tett ajánlatot !
Kérésre adja meg azon növendékeket, akiknek legalább három helyre fogadták el a pályázatukat !
Kérésre adja meg azon zeneiskolákat, ahol csak egyetlen hangszer oktatására írtak ki pályázato(ka)t !

Készítse el az **entitás/akció leírást, az entitások attribútumaival,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

8. Mi a feladata és milyen módszereket használ a JSD *system timing* és *implementation* lépése ?(3 pont)

VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

1994. január 5.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett karakterstringet ! (8 pont)

NEW()	új (üres) stringet hoz létre.
ADD(s,ch)	az s string végére ch karaktert tesz.
CONCAT(s,s)	az első s stringet a második s string konkatenálásával bővíti.
FIRST(s)	megadja az s string elején álló karaktert.
REMSTR(s)	az s string első karakterének levágása után maradó stringet adja meg.
LGTH(s)	egész, az s string hossza.

2. Specifikálja kiterjesztett véges állapotú automatával (EFSM) azt a programot, amely egy szövegfájlt szótagolva ír vissza és minden sor végén megadja a sorban álló szótagok számát, valamint a végén az összes szótagok számát! A szó-ta-go-lós ki-í-rás-ban kö-tô-je-lek vá-laszt-ják el a szó-ta-go-kat. A szótagolás szabályai az alábbiak:

- Egy szótagban pontosan egy magánhangzó állhat.
- A szót kezdő szótagot leszámítva a magánhangzó előtt csak egyetlen mássalhangzó állhat.

Ne foglalkozzon a kettős betűkkel (sz, gy, .. stb) és a kivételekkel !

Szónak tekintendô a betû karakterekbôl álló minimum egy hosszú sorozat. Két szó között tetszôleges számú és minôségû szeparátor állhat. Szeparátor minden nem betû karakter.(7 pont)

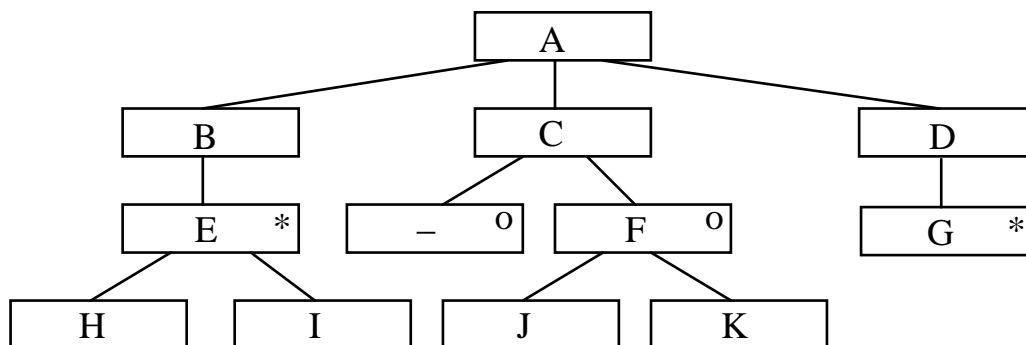
3. Mik a CASE rendszerek fontosabb komponensei ? (4 pont)

4. JSP szerint tervezze meg azt a programot, amely egy szövegfájlból kiírja mindazon szavakat, amelyek legalább 2 mássalhangzóra végzôdnek! A fájl beolvasását követôen írja ki azt is, hogy a szövegben hány - a feltételnek megfelelô - szó fordult elô! Szónak tekintendô a betû karakterekbôl álló minimum egy hosszú sorozat. Két szó között tetszôleges számú és minôségû szeparátor állhat. Szeparátor minden nem betû karakter. (9 pont)

5. Készítse el egy automatikus tengeri meteorológiai bôja adatfolyamábráját ! (5 pont)

A bóját a tengerre telepítik abból a célból, hogy idôjárás adatokat továbbítson rádiôn a Meteorológiai Intézetbe. A bôja megadott idôközönként megméri a levegô és a víz hôtérsekletét valamint a szél sebességét. Ugyanekkor leolvassa a benne elhelyezett Globális Pozíció Meghatározót (GPS) is, amelyik megadja a bôja pillanatnyi földrajzi koordinátáit A bôja óránként vagy a Meteorológiai Intézettôl rádiôn küldött kérésre az összegyûjtött, de még nem közölt mérési adatokat a rádiôn továbbítja .

6. Rajzolja meg az alábbi JSD entitás-struktúra diagram alapján a neki megfelelô állapotgráfot ! (2 pont)



7. Adjon JSD alapú megoldást az alábbi problémára ! (11 pont)

A Hadmérnöki Kar oktatói fakultatív tárgyakat hirdetnek meg. A tárgyakra a Kar hallgatói jelentkezhetnek a szorgalmi időszak kezdete előtt. Egy hallgató tetszőleges számú fakultatív tárgyat vehet föl; egy oktató több tárgyat is meghirdethet. Csak azokat a tárgyakat indítják el, amelyekre a jelentkezők száma minimálisan 10 volt. Valamennyi tárgyból hetente egy alkalommal 2 órás előadást és egy 2 órás gyakorlatot tartanak. A vizsgaidőszakban a tárgyakból különböző számú, de egynél több vizsgát tartanak.

Definiálja az alábbi funkciókat ellátó processzeket:

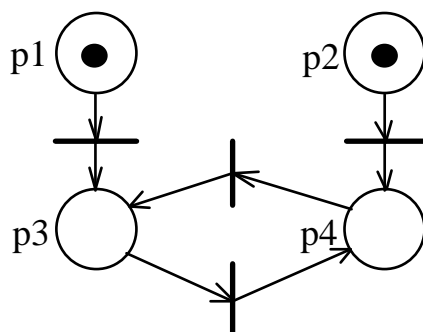
A 10-ik hallgató jelentkezésekor írja ki a tárgy nevét!

Kérésre adja meg azon oktatókat, akiknek több tárgyuk is indult!

Kérésre adja meg azon hallgatókat, akiknek még van hátra vizsgájuk!

Készítse el az **entitás/akció leírást, az entitások attribútumaival,**
az entitás struktúra diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

8. Kiindulva az ábrán szereplő markerezésből adja meg az alább definiált Petri-háló elérhetőségi gráfját!
(4 pont)



VIZSGAKÉRDÉSEK
PROGRAMOZÁS TECHNOLÓGIÁJA
c. tárgyból

1993. december 22.

1. Algebrai axiómák segítségével specifikálja az alábbi műveletekkel jellemzett halmazt ! (3 pont)

NEW()	új halmazt hoz létre.
PUT(s,item)	item-et az s halmazba teszi.
SUB(s,item)	item-et elveszi s-ből. Ha item nem volt s-ben, akkor s változatlan.
IN(s,item)	igaz, ha item eleme s-nek.
LGTH(s)	egész, az s halmaz elemeinek száma.

2. Specifikálja kiterjesztett véges állapotú automatával (EFSM) azt a programot, amely kiírja egy szövegfájlból mindazon sorokat, amelyekben legalább egy oktális szám előfordult! Oktális (8-as számrendszerbeli) szám a 0-7 jegyekből álló és 'o' vagy 'O' (kis o vagy nagy O) betűkre végződő, szeparátorokkal határolt karaktorsorozat . Szeparátor minden nem betű vagy szám karakter. Tetszőleges számú szeparátor állhat egymás után.(8 pont)

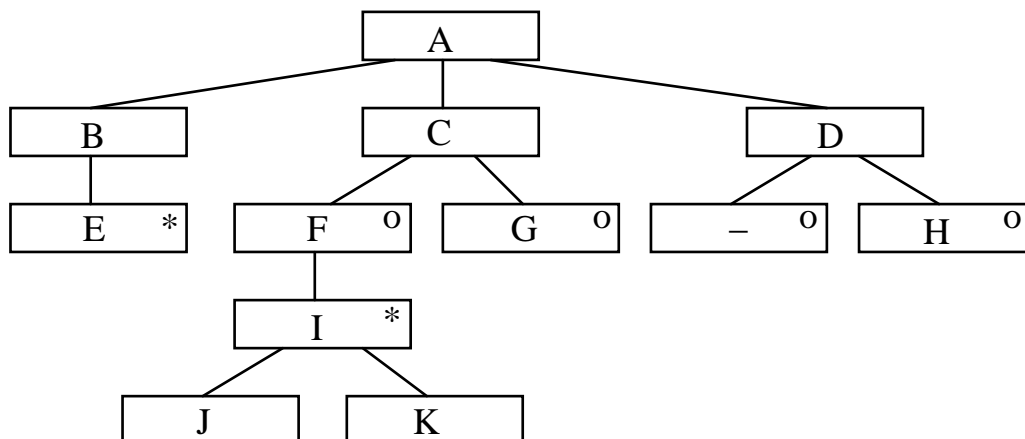
3. Az egyszerűsített COCOMO modell milyen mennyiségek között állapít meg összefüggést ? A kiegészített modell milyen jellemzőket vesz figyelembe ? (3 pont)

4. JSP szerint tervezze meg a 2. pontban adott feladatot! Rajzolja fel az input, az output és a program struktúra ábráját ! A programstruktúrán jelölje be a végrehajtható utasításokat! (10 pont)

5. Készítse el egy mozi jegyárúsításának adatfolyamábráját ! (10 pont)

A mozi előadásait az előadás jellemzőinek (film címe, előadás időpontja, helyárak) megadásával a mozi vezetője definiálja. A pénztáros a definiált és még be nem fejezett előadásokra készpénz fizetése ellenében jegyet ad, amely egyben számla is. A jegyen rajta van a mozinak az adózással kapcsolatos adatain kívül a film címe, az előadás időpontja, az elfoglalt székek azonosítója (erkély, bal, 2. sor 2-5 szék) és a jegy ára. Egy jeggyel több széket is foglalhatunk. A pénztárosnál telefonon előjegyezhetünk jegyet.

6. Rajzolja meg az alábbi JSD entitás-struktúra diagram alapján a neki megfelelő állapotgráfot ! (3 pont)



7. Adjon JSD alapú megoldást az alábbi problémára ! (10 pont)

Az XXX cég megrendelésre, mérték után készít konyhabútort. Az érdeklődők a katalógusban megadott modellek közül választhatnak, de a méreteket nekik kell megadni. A megrendelés a pontos adatokat tartalmazó megrendelőlap kitöltésével és az előleg (előzetesen kalkulált ár 35 %-a) befizetésével történik. Mindaddig, amíg a gyártás nem kezdődik meg, a megrendelő a befizetett előleg 20%-ának hátrahagyásával a megrendelést visszamondhatja. A bútorhoz szükséges üveg és fém alkatrészeket az XXX cég nem maga készíti, hanem alvállalkozókkal gyártatja le. A cég raktárkészletet nem tart, minden megrendeléséhez egyedileg készített el az üveg- és fémszerelvényeket, gyakran egyazon megrendeléshez tartozó részeket különböző alvállalkozóknál. Mikor valamennyi alkatrész beérkezett, a közben legyártott bútorelemeket összeszerelik, majd kiszállítják és a helyszínen összerakják. Ekkor fizet a megrendelő. A cég vezetője jó kapcsolatot tart a vásárlóival, a vásárlást követő néhány évben Karácsonyi üdvözlőt is küld nekik.

Definiálja az alábbi funkciókat ellátó processzeket!

Kérésre adja meg azon alvállalkozókat, akik még nem szállították le a megrendelt üveg- és fémszerelvényeket..

Kérésre adja meg, hogy mely megrendelések várnak kiszállításra.

Kérésre törölje azokat a vásárlókat a névjegyzékből, akik 5 éve vásároltak bútort utoljára.

Készítse el az **entity/action leírást,**
az entity structure diagramot,
az initial modelt,
kapcsolja a modellhez a **funkció processzeket.**

8. Adja meg az irányítatlan gráf objektum modelljét a modellben szereplő objektumok (osztályok) közötti relációk definiálásával! A modell csak a gráf struktúráját (kapcsolatok) tartalmazza. (3 pont)

VIZSGAKÉRDÉSEK

PROGRAMOZÁS TECHNOLÓGIÁJA C. tárgyból

1999. december 21.

1. Algebrai axiómák segítségével specifikáljazz alábbi műveletekkel jellemzett stringet! (6 pont)

CRT()	új (üres) stringet hoz létre
LGTH(s)	az s string karaktereinek számát adja
TAIL(s)	az s string első karakterének levágása után maradó stringet adja
ADD(s,x)	az s string végére rakja az x karaktert
HEAD(s)	az s string első karakterét mutatja meg
PALIN(s)	igaz, ha az s string palindróma

Egy string palindróma, ha az elejétől olvasva ugyanaz, mint visszefelé. Pl.: "görög", "abba"

2. Rajzolja fel a struktúra ábráját (JSP) annak a programnak, amelynek működését az alábbi állapottábla írja le. (6 pont) A számokkal jelölt végrehajtható utasításokat kapcsolja a struktúrához.

	szeparátor	betű	újsor
a	a/1	b/2	a/3
b	c/5	b/4	a/3
c	c/5	b/2	a/3

3. A tervezési elvek között bevezettük a kohézió és csatolás fogalmait. Mit jelentenek? Milyen fő típusaik vannak? (4 pont)

4. Készítse el egy elektronikus parkolóóra adatfolyamábráját! Adja meg a context diagramot és annak 0 szintű felbontását! (6 pont)

A parkolóóra digitálisan kijelzi azt az időtartamot, amíg az ügyfél a parkolóhelyen várakozhat. Ha az idő letelt, akkor további 20 percen keresztül azt mutatja, hogy a várakozó mennyivel lépte túl a kiválasztott időt (negatív percek). A parkolási idő két módon adható meg: pénz bedobásával vagy parkolókártyával. A pénz bedobásakor az időtartam a bedobott érme értékének megfelelően növekszik. Parkolókártya alkalmazása esetén az órára szerelt gomb minden egyes megnyomásakor az időtartam 10 perccel megnő, feltéve, hogy a kártyán erre még van fedezet. A parkolóór speciális kártya használatával lekérdezheti a napi forgalmat (felíródik a kártyájára), valamint hozzáférhet a gyűjtött pénzhez.

5. Adja meg és 1-1 mondattal jellemezze a fontosabb szoftver architektúrákat! (3 pont)

6. Adjon JSD alapú megoldást az alábbi problémára! (10 pont)

A **Macskanyelv** nyelviskola a fontosabb világnyelveket oktatja a tanulni vágyóknak féléves kurzusokban. A kurzusokat a félév kezdete előtt meghirdetik (nyelv, szint, intenzitás pl: spanyol; haladó, 3*2óra-hét), majd lezajlik a

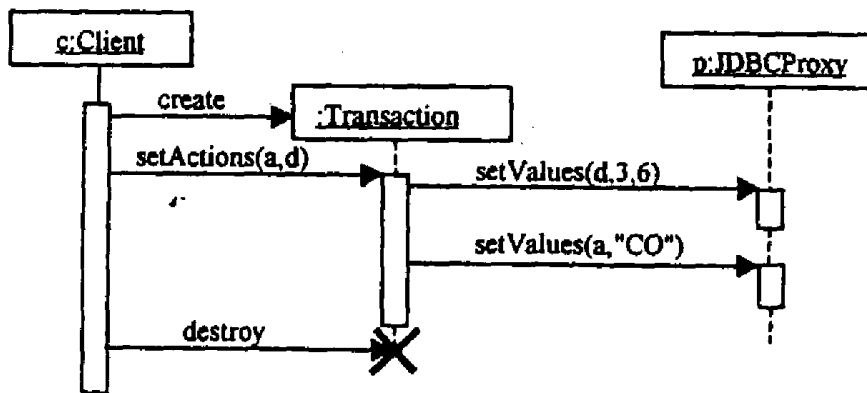
jelentkeztetés. Ennek során a jelentkező a tanfolyami díj befizetésével jelentkezik a kurzusra. Azok a jelentkezők, akik korábban már legalább két tanfolyamon vettek részt, kedvezményes díjat fizetnek. Egy kurzuson belül a jelentkezők számától függően több csoport is indulhat. A csoportok kialakításakor tisztázódik, hogy mikor és hol lesznek a csoport órái és ki lesz az oktató. Egy csoport létszáma maximum 12, de 5 főnél kisebb csoport nem indul. Előfordulhat, hogy kellő érdeklődés hiányában kurzusokat kénytelenek törölni. Az ide jelentkezőknek a díjat visszafizetik. Egy ember egy félévben több kurzusra is járhat párhuzamosan.

Definiálja az alábbi funkciókat ellátó processzeket:

Azonnal jelezzen, ha egy kedvezményes díjra jogosult ember akar befizetni egy kurzusra!
 Kérésre adja meg azon csoportokat, ahol volt olyan foglalkozás, amelyen csak egy ember jelent meg!
 Kérésre adja meg azokat az embereket, akik legalább három kurzusra járnak egyidejűleg!

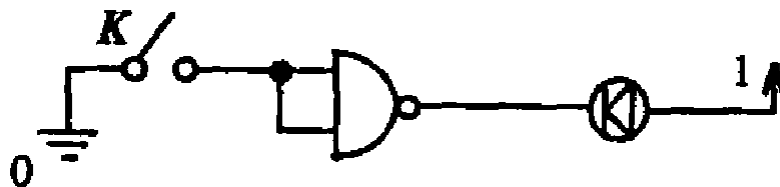
Készítse el az **entitás/akció leírást, az entítások attribútumaival,**
az entitás struktúra diagramot,
az initial modelt,
 kapcsolja a **funkció processzeket.**
 modellhez a

7. Rajzolja fel az alábbi szekvencia diagramnak megfelelő kollaborációs diagramot. (6 pont)



8. Definiáljon olyan osztályokat, amelyekből az alábbihoz hasonló NAND-kapus hálózatok modellje alkotható meg. A modell tartalmazzon NAND-kaput, kapcsolót, csomópontot, tápfesz és föld generátort, valamint LEDet. A modell alapján legyen lehetséges a LED-ek állapotának (világít, sötét) meghatározása. A modell elemei legyenek tetszőlegesen, de értelmesen összekapcsolhatók.

Rajzolja le az osztálydiagramot és adja meg az egyes osztályok metódusainak szignatúráit. (9 pont)



Eredmények értékelése:

Pontszám	Osztályzat
21-	2
28-	3
35-	4
42-	5

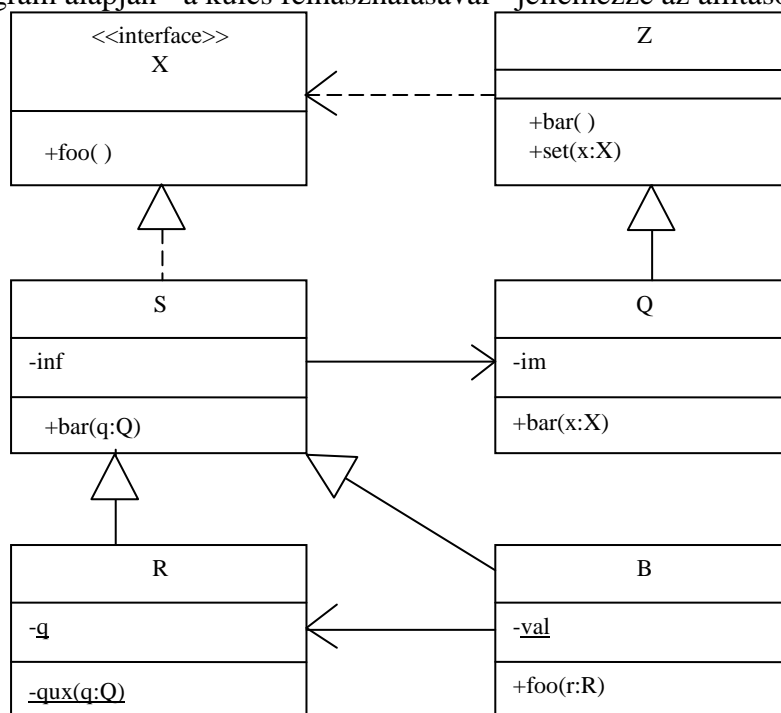
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2009. január 6. 14:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

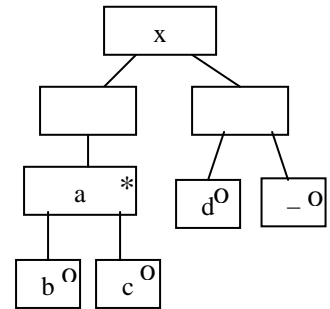
- [E] Q bárhol helyettesíthető S-sel, mert S a Q leszármazottja.
- [A] R qux(q:Q) metódusa nem kaphat paraméterül Z objektumot, mert a metódus absztrakt
- [E] B foo(r:R) metódusa nem hívhat meg a paraméterül kapott objektumon foo() metódust, mert R-nek nincs ilyen metódusa
- [E] S bar(q:Q) metódusa nem módosíthatja az S inf attribútumát, mert az attribútum konstans.
- [E] S bar(q:Q) metódusa nem hívhatja meg a paraméterül kapott objektum bar() metódusát, mert a Q osztálynak nincs ilyen szignatúrájú metódusa.
- [B] Z set(x:X) metódusa nem kaphat paraméterül B objektumot, mert B megvalósítja az X interfészt.
- [B] B módosíthatja egy Q objektum im attribútumát, mert S függ Q-tól.
- [B] R és B bárhol felcserélhetők, mert közös az ősük.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Egészítse ki az alábbi DTD vázat úgy, hogy az a mellékelt JSP struktúra szerinti adatszerkezetet definiálja ! A c elemnek legyen egy opcionális y attribútuma is ! (4 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
```

```
<!ELEMENT x (a*,d?)>
<!ELEMENT a (b|c)>
<!ATTLIST c y CDATA #IMPLIED>
```



```
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA)>
<!ELEMENT d (#PCDATA)>
```

]>

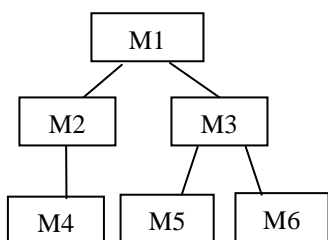
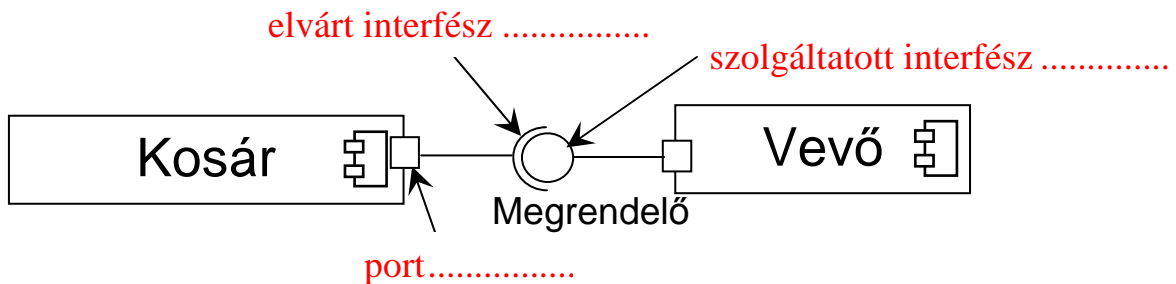
```
<x>
  <a> <c ></c> </a>
  <a> <c y="kedd"></c> </a>
  <d></d>
</x>
```

Készítsen az ábrának megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van két c elem és egy d elem is ! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

3. Mik a konfigurációs menedzsment fő folyamatai ? (4 pont)

- Storage Configuration Items
- Change Management
- Build Management
- Release Management

4. Az alábbi ábrán három UML2 modell elemet megjelöltünk. Adja meg elemenként, hogy az melyik UML2 meta-modell elem példánya ! (3 pont)



5. Egy szoftver struktúráját a baloldali ábra írja le. Legyen az M2 döntési hatáskörében M2, M3 és M4; M3 döntési hatáskörében M3, M4, M5, M6

Van-e döntés hasítás ? (1 pont) IGEN NEM

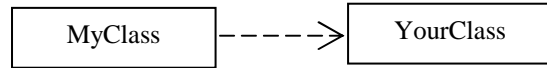
Mit kell tenni, hogy az ellenkező eredményt kapjuk ? (2 pont)

M2-ből M3-at és M3-ból M4-et elvenni

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

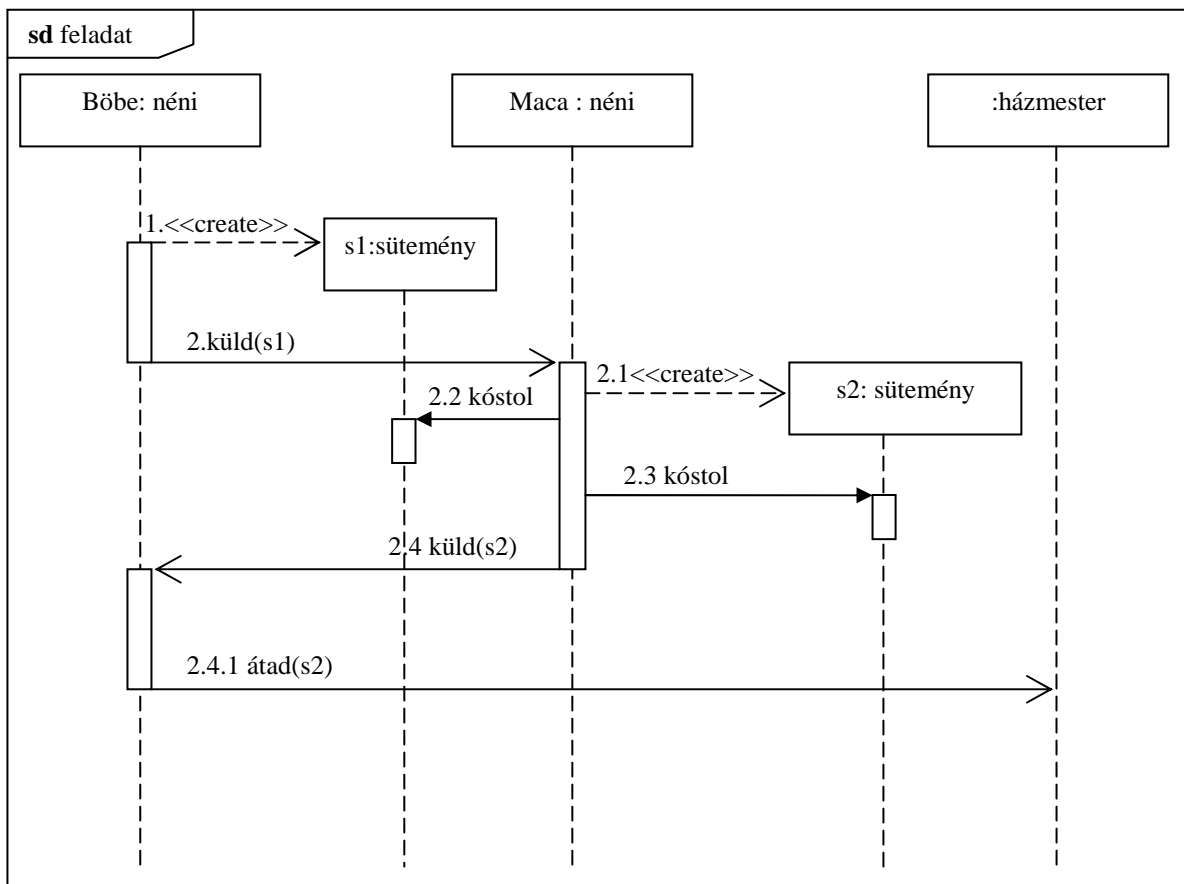
6. Rajzoljon UML2 osztálydiagramot az alábbi Java programrészlet alapján ! Az osztálydiagram ne tartalmazzon olyasmit, ami a kódból nem olvasható ki ! A metódust TILOS feltüntetnie ! (2 pont)

```
class MyClass {  
    public void mx(YourClass y) { }  
}
```



7. Rajzoljon UML2 **szekvenciadiagramot** ! Az üzeneteket hierarchikus számozással lássa el ! (7 pont)

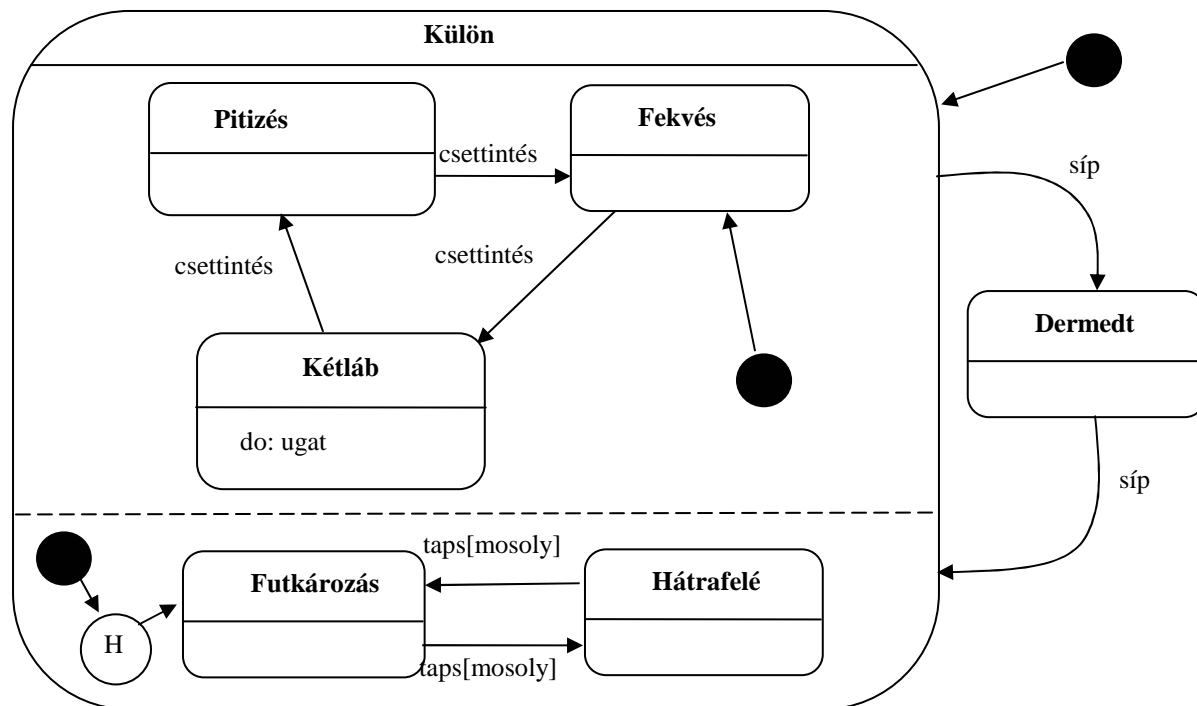
Böbe néni süt egy bejglit, és elküldi Maca néninek. Maca erre gyorsan süt egy másikat, mindkettőt megkóstolja, és a sajátját visszaküldi Böbének. Böbe a süteményt, választ sem várva, átküldi a házmesternek.



8. A felülvizsgálat (review, walkthrough stb.) során minden egyes „akció elem”-hez (hiba, ellentmondás, javaslat stb.) a következőket kell előírni (6 pont) :

- the responsible person
- the action to be taken
- the problem severity and type of bug

9. Rajzoljon UML 2 **állapotdiagramot** (state chart) az a következő leírás alapján a kutyapárról! (7 pont)
A Csinnbumm Cirkusz szerződtette Lali Bohócot. A bohócnak van egy idomított kutyapárja (Csahos és Rühes). Mindkét kutya külön-külön mutatványt tud, de néha ugyanazt csinálják. Csahos csettintésre pitizik, újabb csettintésre fekszik, újabb csettintésre két lábon járva ugat, végül újabb csettintésre ismét pitizik. A műsort a fekvéssel kezdi. Rühös tapsra vált ha, közben a bohóc mosolyog: először futkározik (mindig ezzel kezd), aztán hátrafele megy, majd megint futkározik. Amikor a bohóc megfújja a sípját, a két kutya megdermed, és addig így maradnak, amíg újabb sípszó nem érkezik. Rühös az okosabb, ő ott folytatja, ahol abbahagyta, Csahos mindig fekvéssel kezd.



10. A RUP (Rational Unified Process) egyik munkafolyamatában (workflow) szerződés (contract) készítését javasolja illetve írja elő. Melyik munkafolyamatban esedékes szerződés készítése ? Kik között kell szerződést készíteni ? Milyen fontosabb pontjai vannak a szerződésnek ? (4 pont)

Workflow: analízis **Kik: az operációk és az operációk felhasználói között (rendszer szinten).**
Főbb pontok: Responsibilities, Pre-conditions, Post-condition, Types, Crossrefs, Exceptions, Output

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

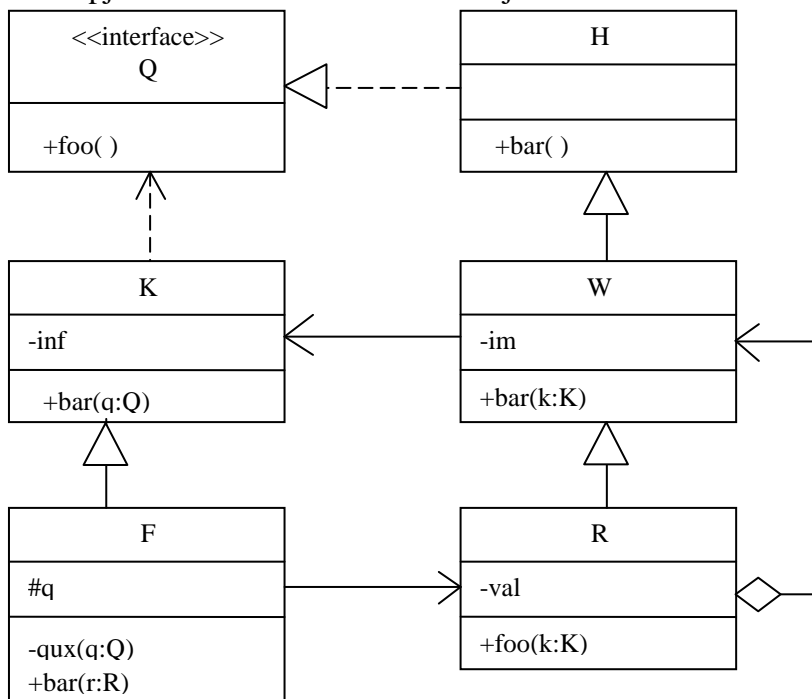
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2009. január 13. 14:00

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



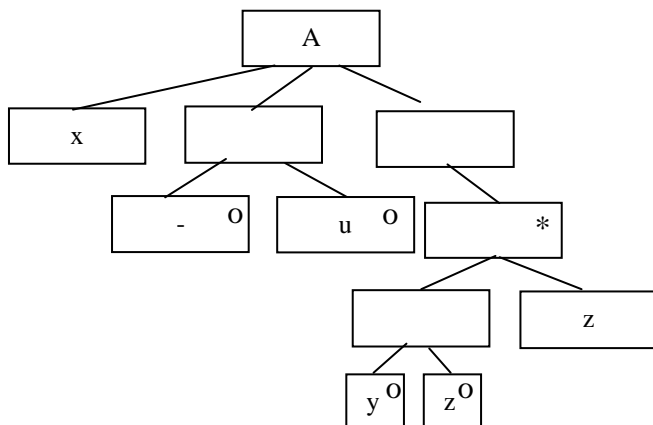
- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [D] H bármikor helyettesíthető R-rel, mert R a H leszármazottja.
- [A] W nem módosíthatja K inf attribútumát, mert az attribútum protected.
- [E] F bar(r:R) metódusa nem hívhatja meg a qux(q:Q) metódust, mert az utóbbi statikus.
- [B] F qux(q:Q) meghívhatja a bar(r:R) metódust a q paraméterrel, mert az R megvalósítja a Q interfészt.
- [E] Az ábrán szereplő összes egy-paraméteres bar metódus kaphat R objektumot paraméterül, mert R megvalósítja az összes említett metódust.
- [E] W bar(k:K) metódusa nem módosíthatja az osztály im attribútumát, mert az attribútum konstans.
- [C] W rendelkezik foo() szignatúrájú metódussal, mert függ K-tól.
- [D] Egy F objektum meghívhatja saját magával mint paraméterrel egy R foo(k:K) metódusát, mert F a K leszármazottja.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

2. Legyen egy A entitás, amelyen az x, y, z és u események fordulhatnak elő. Rajzolja fel az A entitás élettörténetét a JSD szerint, ha az események sorrendjét az alábbi BNF leírással definiáltuk ! (4 pont)

$x[u] \{ [y | z] z \}$



3. A verziókezelésben mit jelentenek az alábbi fogalmak? (2 pont)

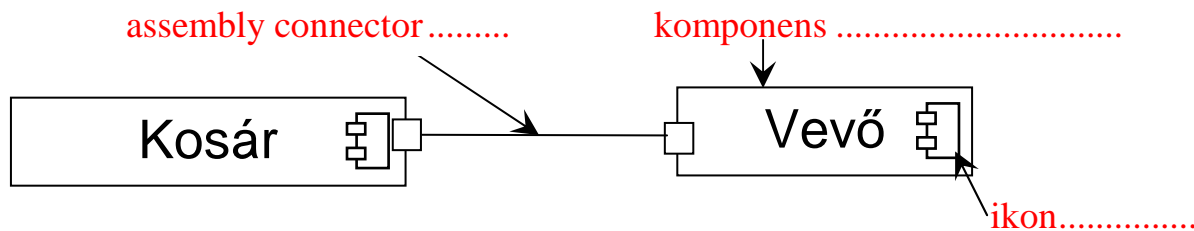
Check-out: **kezelt elem kiemelése felhasználásra a közös tárból**.....

Check-in: **kezelt elem visszahelyezése a közös tárbá**

Milyen elvek szerint kezelik a párhuzamos hozzáférést ? (2 pont)

Reserved check-outs **Modify-update-merge**

4. Az alábbi ábrán három UML2 modell elemet megjelöltünk. Adja meg elemenként, hogy az melyik UML2 meta-modell elem példánya ! (3 pont)



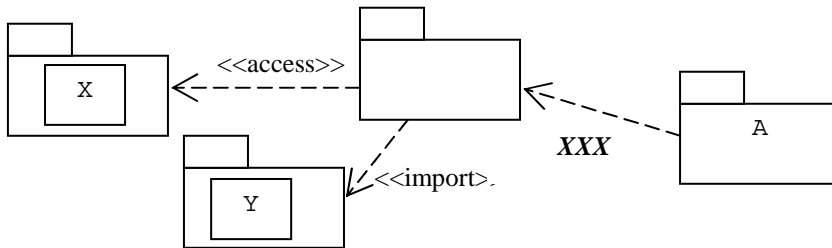
5. Az alábbi táblázat első oszlopába írja be az UML-ben definiált konkurencia szemantikák nevét !

Egy embernek több telefonja van. Miközben az egyikben beszél, egy másikon is hívás érkezik. A táblázatba írja be, hogy a különböző szemantikák szerint a konkurens hívás előfordulhat-e, és mi történik az éppen zajló beszélgetéssel és az új hívással ! (5 pont)

szemantika neve	előfordulhat-e?	mi történik ?
sorrendi (Sequ)	<input type="checkbox"/>	folytatja
őrzött (Guarded)	<input checked="" type="checkbox"/>	folytatja - ha befejezte, felveszi
konkurens (Conc)	<input checked="" type="checkbox"/>	félbeszakítja - azonnal felveszi
	<input type="checkbox"/>	

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

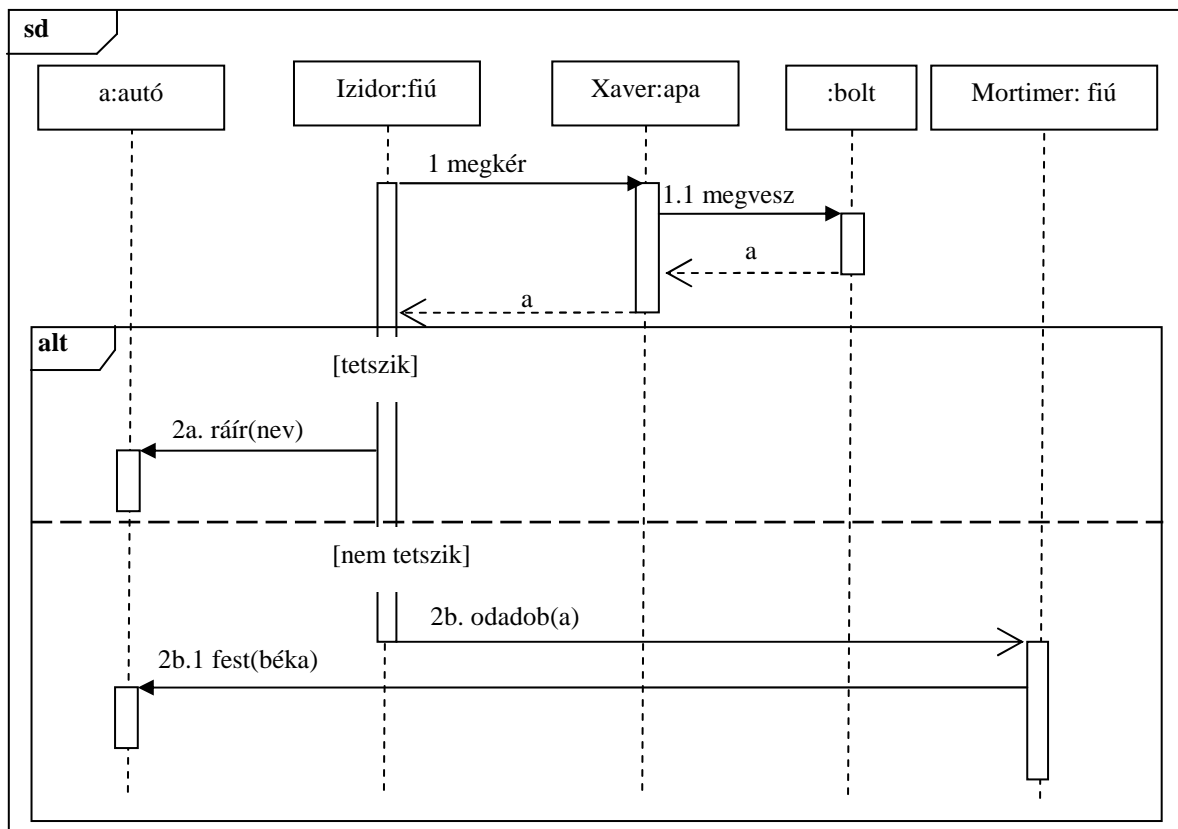
6. A táblázatba írja be, hogy az ábrán látható UML2 csomagdiagramban szereplő A csomagban milyen elemeket látunk – annak függvényében, hogy mi az **XXX** jelű sztereotípa ! (2 pont)



XXX	elemek
<<access>>	Y
<<import>>	Y

7. Rajzoljon UML2 **szekvenciadiagramot** ! Az üzeneteket hierarchikus számozással lássa el ! (7 pont)

Izidor megkéri apukáját, Xavért, hogy vegyen neki egy versenyautót, mire együtt elmennek a boltba, ahol Xavér megveszi a fiának a versenyautót, majd elsiet a dolgára. Izidornak ha tetszik az autó, akkor ráírja a nevét, ha nem, akkor odadobja az öccsének, Mortimernek és elrohan. Utóbbi esetben Mortimer az autóra ráfest egy békát.



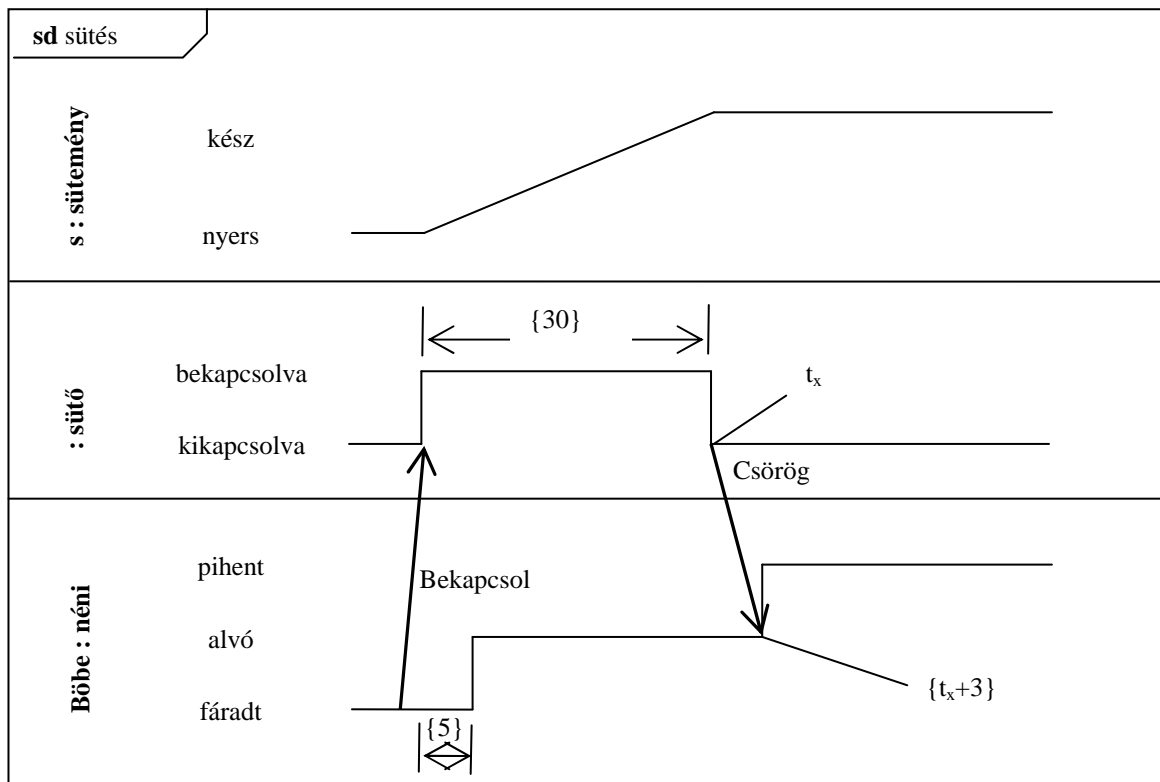
8. Melyek a legfontosabb integrációs stratégiák és azok milyen következménnyel járnak a tesztelésre nézve ? (6 pont)

Top-down test strategy, use stubs to simulate units before they are ready
 Bottom-up test strategy, build units first, use test-beds

9. Az alábbi történet alapján rajzoljon UML 2 időzítési diagramot (timing diagram)! (7 pont)

Böbe néni fáradt, de másnapra süteményt kell sütnie. A nyers süteményt beteszi a sütőbe, amit aztán bekapcsol. Öt perc múlva Böbe néni elalszik. A sütő fél óra elteltével kikapcsol és csörög, amire Böbe néni 3 perc múlva kipihenten felébred, és kiveszi a kész süteményt.

Böbe néni diszkrét állapotai: fáradt, alvó, pihent. A sütő diszkrét állapotai: ki, be. A sütemény folytonosan változik a nyers és a kész között.



10. A RUP (Rational Unified Process) use-case vezérelt.

Miben különbözik a magas szintű (high level) és a kiterjesztett (expanded) use-case ? (2 pont)

A leírás részletezettségében

(magas szintű - név, aktorok, cél, attekintés, referencia

kiterjesztett - eseményfolyam, elő- és utófeltételek, kiterjesztési pontok, relációk, aktivitás és use-case diagram referenciák)

Miben különbözik a lényeges (essential) és a valóságos (real) use-case ? (2 pont)

(A technológia függőségében

lényeges - eszköz, implementáció független

valóságos - implementáció (ablakok, mezők, triggererek)

Eredmények értékelése:

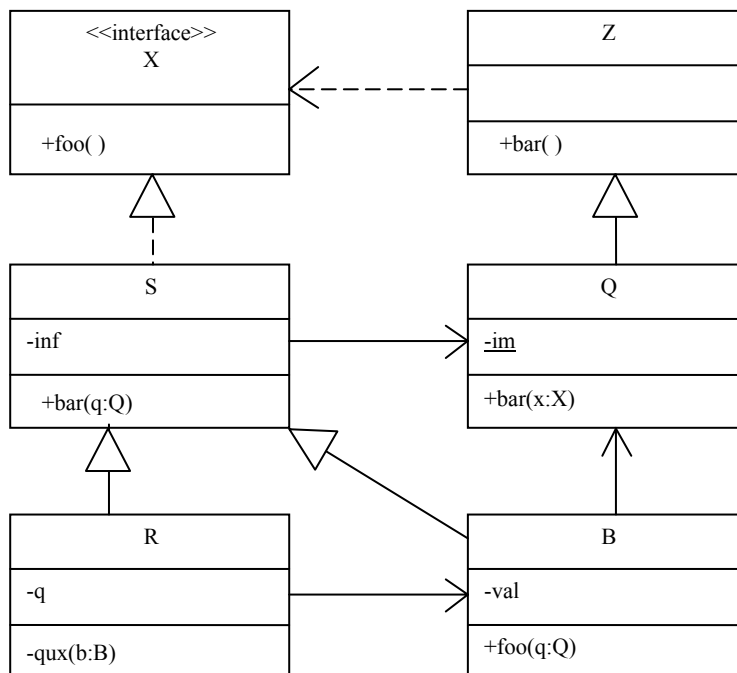
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR A SZOFTVERTECHNOLÓGIA

c. tárgyból
2011. január 4.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [E] Q helyettesíthető S-sel, mert S a Q leszármazottja
- [C] S helyettesíthető B-vel, mert B megvalósítja az X interfészt
- [A] R átadható paraméterül Q bar(x:X) metódusának, mert Q és S interfésze megegyezik.
- [B] B foo(q:Q) metódusa nem látja saját val attribútumának értékét, mert az attribútum privát.
- [E] B interfésze tartalmazza a bar(x:X) metódust, mert a metódus statikus
- [E] Q meghívhatja S bar(q:Q) metódusát, mert mindketten megvalósítják az X interfészt.
- [A] Q bar() metódusa nem módosíthatja az im attribútumot, ezért az attribútum konstans.
- [A] Q nem implementálja a foo() metódust, ezért nem függ az X interfésztől.

2. Adott az alábbi dekorált XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x (b|c)>
  <!ELEMENT a ANY>

  <!ELEMENT b (#PCDATA ①a)*>
  <!ELEMENT c (#PCDATA)>
]>
<x>
  <b>b<a><x><c>d</c>
  ②
  </a>z<b>
  </b></a></b>
</x>
```

Mit írna ① helyére, hogy a DTD jól formált legyen ? (2 pont)

- semmit
- |
- +
- *
- ”, ” (vessző)

Feltételezve, hogy a DTD jól formált, mi állhat ② helyében, hogy az XML érvényes legyen ? (4 pont)

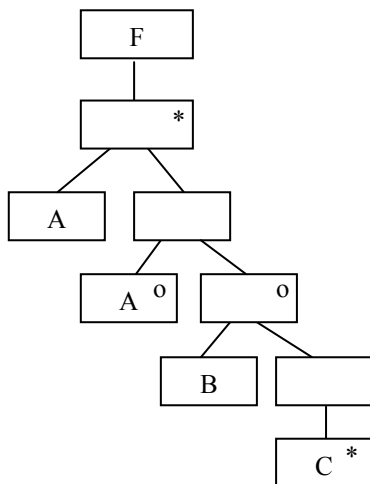
- </x><x/><a><a>
- <a>
- </x>jj<a><a>z
- b<a><x>
- </x>jj<x/><a><a>z
- b<a><x><x/>
- </x>j<a>z
- </x>j<a><a>

3. Jelölje az állítások mellett 1-5-ig, hogy minimálisan melyik CMM szinttől igazak ! Ha az állítás nem értelmezhető, akkor tegyen X-et ! (4 pont)

X	Az összes forráskód minőségét kimerítően (exhaustive) ellenőrzik.
2	A folyamat lépései meghatározottak, de nem szabványosak.
X	Orkesztráció (orchestration) elvén szervezik a munkafolyamatokat.
2	A felülvizsgálatok (review, inspection) ütemterv szerint történnek.

4. Készítsen állapotábrát, amely megfelel az alábbi Jackson diagramnak! (4 pont)

Az induló állapot legyen az ① !

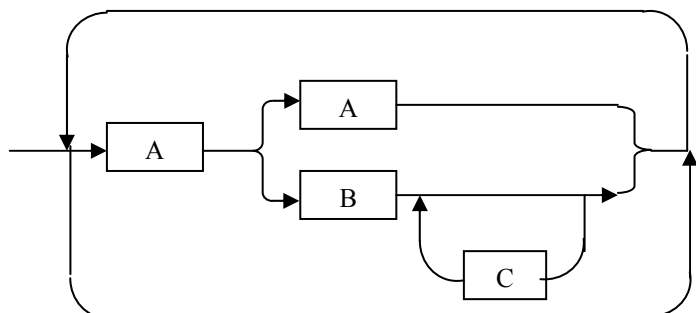


	A	B	C
①	②		
②	①	③	
③	②		③

Rajzolja fel a szintaxis gráfot ! (1 pont)

Adja meg az algebrai leírást ! (1 pont)

$$F = \{A + [A |(B + \{C\})]\}$$



A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Definiálja, a következő fogalmakat ! (3 pont)

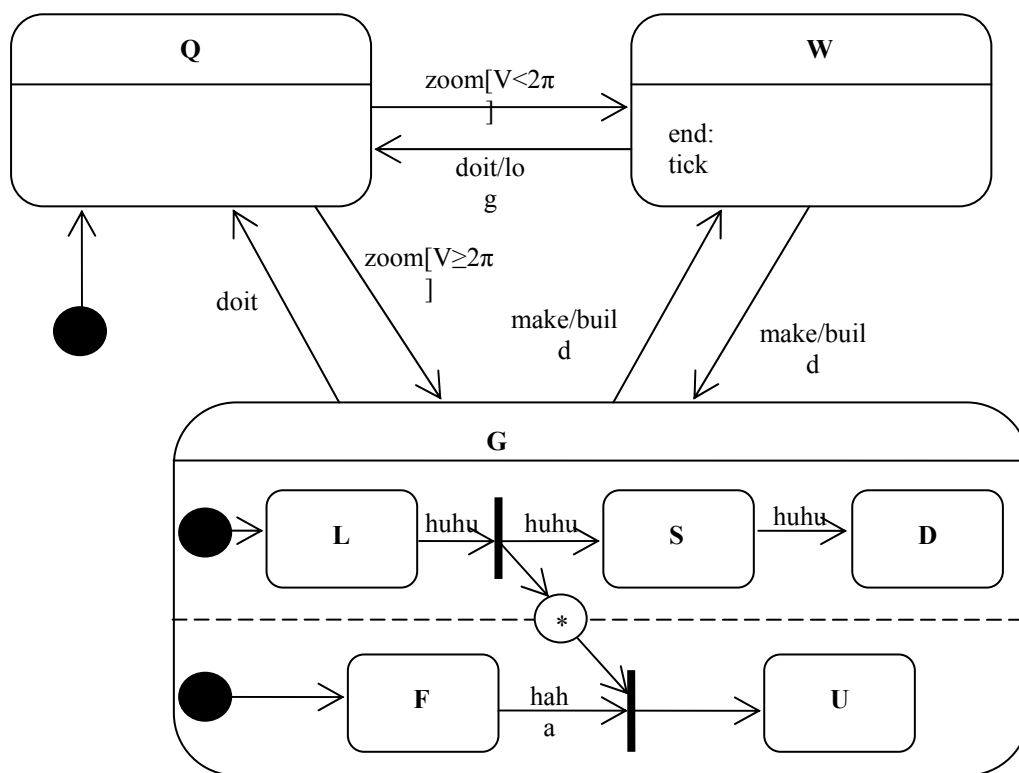
verzió (version) functionally distinct instance of a system

variáns (variant) functionally equivalent, non-functionally distinct

kibocsájtás (release) a version distributed outside the development team

6. Egészítse ki az alábbi UML 2 állapotdiagramot (state chart) a következő leírás alapján! (8 pont)

Egy objektum három fő állapottal (**Q**, **W**, **G**) rendelkezik. A kezdőállapot a **Q**. Ha **Q**-ban **zoom** esemény éri, akkor attól függően, hogy **V** értéke kisebb, mint 2π vagy sem, rendre a **W** vagy a **G** állapotba kerül. Mindkét állapot a **doit** és az **make** események hatására hagyható el. Előbbi esemény esetén visszatér **Q**-ba, utóbbinál pedig (lefuttatva a **build** metódust) **W**-ből **G**-be, **G**-ből **W**-be kerül. **W**-t elhagyva a **tick** metódus hívódik meg. **W**-ből a **doit** eseményre történő állapotváltás során a **log** metódus hívódik meg. **G** állapotban öt alállapot van, amelyek két, független csoportba oszthatók (**L**, **S**, **D**, és **F**, **U**, a csoportok első tagjában kezdünk). **L**-ből **S**-be, **S**-ből **D**-be jutunk a **huhu** esemény hatására. **F**-ből **U**-ba kerülhetünk a **haha** eseményre. Kapcsolat annyiban van köztük, hogy **U**-ba csak akkor kerülhetünk, ha már elhagytuk **L**-t.



7. Milyen szoftver verifikációs módszert alkalmazhatunk a fejlesztés specifikációs fázisában ? (2 pont)

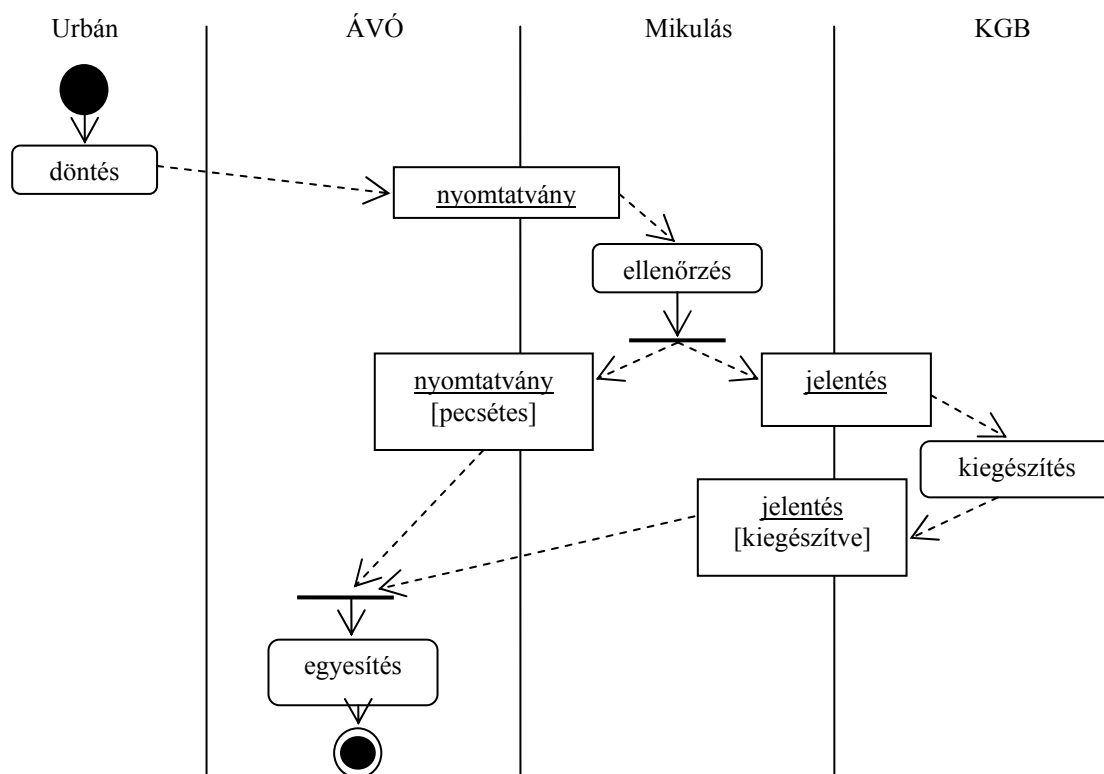
Review

Hogyan verifikálhatók a nem-funkcionális jellemzők ? (2 pont)

Test

8. Készítsen UML 2 aktivitás-diagramot (activity diagram) az alábbi leírás alapján! Jelölje az action-object flow-t is! Használja a kövéren szedett kifejezéseket! (8 pont)

Virgonc **Urbán** úgy **döntött**, hogy maradni kíván a MalacNyúzó Pribékek (MaNyúP) társaságában. Emiatt, az előírásoknak megfelelően, a **Mikulás**nak átad egy **formanyomtatványt**, amin szerepel a DNS mintája, az ujjlenyomata és az íriszképe. A Mikulás a nyomtatványt **ellenőrzi**, majd **lepecsételve** elküldi az Ákombákom Vizslató Orrszarvúnak (**ÁVÓ**), megőrzésre. A Mikulás az ellenőrzés után egy **jelentést** is küld a Központi Gépészeti Bizottságnak (**KGB**), amely a jelentést **kiegészíti** az ítélettel, és továbbítja az Orrszarvúnak, aki a két iratot **egyesíti**.



9. Ismertesse a RUP-ban a fogalmi model kialakításakor használt "térképész elv"-et ! (3 pont)

- use the existing names in the territory
- exclude irrelevant features
- do not add things that are not there

Eredmények értékelése:

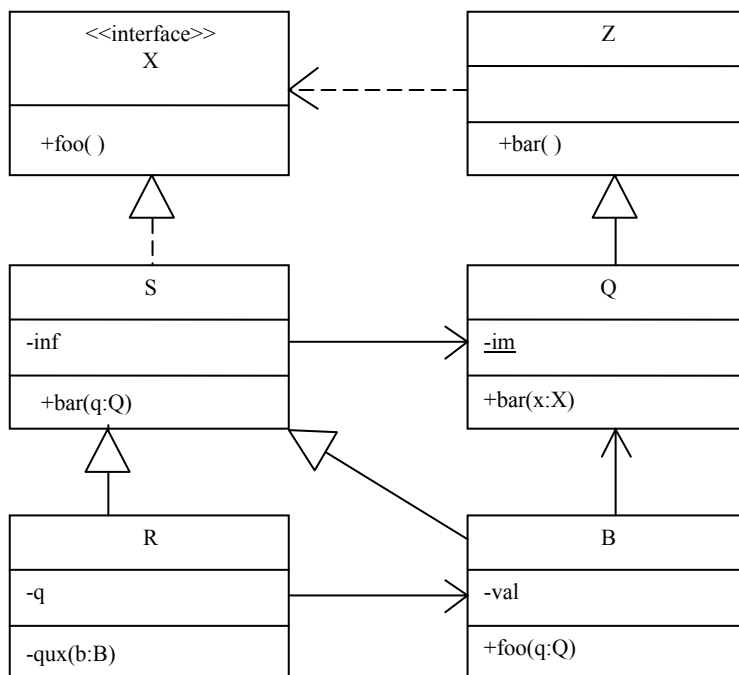
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR B SZOFTVERTECHNOLÓGIA

c. tárgyból
2011. január 4.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B] R helyettesíthető B-vel, mert R függ B-től
- [E] Q meghívhatja S bar(q:Q) metódusát, mert mindketten megvalósítják az X interfészt.
- [E] Q helyettesíthető S-sel, mert S a Q leszármazottja
- [E] B interfésze tartalmazza a bar(x:X) metódust, mert a metódus statikus.
- [B] B foo(q:Q) metódusa nem látja saját val attribútumának értékét, mert az attribútum privát.
- [C] Q bar() metódusa nem módosíthatja az im attribútumot, ezért az attribútum statikus..
- [A] Q nem implementálja a foo() metódust, ezért nem függ az X interfésztől.
- [A] B átvadható paraméterül Q bar(x:X) metódusának, mert Q és S interfésze megegyezik.

2. Adott az alábbi dekorált XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE y [
  <!ELEMENT y (a+,b)>
  <!ELEMENT a ANY>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA ①y)*>
]>
<y>
  <a><y><a>
    ②
    <b></b></y>
  a/</a><b>/b</b>
</y>
```

Mit írna ① helyére, hogy a DTD jól formált legyen ? (2 pont)

- semmit
- |
- +
- *
- ”, ” (vessző)

Feltételezve, hogy a DTD jól formált, mi állhat ② helyében, hogy az XML érvényes legyen ? (4 pont)

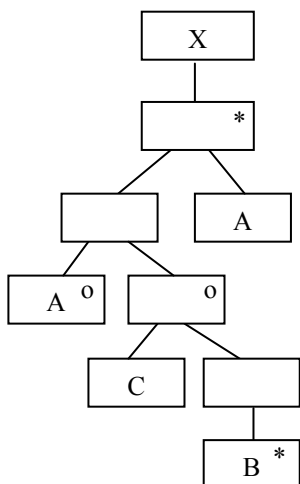
- <c>/b</c>b
- <y><c>b</c></y>
- <a><c>b</c>b
- b/bb
- <c>b</c></y>
- <a>a/b
- <c>b</c>
- <y><c>b</c></y>

3. Jelölje az állítások mellett 1-5-ig, hogy minimálisan melyik CMM szinttől igazak ! Ha az állítás nem értelmezhető, akkor tegyen X-et ! (4 pont)

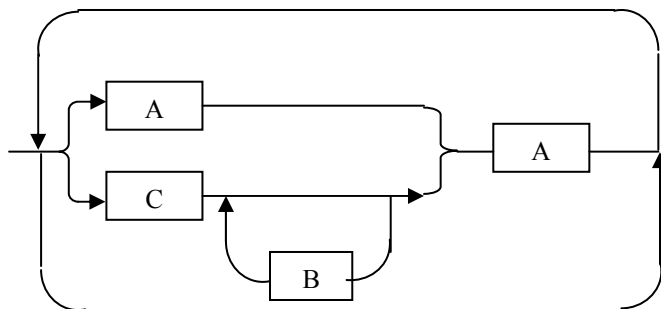
X	Koreográfia (choreography) elvén szervezik a technológiai folyamatokat.
2	Fejlett projekt-menedzsment technikákat és eszközöket alkalmaznak a napi gyakorlatban.
X	A projektekben rendszeresen alkalmazzák az agilis programozást.
X	A vezetők képesek a termékek minőségének közvetlen, számszerű ellenőrzésére.

4. Készítsen állapotábrát, amely megfelel az alábbi Jackson diagramnak! (4 pont)

Az induló állapot legyen az ① !



	A	B	C
①	②		③
②	①		
③	①	③	



Rajzolja fel a szintaxis gráfot ! (1 pont)

Adja meg az algebrai leírást ! (1 pont)

$$X = \{ [A | (C + \{B\})] + A \}$$

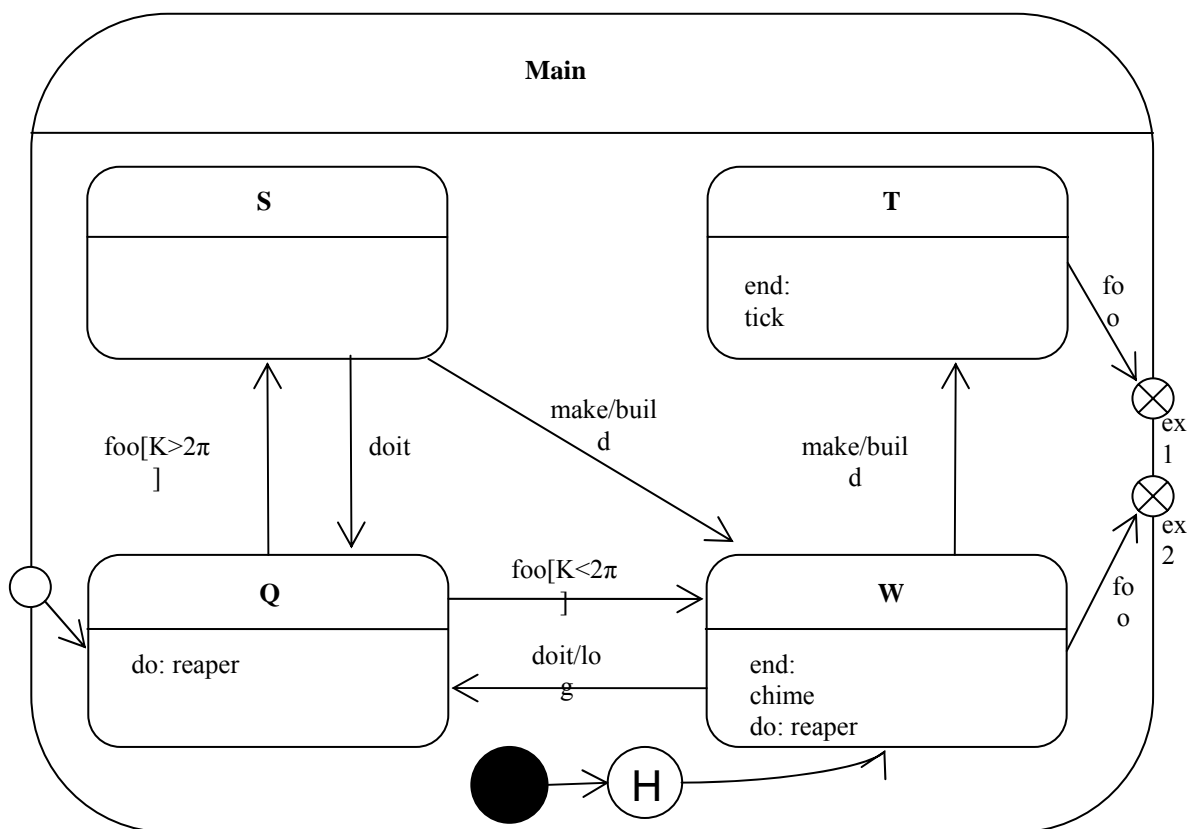
A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Mit jelent a "baseline" ? (4 pont)

A baseline is a 'snapshot' in time of one version of each artifact in the project repository. It provides an official standard on which subsequent work is to be based, and to which only authorized changes can be made. After a baseline is set every subsequent change to a baseline is recorded as a delta until the next baseline is set.

6. Egészítse ki az alábbi UML 2 állapotdiagramot (state chart) a következő leírás alapján! (8 pont)

Egy objektum **Main** állapotában 4 alállapot található (**S**, **T**, **Q**, **W**). A **Main** állapotba egy belépési ponton (entry point) léphetünk be (**en1**), és két kilépési ponton (exit point) hagyhatjuk el (**ex1**, **ex2**). Ha nem **en1**-en lépünk be, akkor abba az állapotba kerülünk, amelyikben utoljára voltunk. Ha nem volt ilyen, akkor a **W**-be. Az **en1**-ből a **Q**-ba kerülünk. Ha **Q**-ban **foo** esemény éri, akkor attól függően, hogy **K** értéke kisebb, mint 2π vagy sem, rendre a **W** vagy az **S** állapotba kerül. Mindkét állapot a **doit** és az **make** események hatására hagyható el. Előbbi esemény esetén visszatér **Q**-ba, utóbbinál pedig (lefuttatva a **build** metódust) **W**-ből **T**-be, **S**-ből **W**-be kerül. **T**-ből kilépéskor mindig lefut a **tick** metódus. **W**-ből a **doit** eseményre történő állapotváltás során a **log** metódus hívódik meg. **Q** állapotban a **reaper** metódus fut. A **foo** esemény hatására **T**-ből **ex1**-be, **W**-ből **ex2**-be lépünk.



7. Egy szoftver felülvizsgálata (review) milyen következményekkel (follow-up) zárulhat ? (3 pont)

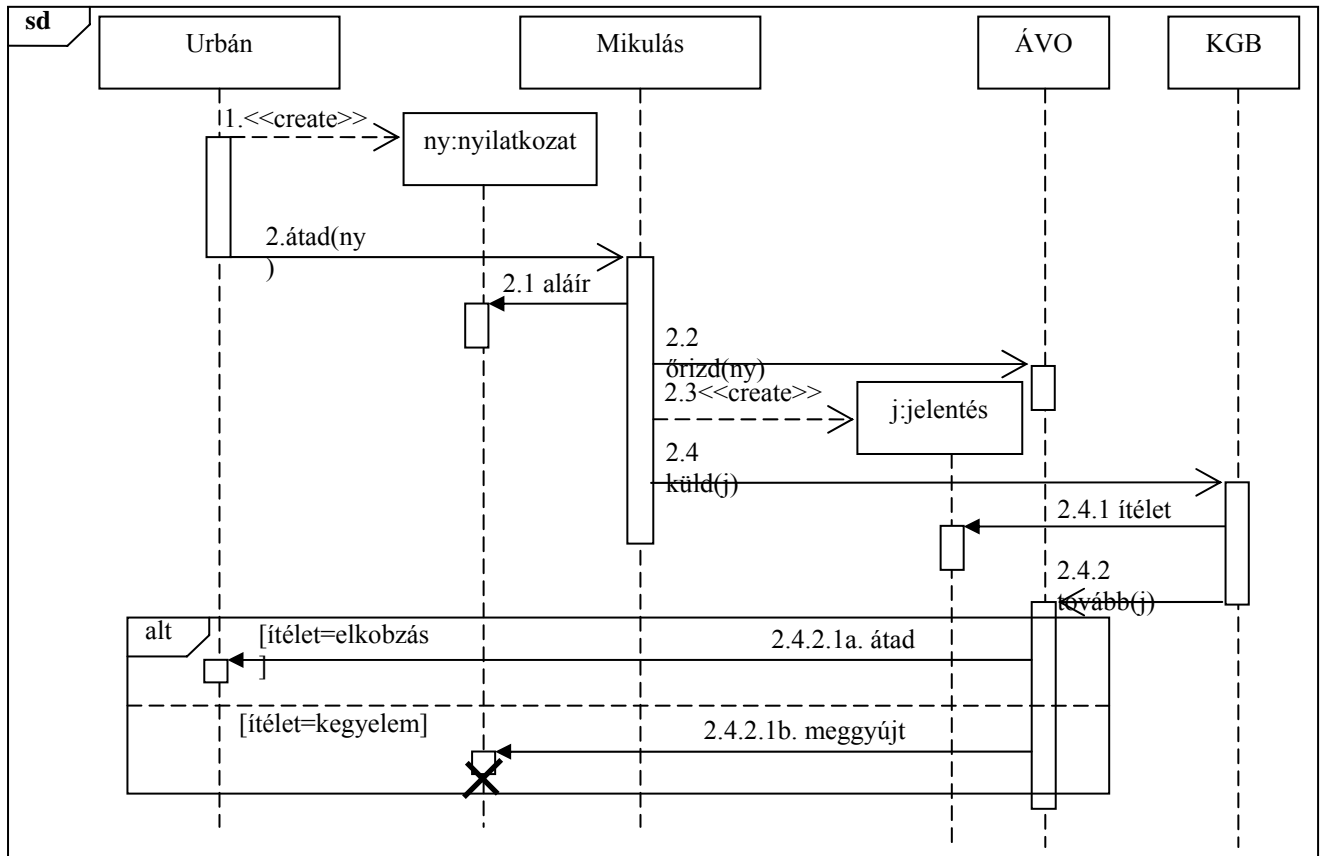
Unqualified acceptance of review item

Conditional acceptance of the review item with minor action (confirmation)

Conditional acceptance of the review item with major action (new review)

8. Készítsen UML 2 szekvencia diagramot (sequence diagram) az alábbi leírás alapján! Használjon hierarchikus számozást is! (8 pont)

Virgonc Urbán úgy döntött, hogy maradni kíván a MalacNyúzó Pribékek (MaNyúP) társaságában. Emiatt, az előírásoknak megfelelően, a Mikulásnak átad egy általa készített nyilatkozatot. A Mikulás a nyomtatványt aláírja, majd elküldi az Ákombákom Vizslató Orrszarvúnak (ÁVO), megőrzésre. A Mikulás ezután egy jelentést készít és küld a Központi Gépészeti Bizottságnak (KGB), amely a jelentéshez hozzáírja az ítéletet, és továbbküldi az Orrszarvúnak. Ha az ítélet vagyonekobzás, akkor az Orrszarvú erről személyesen értesíti Urbánt. Ha méltányos kegyelem, akkor meggyűjtja a nyilatkozatot, és megvárja amíg megsemmisül.



9. A RUP tervezési munkafolyamatában (workflow) az architektúra tervezésekor milyen döntéseket hozunk ? (3 pont)

- Organize the system into packages (subsystems)
- Identifying concurrency
- Allocating packages to processors
- Storage and Persistence
- Handling global resources
- Choosing software control
- Handling boundary conditions

Eredmények értékelése:

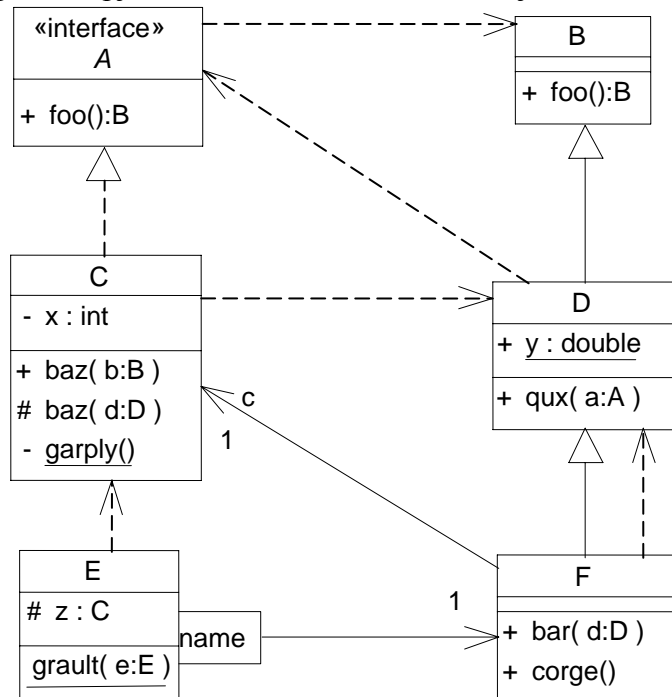
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2011. január 18.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



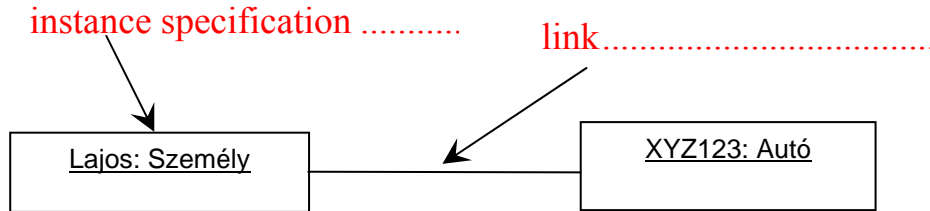
- | | |
|---|--------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (++ -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (++ +) |
| E - egyik tagmondat sem igaz | (-- -) |

- [B] D implementálja az A interfészt, mert A és B interfésze megegyezik.
- [B] F corge() metódusa nem módosíthatja D y attribútumát, mert D y attribútuma statikus.
- [B] E grault(e:E) metódusa nem hívhatja meg e z attribútumának baz(b:B) metódusát, mert a grault() statikus.
- [E] Egy F objektum pontosan egy E objektumot ismer, mert egy E objektum pontosan egy F objektumot ismer.
- [D] C foo() metódusa példányosíthat B típusú objektumot, ezért C függ B-től.
- [E] F bar(d:D) metódusából nem hívható meg c baz(b:B) metódusa, mert C baz(b:B) metódusa nem kaphat paraméterül D típusú objektumot.
- [B] C baz(d:D) metódusa nem hívhatja meg C garply() metódusát, mert C baz(d:D) metódusa nem statikus.
- [A] D qux(a:A) metódusa nem hívhatja meg egy paraméterül kapott C típusú objektum baz(b:B) metódusát, mert A nem helyettesíthető C-vel.

2. Nevezze meg az UML2-ben definiált gyűjteményeknek (kollekcióknak) a – tipizálásra is alkalmazott – két alapvető jellemzőjét ! (3 pont)

1. **rendezettség (ordered)** 2. **egyediség (unique)**.....

3. Adja meg, hogy az alábbi object diagramon a megjelölt elemek mely UML2 meta-modell elem példányai ! (2 pont)

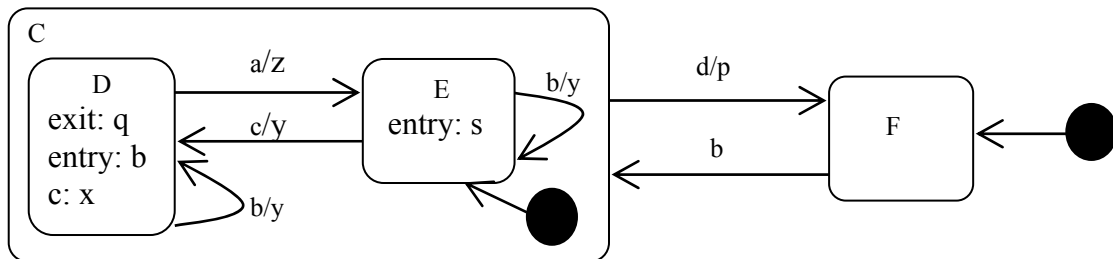


4. Az alábbi táblázatban a szoftver fejlesztési folyamat különféle modelljeit adtuk meg, minden modellhez egy indexet rendelve. A jobboldali táblázatban a fejlesztési folyamattal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik modellhez kapcsolhatóak leginkább ! (A fogalom előtti mezőbe írja be a modell indexét !) Ha a fogalom egyik modellhez sem társítható, akkor a mezőbe tegyen X-et ! (3 pont)

1	Ward-Mellor
2	ICOM
3	Vízesés (Waterfall)
4	V
5	Spirál

3	Karbantartás (maintenance)
5	Kockázat elemzés (risk analysis)
2	Erőforrások (resources)
1	Implementációs függőség (implementational dependency)
4	Egységteszt (unit test)
4	Integrációs teszt (integration test)

5. Rajzolja fel az alábbi UML2 state-chart-nak megfelelő állapotábrát ! (8 pont)



	a	b	c	d
D	E/q, z, s	D/q, y, b	D/x	F/q, p
E		E/y, s	D/y, b	F/p
F		E/s		

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

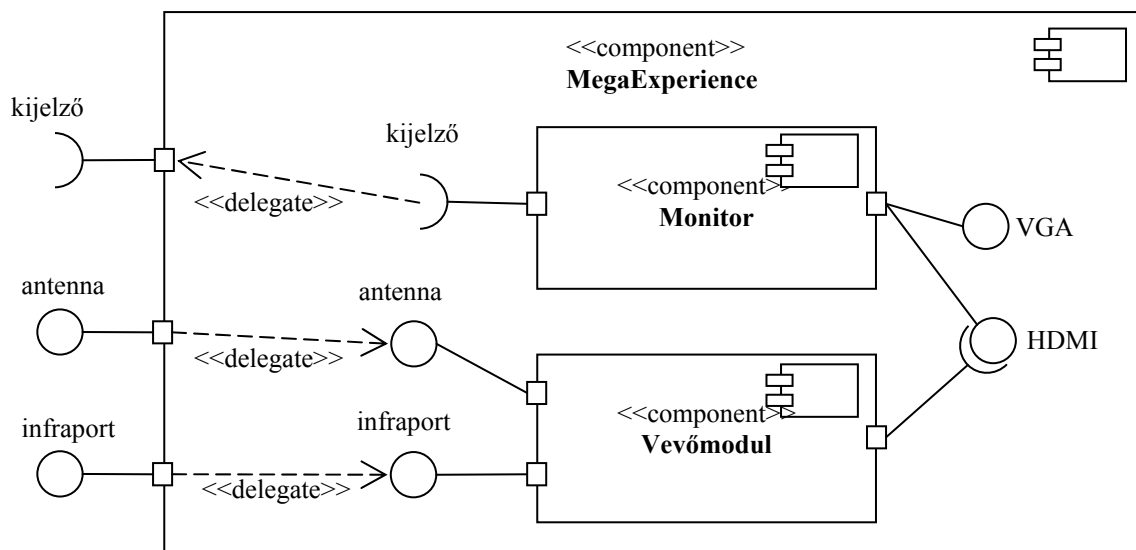
6. Mire szolgál a verziókezelésben alkalmazott „modify-update-merge” stratégia ? Röviden írja le a működésének lényegét ! (3 pont)

mire: termékekhez történő konkurens hozzáférés szabályozása

lényeg: párh. check-out, check-in-kor összefésülés, konfliktusfelold...

7. Készítsen UML2 komponens diagramot az alábbi leírás alapján! (8 pont)

A MegaExperience televíziókészülék a szokásos felülettel rendelkezik: van kijelzője (ami elvárja, hogy valaki nézze), antennabemenete és infraportja. Ha a készüléket szétszedjük, akkor azt látjuk, hogy valójában egy monitorból és egy vevőmodulból áll. A monitor biztosítja a kijelzést, míg az antennabemenet és az infraport a vevőmodulhoz kapcsolódik. A monitor VGA és HDMI interfésszel is rendelkezik, a modul ebből a HDMI interfészt használja a kommunikációhoz.



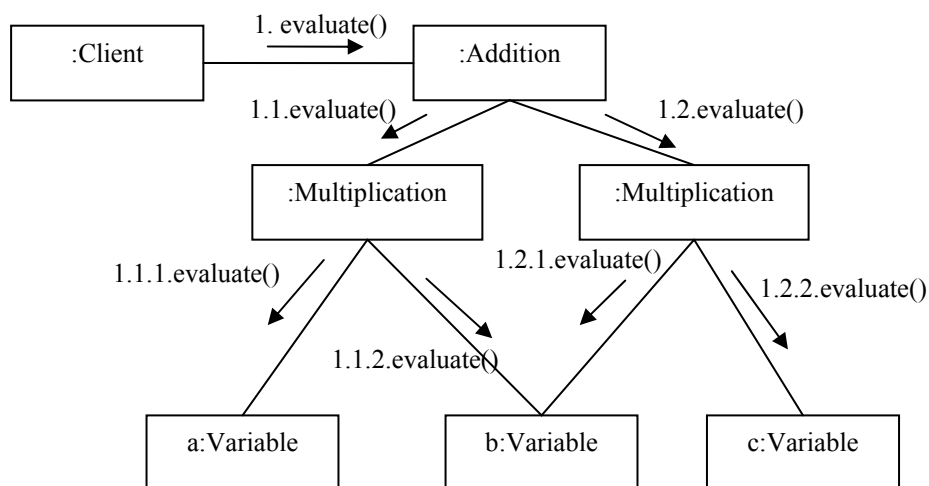
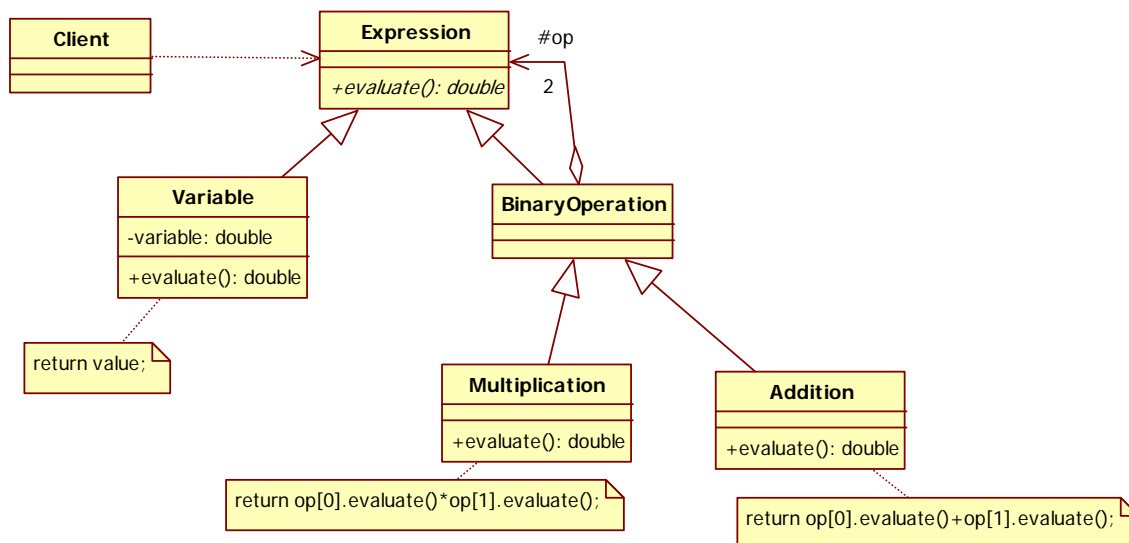
8. A bottom-up tesztelési (integrációs) stratégia esetében milyen kiegészítő eszközre van szükség ? (2 pont)

- **tesztágy (test bed)**

Mi a funkciója az eszköznek ? (2 pont)

- **a tesztelendő alrendszerek kezeléséhez keretet ad**

9. Az alábbi diagramon szereplő osztályokat használjuk algebrai kifejezések modellezésére. Készítsen UML2 kommunikációs diagramot arra az esetre, ha egy kliens kiértékeli az $(a * b) + (b * c)$ kifejezést! A rendszer nem készít felesleges objektumokat. Az üzeneteket hierarchikusan számozza! (7 pont)



10. A Rational Unified Process (RUP) tervezési szakaszában milyen típusú use-case-eket készítünk ? Mi ennek a típusú use-case-nek a fő jellemzője ? (4 pont)

valóságos (real)
a felhasználói felület elemeire hivatkozik

Eredmények értékelése:

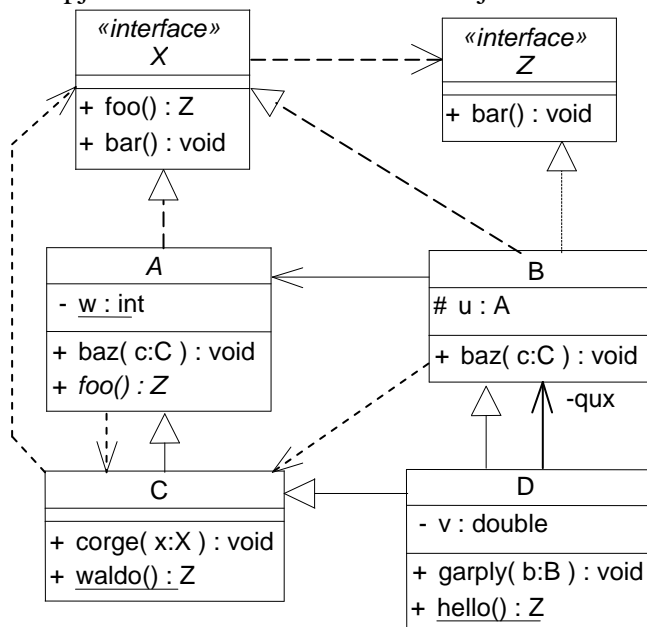
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2011. május 24.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

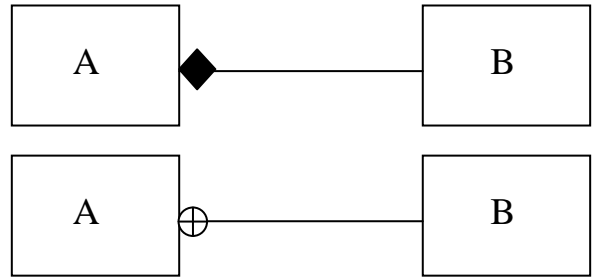
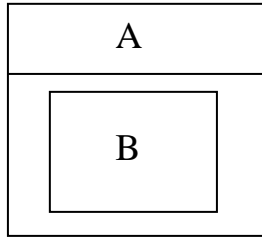
1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



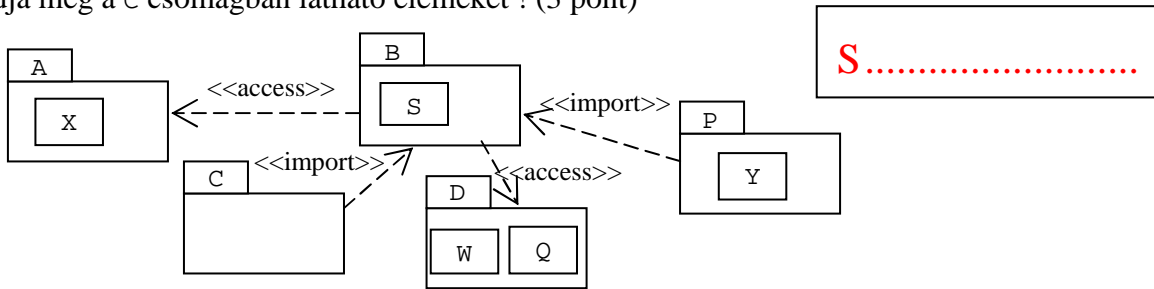
- A - csak az első tagmondat igaz (+ -)
 B - csak a második tagmondat igaz (- +)
 C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
 D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
 E - egyik tagmondat sem igaz (- -)

- [E] D garply metódusa nem módosíthatja a b paraméter u attribútumát, mert protected attribútumhoz csak privát és protected metódusok férhetnek hozzá.
- [A] C corge metódusa kaphat paraméterül D típusú objektumot, ezért a metódus meghívhatja a kapott objektum garply metódusát.
- [B] D garply metódusa kaphat paraméterül A típusú objektumot, mert A és B interfésze megegyezik.
- [B] C waldo metódusa virtuális, ezért a B osztály baz függvénye egy paraméterül kapott D típusú objektumon meghívhatja a waldo metódust.
- [B] A baz metódusa nem módosíthatja A w attribútumát, mert A baz metódusa nem statikus.
- [A] C-nek van bar metódusa, ezért C implementálja a Z interfészt.
- [C] D hello metódusa nem módosíthatja D v attribútumát, mert D v attribútuma privát.
- [B] B baz metódusa nem hívhatja meg B u attribútumának foo metódusát, mert az A osztály foo metódusa absztrakt.

2. Adott az alábbi UML2 diagram. Rajzoljon egy olyan másik UML2 diagramot, amely szemantikailag ugyanazt fejezi ki, mint az adott diagram ! (3 pont)



3. Adja meg a c csomagban látható elemeket ! (3 pont)

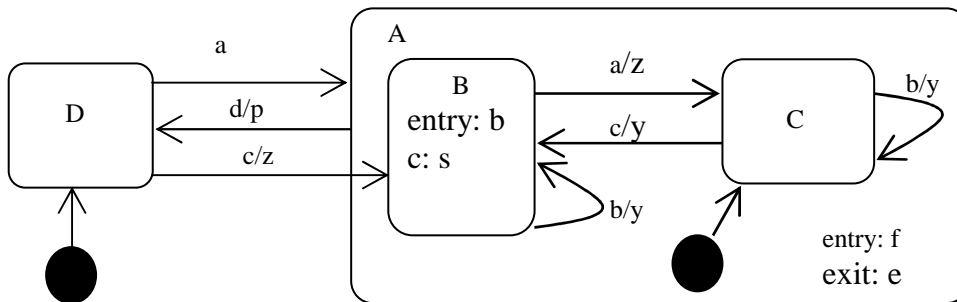


4. Az alábbi táblázatban a szoftver fejlesztési folyamat különféle modelljeit adtuk meg, minden modellhez egy indexet rendelve. A jobboldali táblázatban a fejlesztési folyamattal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik modellhez kapcsolhatóak leginkább ! (A fogalom előtti mezőbe írja be a modell indexét !) Ha a fogalom egyik modellhez sem társítható, akkor a mezőbe tegyen X-et ! (4 pont)

1	Ward-Mellor
2	ICOM
3	Vízesés (Waterfall)
4	v
5	Spirál

1	Absztrakció (abstraction)
4	Rendszerteszt (system test)
5	Prototípus (prototype)
2	Költségterv (budget)

5. Rajzolja fel az alábbi UML2 state-chart-nak megfelelő állapotábrát ! (6 pont)



	a	b	c	d
D	C/ f		B/z, f, b	
C		C/y	B/y, b	D/e, p
B	C/ z	B/ y, b	B/s	D/e, p

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Az alábbi XML leírás jól formált és érvényes-e? Ha nem, akkor mi a baja? (2 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE a [
  <!ELEMENT a ((b, c?)|c)>
  <!ELEMENT b ANY>
  <!ELEMENT c (#PCDATA)>
]>
<a> <b>
<c> <![CDATA[
</c> <b> <c> <b> </c>
]]> </b>
</b>
</a>
```

- IGEN
 NEM

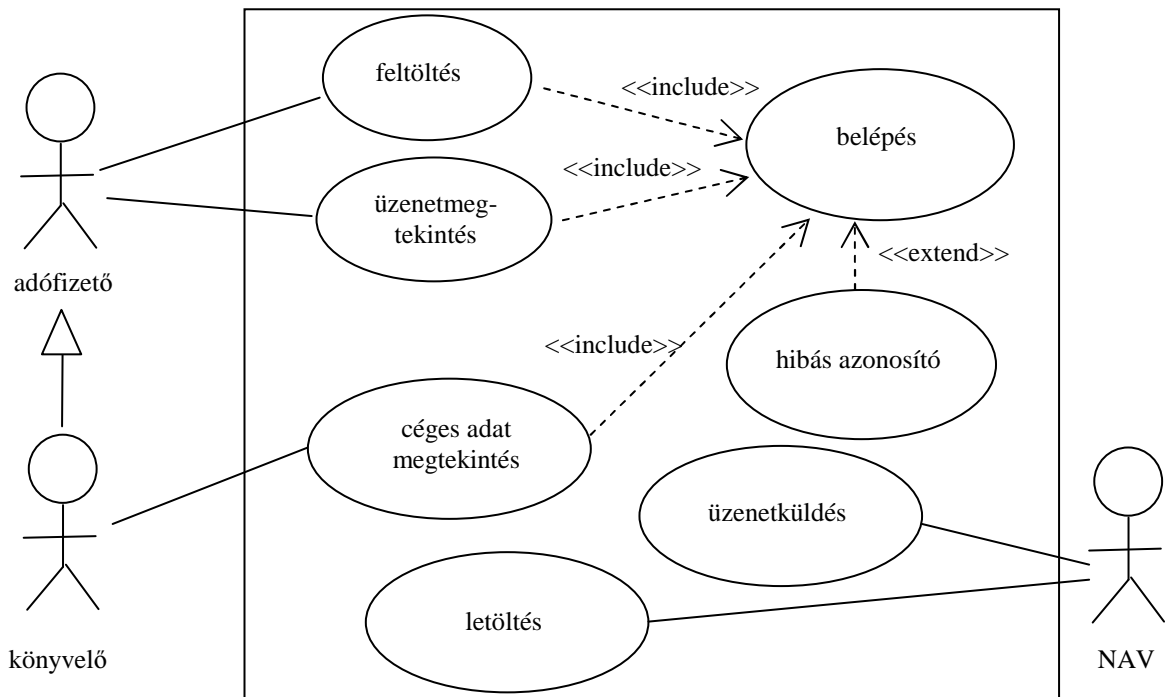
Hiba? **<c> tag nincs lezárva**.....

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van pontosan két c és egy b elem van! (3 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<a>
  <b>
    <c></c>
  </b>
  <c></c>
</a>
```

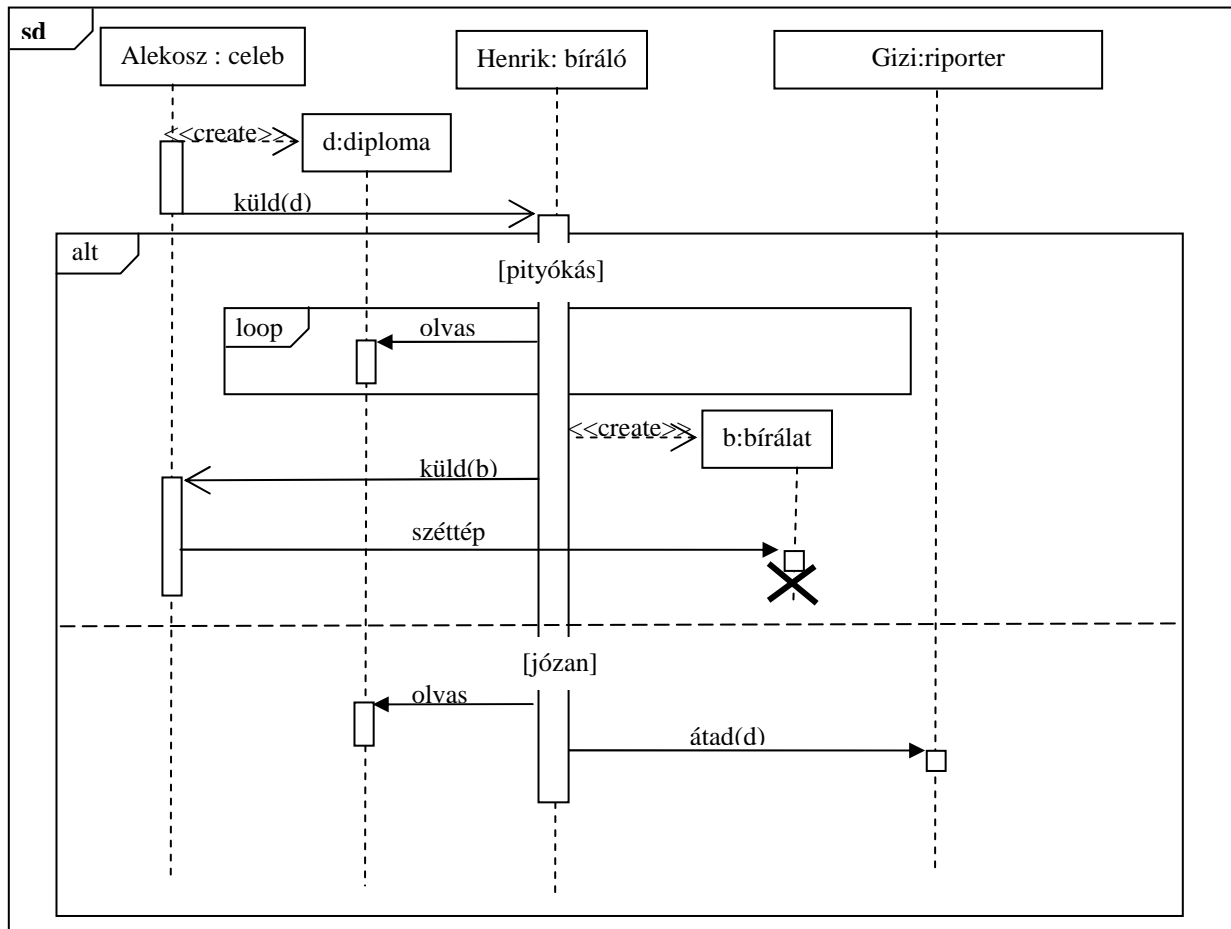
7. Rajzoljon UML2 use-case diagramot az alábbi történet alapján! (7 pont)

Az Ügyfélkapun az adófizető adóbevallást tud feltölteni, illetve hivatalos üzeneteit tudja megnézni. Mindkettőhöz be kell lépnie (név és jelszó megadásával). A könyvelő ezen kívül céges adatokat is meg tud nézni (ehhez szintén be kell lépnie). Ha belépéskor valaki rossz azonosítót ad meg, akkor hibáüzenetet kap. A Nemzeti Adó és Vámhivatal (NAV) le tudja tölteni az adóbevallásokat, és üzenetet tud küldeni.



8. Készítsen UML2 szekvencia-diagramot az alábbi leírás alapján! (8 pont)

Alekosz, a celeb diplomát ír, majd elküldi Hawass Henriknek bírálatra. Ha Henrik pityókás, akkor többször elolvassa a művet, majd ír egy kemény bírálatot, és utóbbit visszaküldi Alekosznak, aki a bírálatot széttépi. Ha Henrik józan, akkor csak egyszer olvassa el a dolgozatot, majd bizalmasan átadja Gizinek, a Flikk oknyomozó riporterének.



9. Adja meg, hogy egy szoftver elem (dokumentáció, kód) felülvizsgálata (review, inspection) során minden egyes azonosított probléma vagy hiba esetén a felülvizsgálók mit rendelkeznek (mi van a jegyzőkönyvben ?) ! (3 pont)

kijelölik a felelőst
 előírják a szükséges tevékenységet (pl. további vizsgálat, átdolgozás).....
 meghatározzák a hiba típusát és a probléma súlyát.....

10. Hogyan értelmezzük az UML-ben az őrzött (guarded) konkurenciát ? (A konkurencia szemantikája guarded) (3 pont)

multiple calls from concurrent threads may occur simultaneously to one instance, but only one is allowed to commence. Others are blocked

Eredmények értékelése:

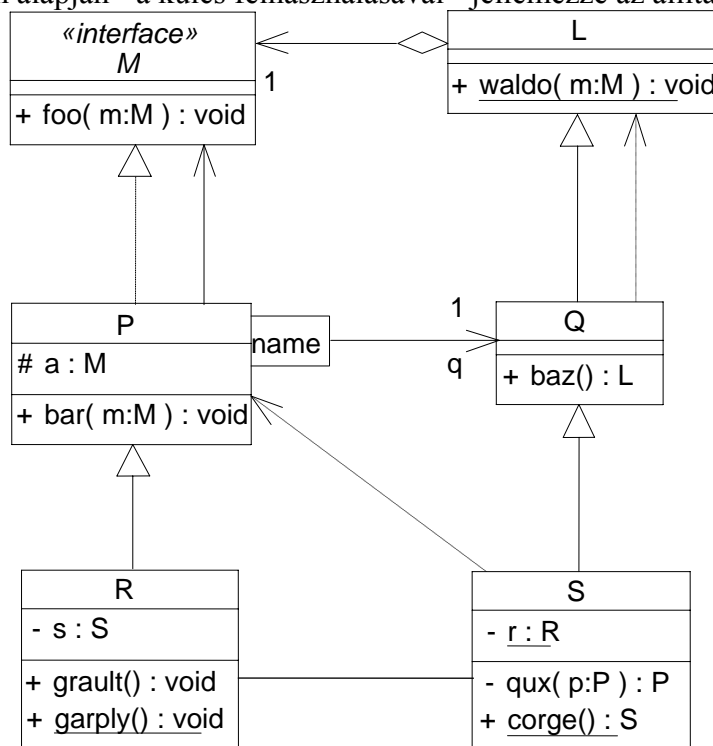
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2011. június 7.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

[B] **R** **garply** metódusa meghívhatja az **s** attribútum **corge** metódusát, mert a **garply** és a **corge** metódus is statikus.

[B] **P** nem hívhat **Q** osztályon **waldo** függvényt, mert **L** **waldo** függvénye nem virtuális.

[A] **Q** függ **M**-től, mert **Q** ismeri **P**-t.

[A] Az **M** típus közvetlenül nem példányosítható, ezért **R** **gault** metódusa nem hívhatja meg **P** **a** attribútumának **foo** metódusát.

[B] Egy **P** típusú objektum pontosan egy **Q** típusú objektumot ismer, ezért **P** függ **Q**-tól.

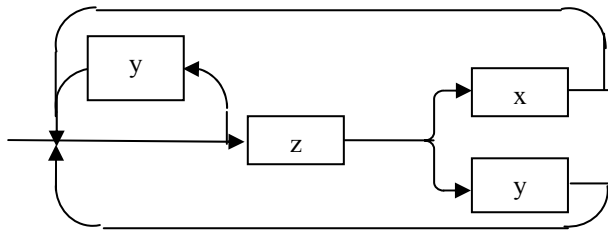
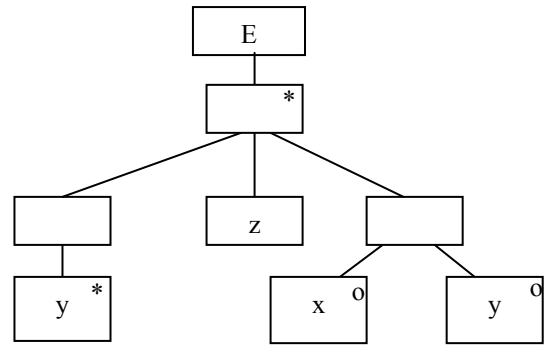
[A] **S** az **L** leszármazottja, ezért **Q** **baz** függvénye példányosíthat **S** típusú objektumot.

[D] **S** **qux** függvénye nem módosíthatja az **r** attribútum **s** attribútumát, mert **R** **s** attribútuma privát.

[D] **S** ismeri **M**-et, mert az **S** osztály **L** őse függ **M**-től.

2. Az E entitás élettörténetét az alábbi állapottáblával írhatjuk le (A a kezdőállapot).
Rajzolja fel az élettörténetet JSD ábrán és szintaxisgráffal! (6 pont)

	x	y	z
A		A	B
B	A	A	



3. Mi a lényege a JSD-ben a “funkció” lépésnek (function step)? (2 pont) Rossz válasz esetén pontot veszít !

- Az entitások funkcióinak specifikálása
- A funkcionális dekompozíció
- Az entitások kibővítése funkció-entitásokkal
- Funkció processzek élettörténetének tervezése
- Funkció processzek felvétele a processzek hálózatába**
- Funkció processzek ütemezése

Mi az **előző** lépés? (1 pont)

initial model

Mi a **következő** lépés? (1 pont)

timing

4. Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe? (3 pont)

komponensek: eseményforrás, eseményfigyelő

szerepek: az eseményfigyelők beregisztrálják magukat a forrásoknál, akik a náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek

komponensnek

5. Sorolja fel a Rational Unified Process (RUP) életciklus modelljében szereplő „támogató munkafolyamatokat” (supporting workflows)! (3 pont)

konfigurációs menedzsment

menedzsment

környezet

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. A RUP-ban (Rational Unified Process) alkalmazott használati esetek (use-case-ek) különböző szempontok szerint csoportosíthatóak. Jellemezze a **lényeges** (essential) és a **valóságos** (real) use case-eket! Miben különböznek egymástól? (3 pont)

lényeges - eszköz, implementáció független
valóságos - implementációs (ablakok, mezők, triggerek)
a technológiai függőségében

7. Adott az alábbi XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

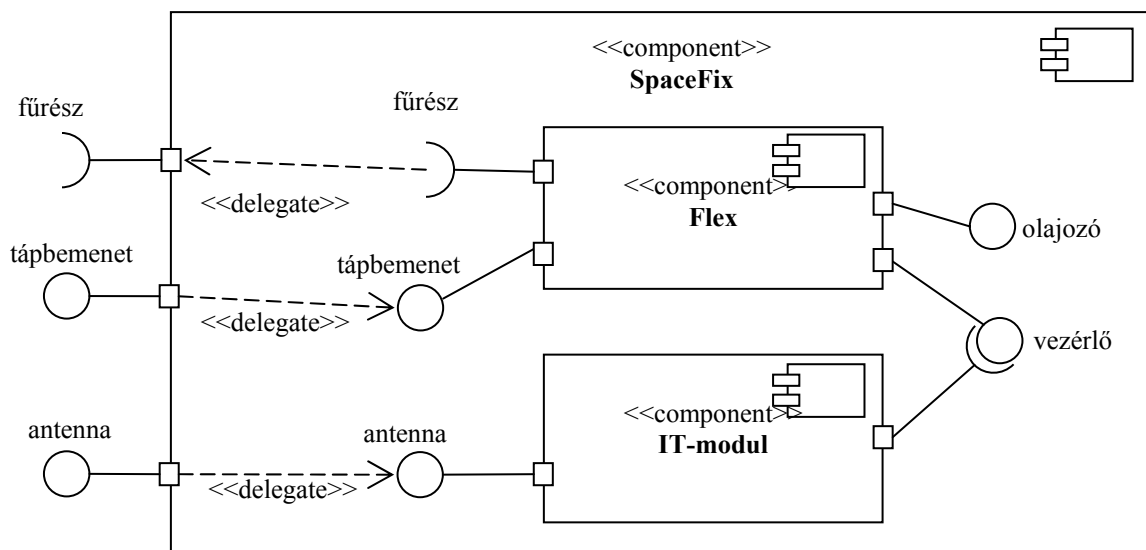
```
<!DOCTYPE x [
  <!ELEMENT x      (y+, c)>
  <!ELEMENT y      (a | (b+, c?))>
  <!ELEMENT a      (#PCDATA)>
  <!ELEMENT b      (#PCDATA)>
  <!ELEMENT c      (#PCDATA)>
]>
```

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van pontosan két c elem van! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

```
<x>
  <y>
    <b></b>
    <c></c>
  </y>
  <c></c>
</x>
```

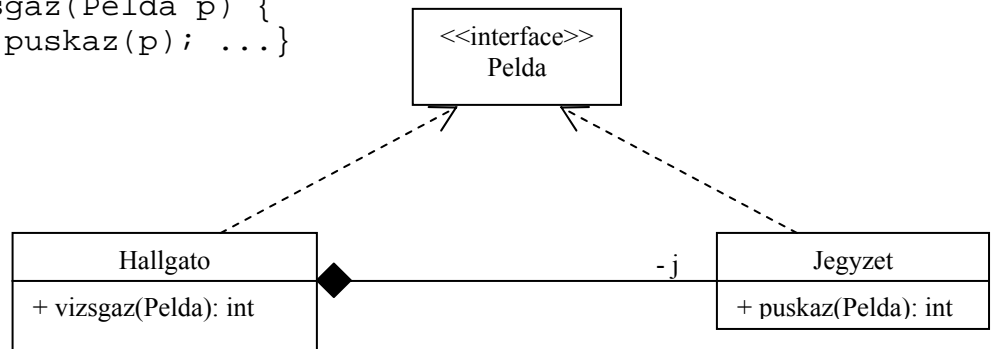
8. Készítsen UML2 komponens diagramot az alábbi leírás alapján! (7 pont)

A SpaceFix magyar műholdkarbantartó-modul felületén van fűrész (ami elvárja, hogy valamit fűrészelhessen), antennabemenet és tápbemenet. Ha a modult szétszedjük, akkor azt látjuk, hogy valójában egy flexből és egy IT-modulból áll. A flex biztosítja a fűrészelést és kapja a tápbemenetet, míg az antennabemenet az IT-modulhoz kapcsolódik. A flex olajozónyílással és vezérlőgombbal is rendelkezik, az IT-modul ebből a vezérlőgombot nyomkodja a flex irányításához.



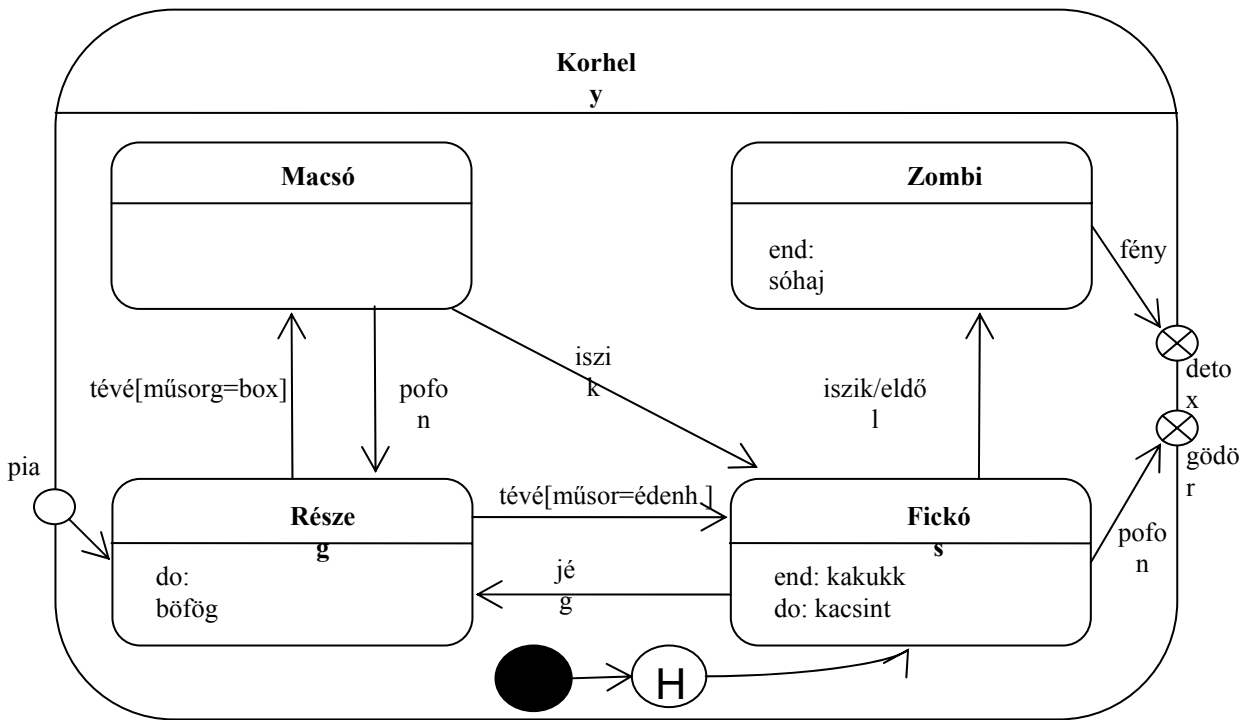
9. Rajzoljon az alábbi programrészletnek megfelelő UML2 osztálydiagramot! Csak olyan jellemzőket ábrázoljon, amely a kódrészletből kiolvasható! (6 pont)

```
public interface Pelda {.....}
public class Hallgato {
    private Jegyzet j;
    public int vizsgaz(Pelda p) {
        int n = j.puskaz(p); ...}
}
```



10. Egészítse ki az alábbi UML 2 állapotdiagramot (state chart) a következő leírás alapján! (8 pont)

Egy objektum **Korhely** állapotában 4 alállapot található (**részeg, fickós, macsó, zombi**). A **korhely** állapotba egy belépési ponton (entry point) léphetünk be (**pia**), és két kilépési ponton (exit point) hagyhatjuk el (**detox, gödör**). Ha nem pia-n lépünk be, akkor abba az állapotba kerülünk, amelyikben utoljára voltunk. Ha nem volt ilyen, akkor a fickósba. A pia-ból a részegbe kerülünk. Ha részeg, folyton bőfög. Ilyenkor, ha tévét néz, és box megy, akkor macsó lesz, ha az éden hotel megy, akkor meg fickós. Fickósan mindig kacsint. Fickósból részeg jég hatására, macsóból részeg pofon hatására lesz. Ha macsó és iszik, akkor fickós lesz. Bármikor, amikor abbamarad a fickósság, akkor kakukkol. Ha fickós és iszik, akkor eldől és zombi lesz. Ha fickós és pofont kap, akkor a gödör kilépési ponthoz jut. Zombi állapotból kilépve sóhajt. Zombiból fény hatására a detox-ba kerül.



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

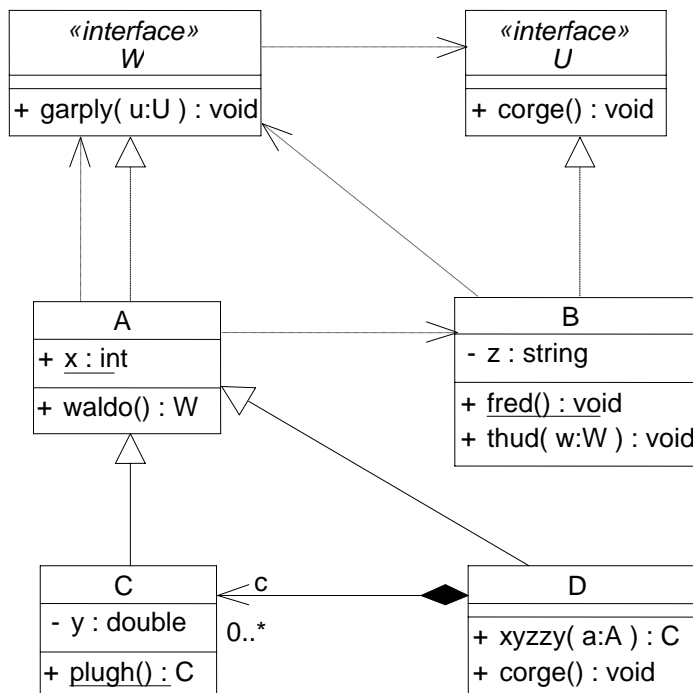
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2011. június 14.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

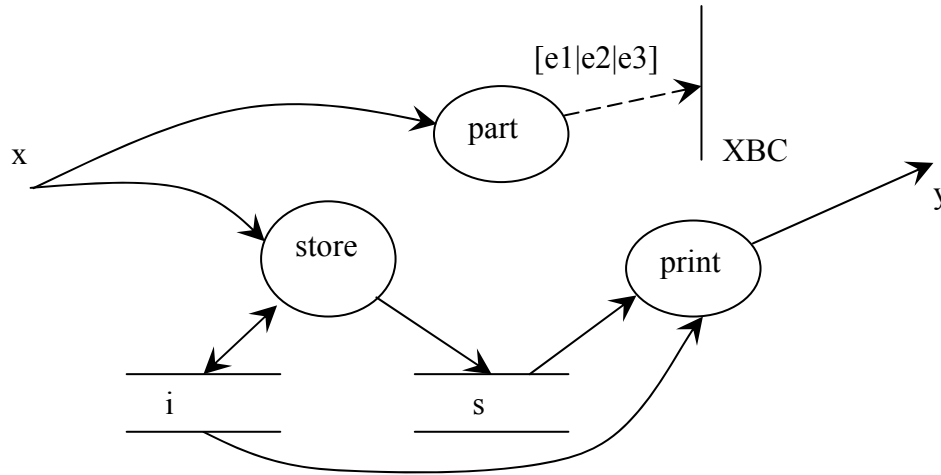
- [A] U interfésze részhalmazza D interfészének, ezért D megvalósítja az U interfészt.
- [B] D xyzyz függvénye visszaadhatja eredményként a paraméterként kapott a objektumot, mert C az A leszármazottja.
- [B] A waldo függvénye nem példányosíthat B típusú objektumot, mert B nem implementálja a W interfészt.
- [E] C plugh függvénye nem módosíthatja A x attribútumát, mert A x attribútuma protected.
- [E] Egy C objektum sok D objektumot tartalmaz, ezért C ismeri D-t.
- [C] B fred függvénye nem módosíthatja a z attribútum értékét, mert B z attribútuma nem protected.
- [E] B thud függvénye meghívhatja egy paraméterül kapott C típusú objektum plugh függvényét, mert C plugh függvénye virtuális.
- [E] C nem függ U-tól, mert C A ősoztálya sem függ U-tól.

2. Egy program (amelynek bemenete x , kimenete y) működését az alábbi XBC nevű állapototábla írja le:

	e1	e2	e3
A1	A1/-	A1/-	A2/s[i++]=x
A2	A3/-	A3/-	A2/s[i++]=x
A3	A3/s[i++]=x	A3/-	A1/y=print(s,i)

ahol e1, e2, e3 események, amelyek a $\text{part}(x)$ függvény lehetséges értékei, s egy 1000 elemű x típusú értékek tárolására szolgáló tömb, i integer index.

Rajzolja fel a program **adatfolyamábráját** ! (7 pont)



3. Az alábbi táblázatban a szoftver fejlesztési folyamat különféle modelljeit adtuk meg, minden modellhez egy indexet rendelve. A jobboldali táblázatban a fejlesztési folyamattal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik modellhez kapcsolhatóak leginkább ! (A fogalom előtti mezőbe írja be a modell indexét !) Ha a fogalom egyik modellhez sem társítható, akkor a mezőbe tegyen X-et ! (3 pont)

1	Ward-Mellor
2	ICOM
3	Vízesés (Waterfall)
4	V
5	Spirál

3	Karbantartás (maintenance)
5	Prototípus (prototype)
2	Erőforrások (resources)
5	Kockázat elemzés (risk analysis)
1	Absztrakció (abstraction)
4	Egységteszt (unit test)

4. A System Specification Diagram (SSD, Rendszer specifikációs diagramm) a JSD melyik lépésében jelenik meg először ? (2 pont)

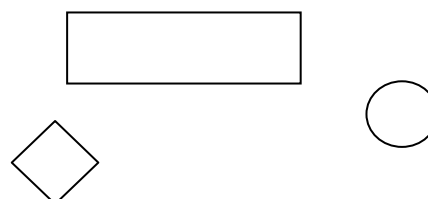
Initial model step

Milyen elemek szerepelnek rajta és azokat hogyan jelöljük ? (4 pont)

Processz

Data stream connection

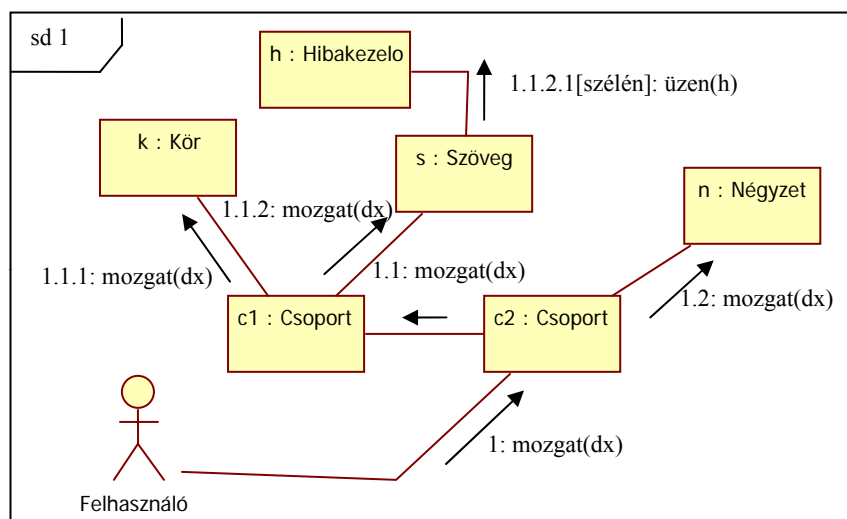
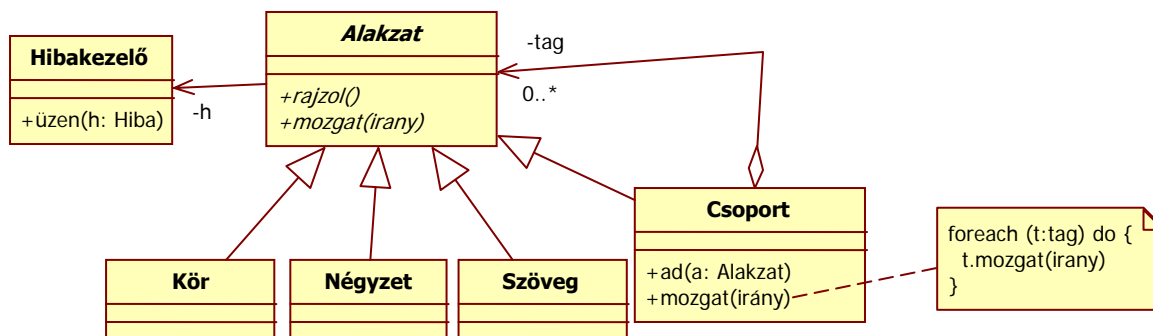
State vector connection



A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Az alábbi UML osztálydiagram segítségével, a szöveg alapján rajzoljon UML 2.0 kommunikációs diagramot! (7 pont)

Egy rajzszerkesztő program modellje az ábrán látható A felhasználó korábban készített egy kört, egy szöveget és egy négyzetet. A kört és a szöveget csoportba foglalta, majd az így kapott csoportot és a négyzetet újabb csoportba tette. (Az ezzel kapcsolatos metódushívásokat ne jelölje az ábrán!) Jelölje viszont a kommunikációs diagramon, ahogy a felhasználó meghívja az utolsó csoporton a mozgat() tagfüggvényt! A szöveg, ha a képszelére kerül, értesíti a hibakezelő objektumot. Alkalmazzon hierarchikus számozást!



6. Mik a hasonlóságok az adatfolyam (DFD) és a use-case (UC) modellek között? (3 pont)

funkcionalitást írnak le

terminátor - actor

folyamat (process) - use-case

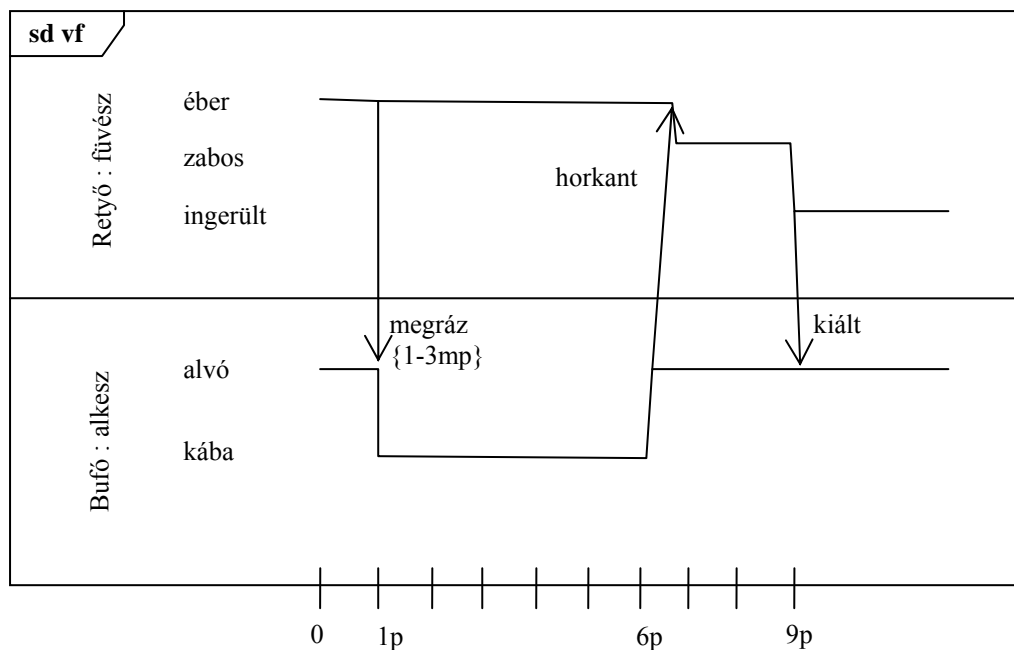
Tételezzük fel, hogy az A folyamat tartalmazza a B folyamatot.

Hogyan ábrázoljuk ezt az adatfolyam modellben és a use-case diagrammon? (3 pont)

DFD - B az A processzt kifejtő DFD-n lesz egy process

UC - B-t <<include>>-olja A

7. Az alábbi történet alapján rajzoljon UML 2.0 időzítési diagramot (timing diagram)! (8 pont)
 Retyő, a fűvész és Bufó, az alkesz barátok. A történet kezdetekor Retyő éber, Bufó alvó állapotban van. 1 perc után Retyő (1-3 másodpercig) megrázza Bufót, mire az kába állapotba kerül. Bufó 5 perc elteltével visszaalszik, és horkant, amire Retyő zabos lesz. 3 perc múlva Retyő ingerültté válik és Bufóra rákiált, aki erre nem reagál.



8. Definiáljunk egy adatszerkezetet a következő Java osztállyal !

```
public class BinarisFa {
    public String i;
    public BinarisFa bal;
    public BinarisFa jobb;
}
```

Készítse el a fenti adatszerkezet DTD leírását ! (5 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE BinarisFa [
    <!ELEMENT BinarisFa (i, bal, jobb)>
    <!ELEMENT jobb (BinarisFa?)>
    <!ELEMENT bal (BinarisFa?)>
    <!ELEMENT i (#PCDATA)>
]>
```

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

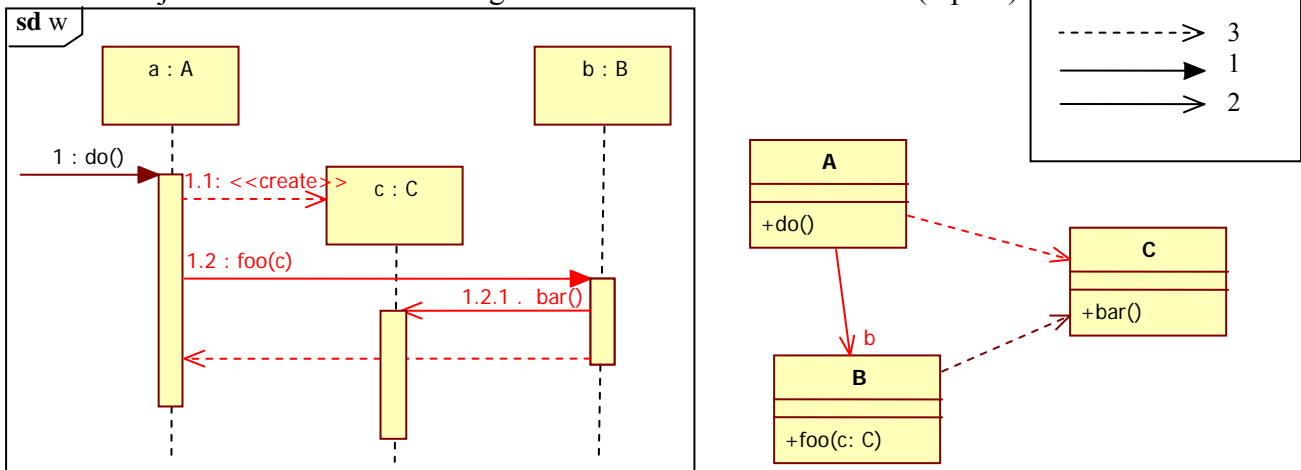
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2011. december 20.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Az ábrán egy UML2 osztálydiagram és egy UML2 szekvenciadiagram látszik. A két diagram szemantikailag összefügg, de hiányos. Rajzolja be a hiányzó jelölő-elemeket! Ahol lehet, lássa el őket feliratokkal is! A felhasználható jelölőelemek és számosságuk a mellékelt keretben látható. (6 pont)



2. Töltse ki az alábbi kódrészlet hiányzó részeit a szabványos Java API elemeivel úgy, hogy mind szintaktikailag, mind szemantikailag helyes megoldás szülessen! (1 pont)

```
FileInputStream fis = new FileInputStream("test.txt");
```

```
InputStreamReader foo = new InputStreamReader . (fis);
char c = foo.read( );
```

Adja meg az egyenlőségjel utáni kódrészlet módosított változatát, hogy olyan fájlból tudjunk beolvasni, aminek a létrehozásakor a *GZIPOutputStream* osztályt használtuk! (1 pont)

```
new InputStreamReader(new GZIPInputStream(fis)) ..
```

Milyen szál állapot-értékkel tér vissza a Javában a *Thread.currentThread().getState()*? (2 pont)

RUNNABLE (esetleg RUNNING)

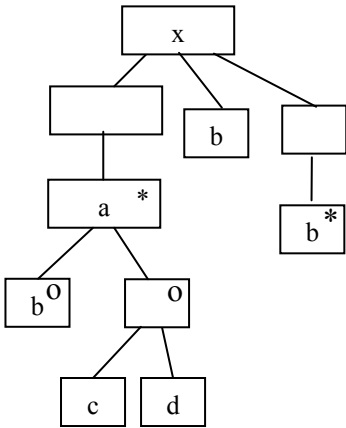
Jelölje (karikázza be) a Java nyelvvel kapcsolatos állítások igazságtartalmát! (2 pont)

H Egy változó statikus típusa nem lehet absztrakt osztály.

H *JScrollPane*-be csak olyan *SWING* komponenst szabad tenni, aminek nincs saját görgetősávja.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total >= 0

3. Adott az alábbi ELH-ábra.



```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x      ①>
  <!ELEMENT a      ②>
  <!ELEMENT b      (#PCDATA)>
  <!ELEMENT c      (#PCDATA)>
  <!ELEMENT d      (#PCDATA)>
]>
<x>
  <b>a</b>
  <b>cd</b><b>*</b>
  <b>b,b*</b>
</x>
```

Az alábbiak közül ① helyén melyik válasz írja le helyesen az ELH-ábrát ? (3 pont)

- (-, b, b*)
- (* , b? , b*)
- (a* , b+)
- ((a, b, b)?)
- (a, b((c | d) , b, b*)
- (a* , b, b*)
- (a, b | (c, d) , b*)
- (, b+b)
- egyik sem

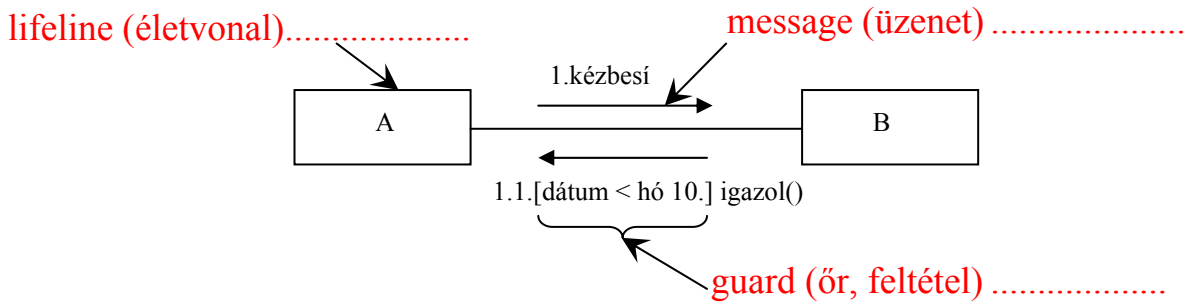
Az alábbiak közül ② helyén melyik válasz írja le helyesen az ELH-ábrát ? (2 pont)

- (b+ (c, d))
- ((b+ (c, d) *) , b?)
- (b* (c | d))
- (b, (c | d))
- (b | (c, d))
- ((b, c) | (b, d)?)
- ((b | (c, d) *) , b+)
- ((b | (c, d)) , b, b*)
- egyik sem

A baloldalon álló XML adatszerkezet (2 pont)

- mint XML szerkezet szintaktikai hibás
- megfelel az ELH-ábrának (érvényes)
- nem felel meg az ELH-ábrának (érvénytelen)
- érvényessége nem dönthető el

4. Az alábbi ábrán három UML2 modell elemet megjelöltünk. Adja meg elemenként, hogy az melyik UML2 meta-modell elem példánya ! (3 pont)



5. Az interpreter architektúra melyik komponensének feleltethető meg egy állapottábla ? (2 pont)

Pseudocode

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

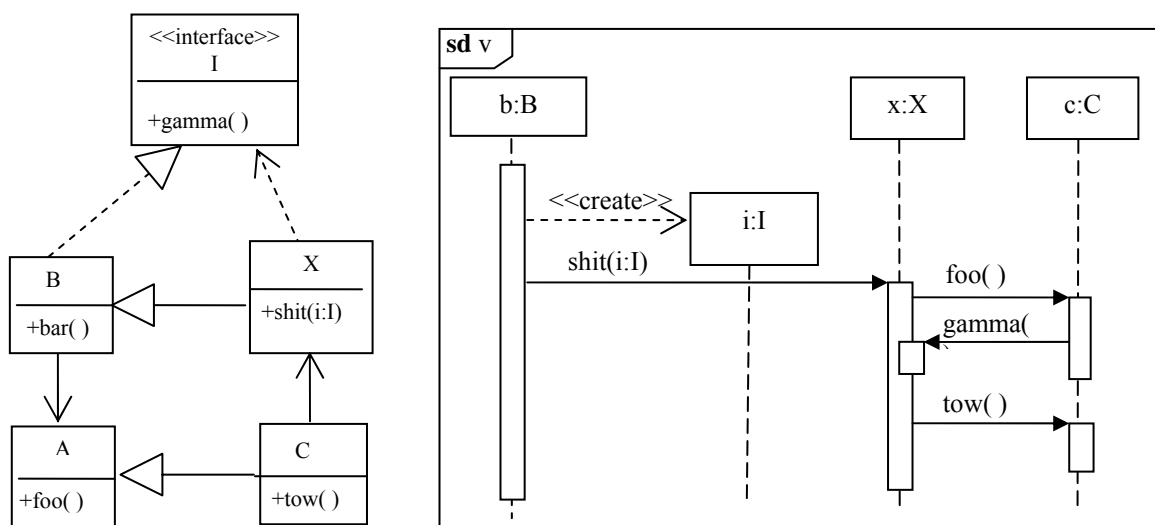
6. Sorolja fel, hogy a Scrum módszertanban kik tartoznak a disznók (pigs) közé ! (3 pont)

Product owner, Scrum Master, Team

A szoftverfejlesztés „spirális modelljé”-nek a 1. szektorában mi a megoldandó feladat ? (2 pont)

- | | | | |
|-------------------------------------|---------------------|--------------------------|-------------------------|
| <input type="checkbox"/> | kockázatok becslése | <input type="checkbox"/> | projekt definiálása |
| <input type="checkbox"/> | tervezés | <input type="checkbox"/> | fejlesztés és validálás |
| <input checked="" type="checkbox"/> | célok kijelölése | <input type="checkbox"/> | specifikálás |

7. Az ábrán egy UML2 osztálydiagram és egy UML2 szekvenciadiagram látszik. A két diagram szemantikailag összefügg. Az osztálydiagramot hibátlannak tekintve milyen hibákat, ellentmondásokat talál a szekvenciadiagramban? Sorolja fel a hibákat ! (6 pont)



Interfész nem példányosítható

B nem látja **X**-et, ezért annak `shit(i:I)` metódusa nem hívható

X nem láthatja **C** `tow()` metódusát

8. A kulcs felhasználásával jellemezze a Java nyelvvel kapcsolatos állításokat ! (3 pont)

- | | |
|--|--------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (++ -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+++) |
| E - egyik tagmondat sem igaz | (--) |

[B] Minden `List` interfészt megvalósító objektum értékül adható `Set` típusú változónak, mert a `Set` minden metódusa megtalálható a `List`-ben is.

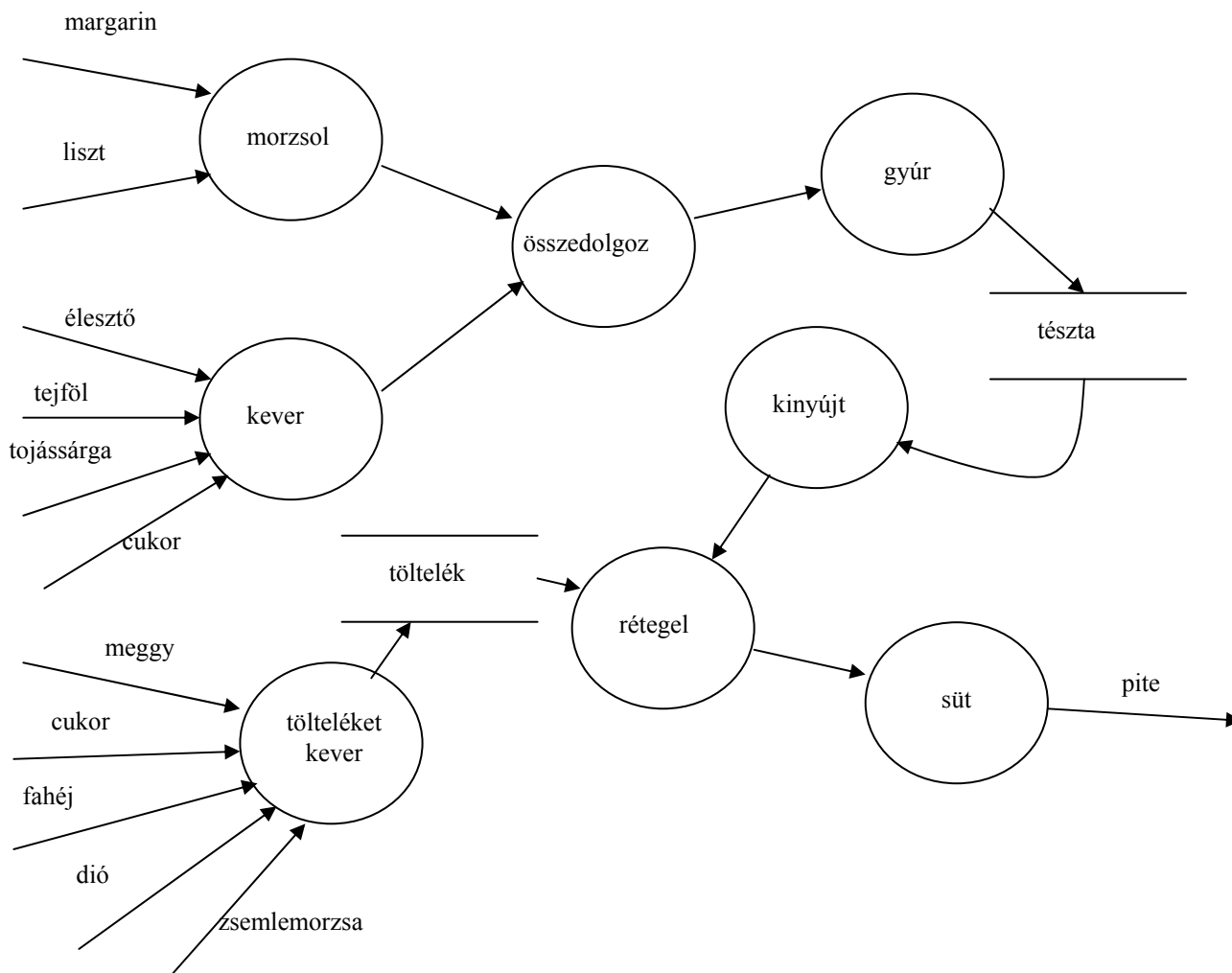
[D] Egy objektum `wait` metódusát csak az objektum monitorában tartózkodó szál hívhatja meg, mert a `wait` hívása során a szál kilép az objektum monitorából.

[B] Absztrakt osztálynak nem lehet `final` metódusa, mert az osztályból nem lehet példányt létrehozni.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

9. Izidor meggyes pitét süt karácsonyra. Rajzoljon adatfolyam ábrát (context diagram NEM KELL!!), amely specifikálja a meggyes pite elkészítésének folyamatát ! (8 pont)

A margarint a liszttel elmorzsoljuk. Az élesztőt a tejfölben tojássárgájával és cukorral jól elkeverjük, majd a lisztes margarinnal összedolgozzuk. Meggyúrjuk a tésztát, amit fél óráig hűvös helyen pihentetünk. A meggyet cukorral, fahéjjal, dióval és a zsemlemorzsával keverve elkészítjük a tölteléket. A tésztát kinyújtjuk, majd a töltelékkel rétegelve (tészta, töltelék, tészta) 180 fokon 20-25 percig sütjük.



10. A Subversion munkapéldányát (working copy) két tényező jellemzi:

- a munkapéldány az utolsó letöltés (check out/update) óta változott-e vagy sem (changed/unchanged)
- a repositoryban levő revízió változott-e a munkapéldány letöltése óta (out of date/current)

Az alábbi táblázatba írja be, hogy a két változótól függően mi történik a commit (check in) esetén (4 pont)

	out of date	current
unchanged	-	-
changed	out of date error	publish

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

3. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (4 pont)

- I H** Egy objektum referenciáját tartalmazó változón csak olyan metódus hívható meg, amilyen a változó statikus típusában is szerepel.
- I H** Egy változó dinamikus típusa nem lehet absztrakt osztály.
- I H** Előfordulhat, hogy két szál (T1 és T2) ugyanazon objektum ugyanazon synchronized metódusát futtatva T1 T2 sorrendben lép be, de T2 T1 sorrendben lép ki.
- I H** Ha a t tömböt paraméterként átadjuk egy metódusnak, amelyik a tömb egy elemének új értéket ad, akkor az eredeti tömbben is módosul az érték.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total >= 0

4. Az alábbi táblázatban a fontosabb szoftver architektúra típusokat adtuk meg, minden típushoz egy indexet rendelve. A jobboldali táblázatban szoftver architektúrákkal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik architektúra típushoz kapcsolhatóak leginkább ! (A fogalom előtti mezőbe írja be a típus indexét !) Ha a fogalom egyik típushoz sem társítható, akkor a mezőbe tegyen X-et ! (3 pont)

1	esemény alapú implicit hívást használó (event-based implicit invocation)
2	csővezeték+szűrők (pipes and filters)
3	tábla (blackboard)
4	interpreter
5	szolgáltatás orientált architektúra (SOA)

2	batch (kötegelés)
4	virtuális gép (virtual machine)
1	regisztráció (registration)
3	központi adatbázis (repository)
5	kötés (binding)
1	szórás (broadcast)

5. Jelölje be az alábbi táblán, hogy az egyes szerződéses feltételek megszegése esetén melyik fél a hibás! (3 pont)

	kliens	szerver
előfeltétel (precondition)	X	
invariáns (invariant)		X
utófeltétel (postcondition)		X

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total >= 0

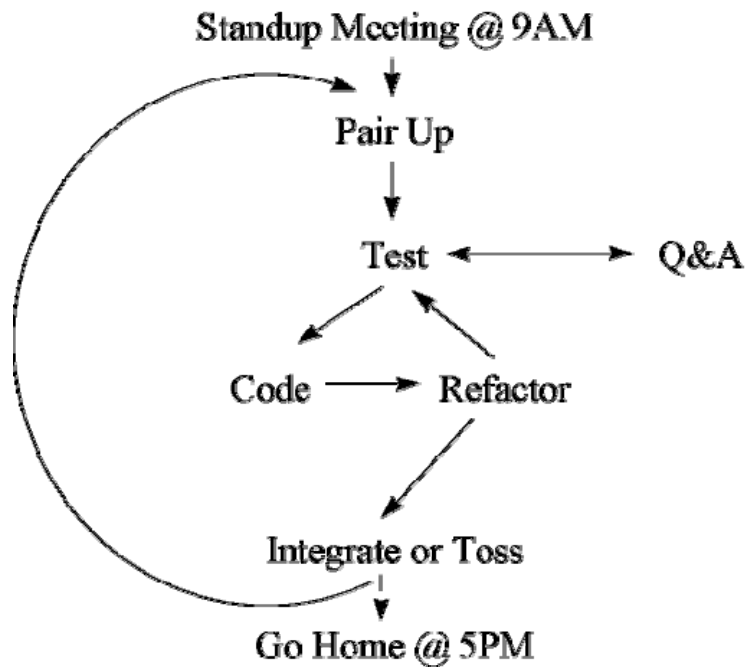
6. Készítse el azt az állapottáblát, amely megfelel az alábbi DTD-vel definiált adatszerkezetnek ! A tábla az 1. állapotban kezdődik, és segítségül beírtuk az a illetve b esetén következő állapotokat. Kötőjellel (-) jelölje, ha egy elem, egy állapotban nem következhet! (5 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a|(b+,c?)),c)*>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
```

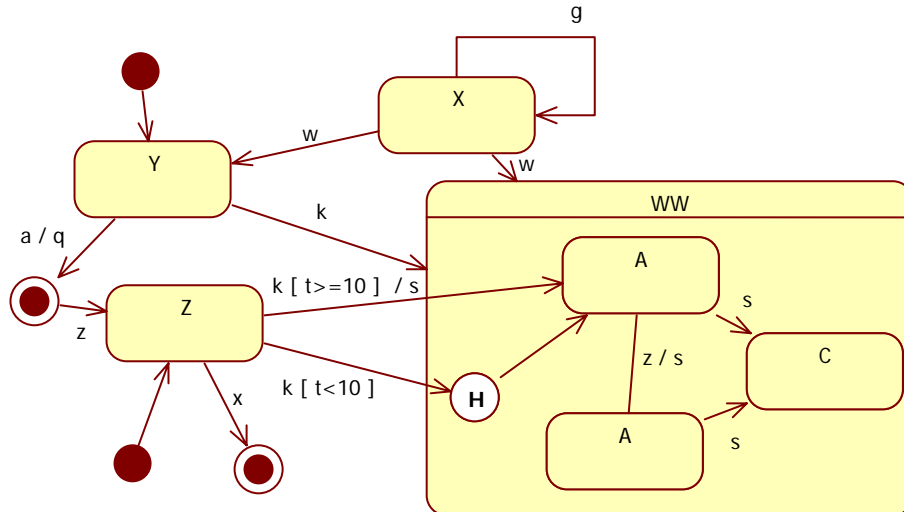
	a	b	c
1	2	3	-
2	-	-	1
3	-	3	4
4	2	3	1

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

7. Írja le (folyamatábrával) az XP módszertan szerinti fejlesztésben egy nap történéseit! (6 pont)



8. Milyen szintaktikai és szemantikai hibák találhatók az alábbi UML2 állapot-diagramon (state-chart)? (8 pont)

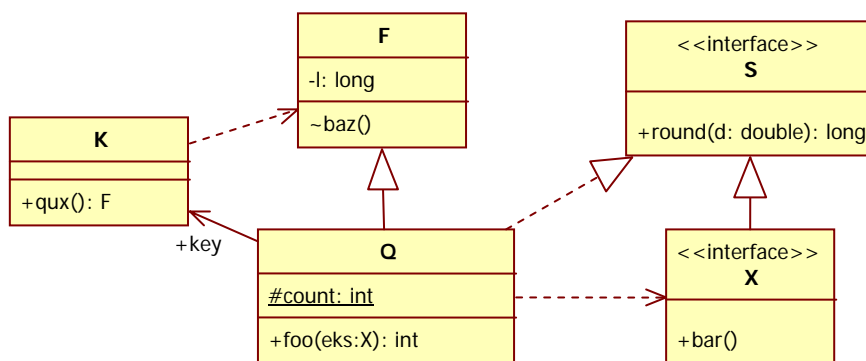


- Két kezdőállapot
- Ugyanarra az eseményre két állapotba is mehet
- Végállapotból kilépünk
- Nincs kezdőállapot
- Ugyanolyan nevű állapotok
- Nincs nyíl
- Csak kimenő állapot
- Csak bemenő állapot

9. Az alábbi Java kódrészletek alapján rajzoljon UML2 osztálydiagramot! (8 pont)

```

class Q implements S extends F {
    protected static int count;
    public K key;
    public int foo(X eks) {
        eks.bar();
    }
}
interface X extends S {
}
interface S {
    long round(double d);
}
class F {
    private long l;
    long baz() { return l; }
}
class K {
    F qux() { return new F(); }
}
    
```



10. A mellékelt táblázatban jelölje be, hogy a felsorolt fogalmak az UML2 4-rétegű meta-modell szerkezetének melyik rétegébe tartoznak! (4 pont)

	M0	M1	M2	M3
Izidor	X			
Actor			X	
State			X	
Autó		X		
Barnabás	X			
UseCase			X	
Ügyfél		X		
Ember		X		

Eredmények értékelése:

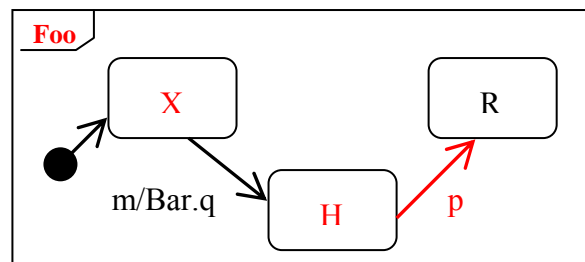
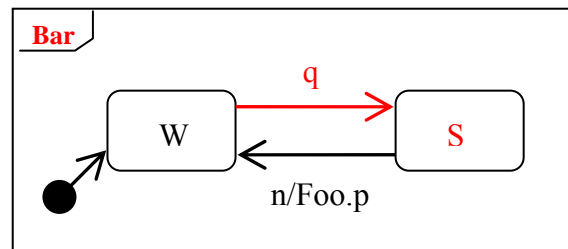
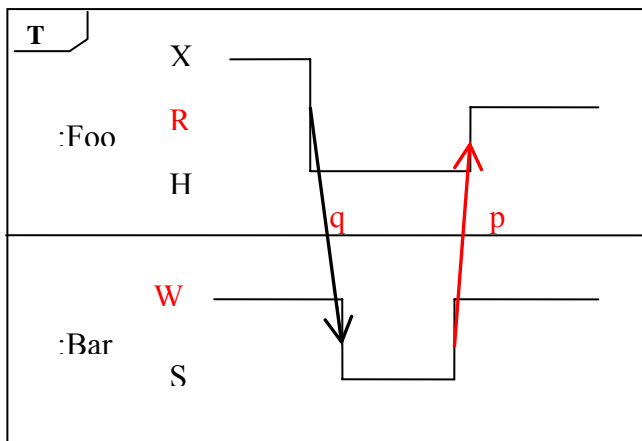
Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2012. január 17.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Izidor elkészített három – szemantikailag összefüggő, hibátlan – UML2 diagramot. Miközben vacsorázott, a kis huncut Zsilip (Izidor öccse) kiradírozott néhány elemet (szöveg, vonal, nyíl stb.) Segítségül Izidornak visszaállítani az ábrákat! Rajzold be a hiányzó elemeket és feliratokat, úgy, hogy az ábrák helyesek és összefüggőek legyenek! (5 pont)



2. Sorrendben adja meg, hogy a *foo* metódus teljes lefutása során a futató szál milyen állapotokat vesz fel! (Segítségül megadtuk az induló állapotot.) Tételjeze fel, hogy a metódus meghívásakor egy másik szál ugyanazon objektum *bar* metódusát hajtja végre! (5 pont).

```

synchronized void bar() {
    ...
}
synchronized public void foo() {
    try {
        this.wait();
    } catch (Exception e) { }
}
    
```

1	RUNNABLE (RUNNING)
2	BLOCKED
3	RUNNABLE (RUNNING)
4	WAITING
5	BLOCKED
6	RUNNABLE (RUNNING)
7	

3. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (4 pont)

I H Egy változó dinamikus típusának nem lehet *abstract* metódusa.

I H Egy változó statikus típusának nem lehet *final* metódusa.

I H Egy *interface* típusnak csak egyetlen közvetlen őse lehet.

I H Az alábbi kódrészlet futtatása végén a *b* változó értéke *true*.

```
String s1 = new String("Hello");
String s2 = s1;
s2 += " world";
boolean b = (s1 == s2);
```

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total >= 0

4. Adott az alábbi dekorált XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE z [
  <!ELEMENT z      (f+,x)>
  <!ELEMENT f      ANY>
  <!ELEMENT x      (#PCDATA)>
  <!ELEMENT k      (#PCDATA ① z)*>
]>
<z>
  <f><z><f>
    ②
    <x></x></z>
    f/</f><x>/x</x>
</z>
```

Mit írna ① helyére, hogy a DTD jól formált legyen? (2 pont)

- +
- *
- |
- ”, ” (vessző)

Feltételezve, hogy a DTD jól formált, mi állhat ② helyében, hogy az XML érvényes legyen? (4 pont)

- <x><k>/x</k></f>x</f>
- <z><k></x></f>x</k></z>
- <x><k></x></f>x</k></f>
- <x>x/x</x></f>x</f>
- <k><x></f>x</k></z>
- <f><k>x</k></f>x</f>
- <f><x>f</x>/x</f></f>
- <z><k>x</k></z>

5. A mellékelt állapotgéppel specifikált programot az interpreter architektúra szerint kívánjuk megvalósítani. Az állapottábla alapján specifikálja a

	A	B	C
S1	S1/X	S1/X	S2/Y
S2		S2/Y	S3/Y
S3	S1/X	S3/Y	

“control state” értékeit! (1 pont): **S1|S2|S3**

“pseudocode” utasításait! (3 pont): **S1/X, S2/Y, S3/Y**

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. A kulcs felhasználásával jellemezze az alábbi Java kódrészlettel kapcsolatos állításokat ! (4 pont)

- A - csak az első tagmondat igaz (+ -)
- B - csak a második tagmondat igaz (- +)
- C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
- D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
- E - egyik tagmondat sem igaz (- -)

```
public class X {
    private long l;
    private static void foo() { System.out.println("hello"); bar(); }
    public void bar() { System.out.println("world"); }
    public void bar(X x) { System.out.println("world"); x.l=10; }
    protected X copy() { return new X(); }
    public void do() { bar(copy()); }
}
```

- [A] A *foo()* metódusban hibás a *bar()* hívása, mert privát láthatóságú metódusból nem hívhatunk publikusat.
- [A] A *do* metódus fordításkor hibát jelez, mert publikus metódus paraméterének nem lehet protected metódus visszatérési értékét adni.
- [E] *X* osztály nem példányosítható, mert nincs paraméter nélküli konstruktora.
- [B] *X* csak egyetlen másik osztálynak lehet a közvetlen őse, mert a Javában nincs többszörös öröklés.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

7. Jellemezzünk egy (K,V) párokból álló Map-et – ahol K egy kulcs, V egy 0-nál nagyobb egész érték – az alábbi műveletekkel! Adja meg az algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveletet. (6 pont)

- NEW() új (üres) map-et hoz létre.
- PUT(m,(k,v)) az *m* map-hez hozzáveszi a *k,v* párt, ha *k* nem szerepelt a map-ben. Ha a map-ben már van *k*,akkor a hozzá tartozó *v*-t az új *v*-vel helyettesíti, ha az kisebb a map-ben szereplőnél.
- GET(m,k) megadja az *m* map-ben a *k* kulcshoz tartozó *v*-t. Ha a map-ben a megadott *k* nem szerepel, akkor az eredmény 0.
- MIN(m) az *m* map-ben szereplő legkisebb *v*. (üres map esetén nulla)

GET(NEW(),z) = 0 -0.5 pont, ha hiányzik

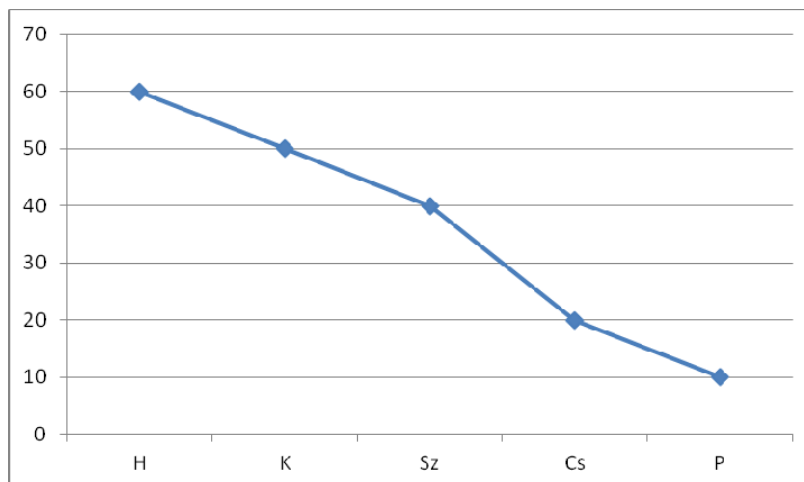
MIN(NEW()) = 0 -0.5 pont, ha hiányzik

GET(PUT(m,(k,v)),z) = if (k == z && (GET(m,k) == 0 || GET(m,k) > v)) v else GET(m, z) 3 pont

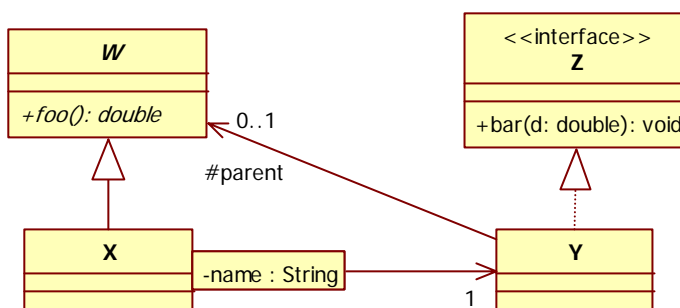
MIN(PUT(m,(k,v))) = if (MIN(m) == 0 || MIN(m) > v) v else MIN(m) 3 pont

8. Adott a következő Scrum product backlog (teendők listája). Feltételezve, hogy a munka pontosan a terveknek megfelelően haladt, rajzolja fel a végső burndown chart-ot (napi eredménykimutatás)! (6 pont)

	H	K	Sz	Cs	P
brummogás	5	4	5	1	
tekerés			11	2	6
számolás	3	6	4	4	
okoskodás	2			3	4



9. Izidor karácsonyra kapta az alábbi szép UML2 osztálydiagramot, azzal, hogy írja meg az ábrán látható osztályok és interfészek Java forráskódját. Izidor még nem készült fel a Szofttech vizsgára, ezért nem megy neki a dolog. Segíts neki, és készítsd el te a kódot! Minden, az ábrából következő metódust és attribútumot vegyél fel! Ha szükséges, használd a szabványos Java API osztályait is! Minden osztály publikus. A metódusok törzse legyen üres! (10 pont)



```

public abstract class W {
    public abstract double foo();
}
public class X extends W {
    private Map<String, Y> hm;
    public double foo() {}
}
public class Y implements Z {
    protected W parent;
    public void bar(double d) {}
}
public interface Z {
    public void bar(double d);
}
    
```

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból

2012. május 22.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (7 pont)

- H** *for* ($S \ x : z$) fejlécű *for* ciklusban a z referencia csak tömbre vagy a JDK-val szállított gyári kollekciók példányaira referálhat.
- H** egy szál egyszerre csak egy objektum monitorában tartózkodhat.
- H** *synchronized* kulcsszó használatával elkerüljük a deadlock kialakulását.
- H** szálak nem képesek saját magukat közvetlenül *waiting* állapotból *notify*-jal felébreszteni.
- H** előfordulhat, hogy két szál (T1 és T2) ugyanazon objektum ugyanazon *synchronized* metódusát futtatva T1 T2 sorrendben lép be, de T2 T1 sorrendben lép ki.
- H** egy változó statikus típusa nem lehet a változó dinamikus típusának leszármazottja.
- H** egy metódust el lehet látni egyszerre *abstract* és *final* módosítóval is.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total ≥ 0

2. Mik a hasonlóságok az adatfolyam (DFD) és a use-case (UC) modellek között? (3 pont)

funkcionalitást írnak le
terminátor - actor
folyamat (process) - use-case

Tételezzük fel, hogy az A folyamat tartalmazza a B folyamatot.

Hogyan ábrázoljuk ezt az adatfolyam modellben és a use-case diagrammon? (2 pont)

DFD - B az A processzt kifejtő DFD-n lesz egy process

UC - B-t <<include>>-olja A

3. Sorolja fel a Rational Unified Process (RUP) életciklus modelljében szereplő „támogató munkafolyamatokat” (supporting workflows)! (3 pont)

konfigurációs menedzsment
menedzsment
környezet

4. Töltse ki az alábbi kódrészlet hiányzó részeit a szabványos Java API elemeivel úgy, hogy mind szintaktikailag, mind szemantikailag helyes megoldás szülessen! (1 pont)

```
FileOutputStream fis = new FileOutputStream("test.txt");
```

```
OutputStreamWriter foo =  
    new OutputStreamWriter (fis);  
foo.write('A'); foo.write('B');
```

Adja meg az egyenlőségjel utáni kódrészlet módosított változatát, hogy olyan fájlba tudjunk írni, aminek a beolvasásához a *GZIPInputStream* osztályt kell használnunk! (1 pont)

```
new OutputStreamWriter(new  
                        GZIPOutputStream(fis));
```

5. A specifikáció célja a követelményeknek eleget tevő rendszer formális leírása. Milyen három fontos nézőpontból készítjük a leírásokat? (3 pont)

funkcionalitás

szerkezet (struktúra).....

dinamika (viselkedés)

6. Mik a konfigurációs menedzsment fő folyamatai? (4 pont)

Storage Configuration Items

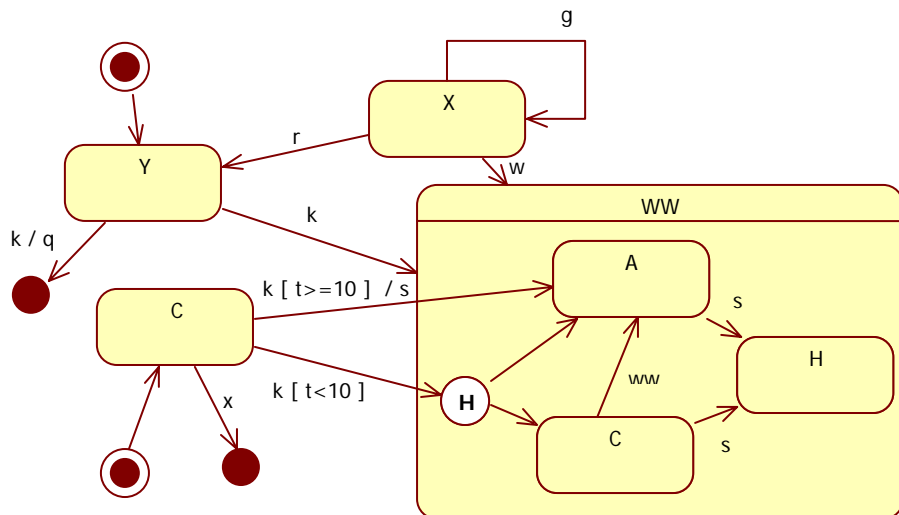
Build Management

Change Management

Release Management

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

7. Milyen szintaktikai és szemantikai hibák találhatók az alábbi UML2 állapot-diagramon (state-chart)? (8 pont)



- Végállapotból kilépünk
- Kezdőállapotba belépünk
(kezdő és végállapotok fel vannak cserélve)
- Több kezdőállapot is van
- WW-ben nincs kezdőállapot
- WW-ből nem jutunk végállapothoz
- Y-ból két k kilépés is van
- WW-ben a histpryból kétfele is megyünk
- X forrás (nincs belépő átmenet)

8. Nevezze meg sorrendben a CMM (Capability Maturity Model) szintjeit! (5 pont)

1. **kezdetleges (initial)**
2. **ismétlődő (repeatable)**
3. **definiált (defined)**
4. **irányított (managed)**
5. **optimalizált (optimizing)**

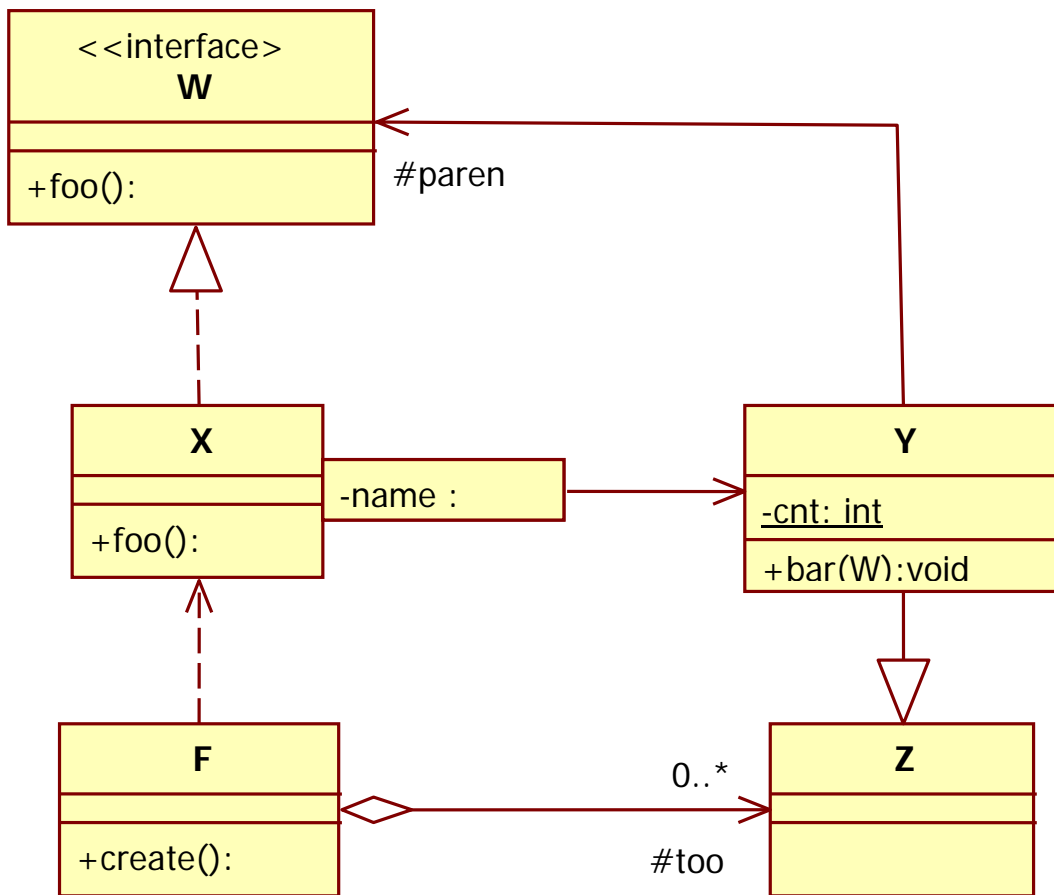
9. Adja meg, hogy az alábbi UML2 kollekció-jellemzők definiálása esetén melyik *java.util*-beli kollekció-interfészt használhatjuk! Adjon meg mindegyikhez egy tipikus *java.util*-beli megvalósítást is! (5 pont)

UML	Java util interfész	Java util megvalósítás
ordered	List	ArrayList, Vector
unique	Set	HashSet
unique, ordered	SortedSet	TreeSet
qualified	Map	HashMap, Hashtable
qualified, ordered	SortedMap,	TreeMap

10. Rajzoljon UML2 osztálydiagramot az alábbi Java kódrészlet alapján! (8p)

```

public interface W {
    public double foo();
}
public class X implements W {
    private java.util.Map<String, Y> hm;
    public double foo() {return 1; }
}
public class Z {}
public class Y extends Z {
    protected W parent;
    public void bar(W p) { parent = w; }
    static private int cnt;
}
public class F {
    protected java.util.List<Z> tool;
    public X create() { return new X(); }
}
    
```



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból
2012. június 5.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

1. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (5 pont)

- I** **H** Statikus attribútumot csak statikus metódusból lehet elérni.
- I** **H** Példánymetódust kivétel nélkül csak példánymetódusból hívhatunk.
- I** **H** Egy osztály lehet akkor is absztrakt, ha nincs absztrakt metódusa.
- I** **H** A *java.lang.String* osztálynak vannak a string értékét (tartalmát) megváltoztató metódusai.
- I** **H** A primitív típusokhoz tartozó csomagoló osztályok (wrapper classes) nem változtathatók (immutable)

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total ≥ 0

2. Adja meg, hogy az alábbi állítások melyik szabványos Java kivételosztályra igazak! Ha többre is igaz, akkor az öröklési hierarchiában legmagasabban levő (ős)osztályt jelölje meg! (6 pont)

A	Throwable	B	Exception	C	InterruptedException
D	RuntimeException	E	NullPointerException	F	Error

- [A]** Minden kivételként eldobható osztály ősoosztálya.
- [F]** JVM szintű rendszerhiba
- [F]** Elkapása nem kötelező, nehéz rá felkészülni, nehéz lekezelni.
- [B]** Programhiba
- [D]** Elkapása nem kötelező, "zajt" vinne a kódba.
- [A]** Az *initCause* metódusával be lehet állítani az okozó kivételt

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total ≥ 0

3. Minek az ellenőrzésére irányulnak az alábbi tesztelések? (4 pont)

Rendszerteszt (System test)	Specifikáció
Validálás (Validation)	Követelmények

4. A `Java.util.Stack<E>` osztály metódusai közötti kohézió milyen típusú? (2 pont)

Kommunikációs (ugyanazon adatszerkezeten operálnak)

5. Mi a szerepe a JUnit-ban az alábbi annotációknak? (2 pont)

@BeforeClass	Tesztosztály közös inicializálása – erőforrások lefoglalása
@After	Egyedi tesztet lefutása után erőforrások elengedése

Egy JUnit teszt eredménye *pass* vagy *fail* vagy *error* lehet. A *pass* a tesztben foglalt állítás beteljesülését jelenti. Mi a jelentése a másik két eredménynek? (2 pont)

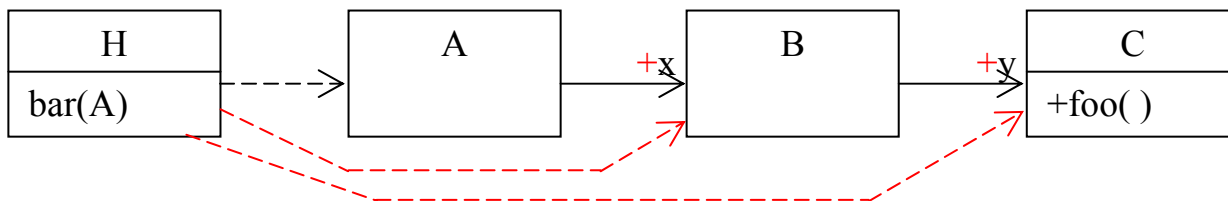
fail	Az állítás nem teljesül.
error	A tesztmetódus általa nem kezelt kivételt dob.

6. Az UML2 Activity diagram egy másik UML2 diagram speciális esetének tekinthető. Melyik ez a diagram? Hasonlítsa össze a két diagramot! (3 pont)

Activity diagram	Állapot..... diagram
state – tevékenységet hajt végre	state (eseményre vár)
transition – a tevékenység befejezésekor (belső) automatikusan	transition (külső esemény hatására)

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

7. Adott az alábbi UML2 diagram, ahol a H osztály **bar(A)** metódusát a következőképp implementáljuk:
`public void bar(A a) {a.x.y.foo();}`

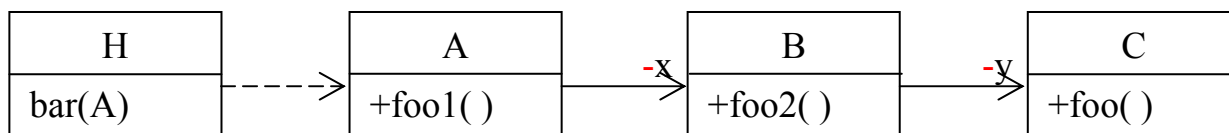


A **bar(A)** metódus implementációjára figyelemmel egészítse ki a diagramot ! (2 pont)

Milyen tervezési problémákat ismer fel? (2 pont)

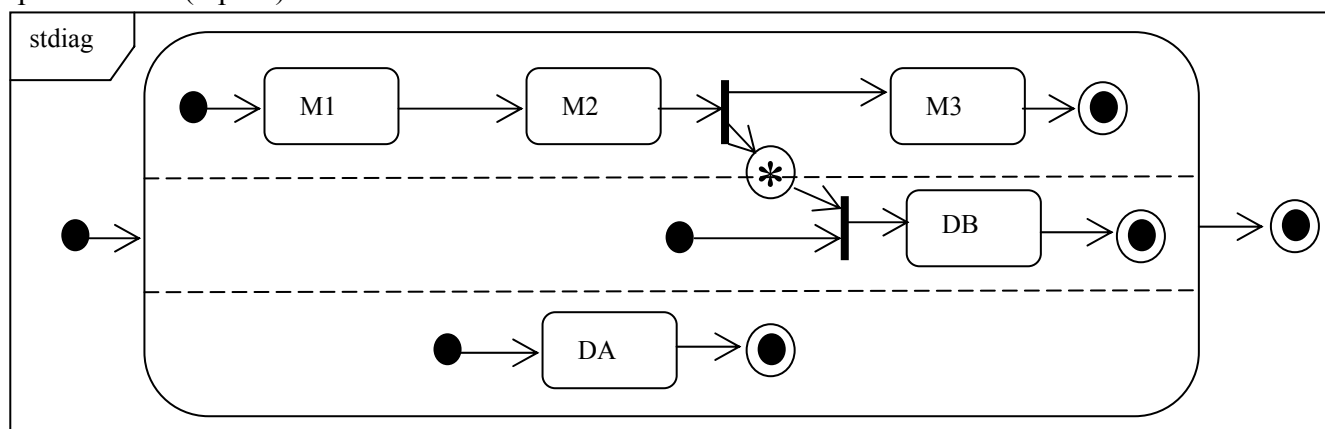
Ellentétes a Demeter-törvénnyel (H össze van csatolva mindegyik osztállyal, A, B gyengén kohézív.)

Rajzolja fel a jól tervezett változat UML2 diagramját, és adja meg a metódus(ok) implementációját! (6 pont)

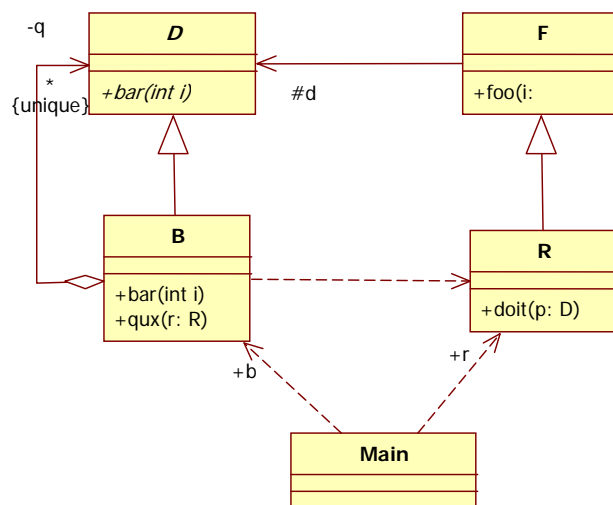
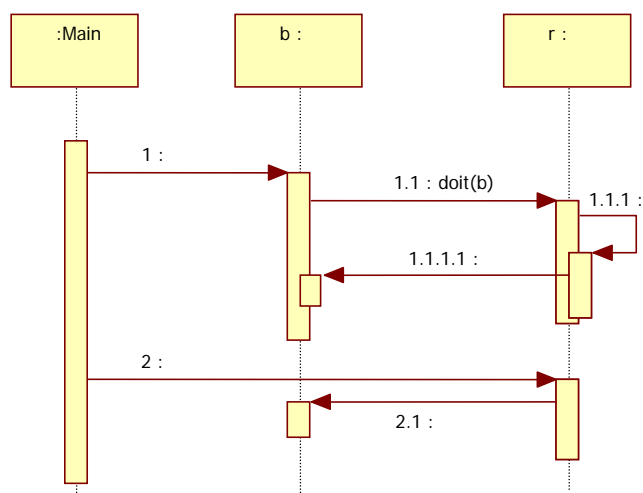


```
public void bar(A a) {a.foo1();}
public void foo1() {x.foo2();}
public void foo2() {y.foo();}
```

8. Egy tantárgy teljesítéséhez három mérést (M1, M2, M3) kell elvégezni, és két házi dolgozatot kell beadni (DA, DB). A méréseket szigorúan szám szerinti sorrendben kell elvégezni, a két dolgozat tetszőleges sorrendben készíthető el. A DB dolgozat csak az M2 mérés elvégzését követően készíthető el. Rajzoljon UML2 állapot modellt! (8 pont)



9. Java nyelven implementálja az alábbi osztály és szekvenciadiagramon megtervezett osztályokat és metódusokat! A Main osztály működése a main metódusban valósul meg, ami a futása elején létrehoz egy B és egy R objektumot (ez az ábrán helyhiány miatt nincs jelölve). (8 pont)



```

abstract class D {
    public abstract void bar(int i);
}
class B extends D {
    private Set<D> q;
    public void bar(int i) {...}
    public void qux(R r) {
        r.doit(this);
    }
}
class F {
    protected D d;
    public void foo(int i) {
        d.bar(i);
    }
}
class R extends F {
    public void doit(D p) {
        d = p;
        foo(3);
    }
}
public class Main {
    public static void main(String[] args) {
        B b = new B();
        R r = new R();
        b.qux(r);
        r.foo(7);
    }
}
  
```

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

3. JUnittal tesztelni akarjuk a nem kezelt kivételt. Írjon olyan tesztmetódust, amely akkor fut le sikeresen, ha nem kezelt ArithmeticException (pl. nullával osztáskor) kivételt dob ? (2 pont)

```
@Test(expected=ArithmeticException.class)
public void test() {program, ami kivételt dob}
```

4. Sorolja fel, hogy a Scrum módszertanban kik tartoznak a csirkék (chickens, ancillary roles) közé ! (3 pont)

Stakeholders (customers, vendors)
Managers (including Project Managers)

Mi történik refaktoráláskor (refactoring) az XP agilis módszertanban ? (2 pont)

A szoftvert úgy fejlesztjük tovább, hogy a külső viselkedés változatlan marad, de a belső szerkezet megújul.

5. Milyen tesztelési (integrációs) stratégia esetében alkalmaznak teszt betétet (test stub) ? (2 pont)

top-down

Mi a funkciója a teszt betétnek ? (2 pont)

a még hiányzó alárendelt szoftver komponenst helyettesíti (pl. under construction weblap)

Egy szoftver termék felülvizsgálata (review) milyen következményekkel (follow-up) zárulhat ? (3 pont)

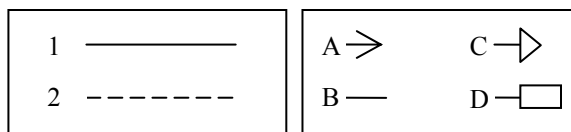
Unqualified acceptance of review item

Conditional acceptance of the review item with minor action
(confirmation)

Conditional acceptance of the review item with major action (new review)

6. Az alábbi Java kódrészlet alapján töltsse ki a táblázatot. A rubrikákba azt kell beírni, hogy a két osztály közti milyen jelölést kellene rajzolni UML 2 osztálydiagram esetén. Minden rubrikába egy számot és egy betűt kell írni a kulcs alapján: a szám jelzi a vonal stílusát, a betű a *fejlécben* levő osztály oldalára kerülő végződést. (10 pont)

```
class A {
    String name;
    public E e;
}
class B extends E {
    public void foo(C c) {}
    public D foo() { return new D(); }
}
class C {
    private Map<String, A> list;
}
class D {
    public void bar(A a) { a.e.qux(); }
}
class E {
    public void qux() {}
}
```



	A	B	C	D	E
A	-	-	1D	2B	1A
B	-	-	2A	2A	1C
C	1A	2B	-	-	-
D	2A	2B	-	-	2A
E	1B	1B	-	2B	-

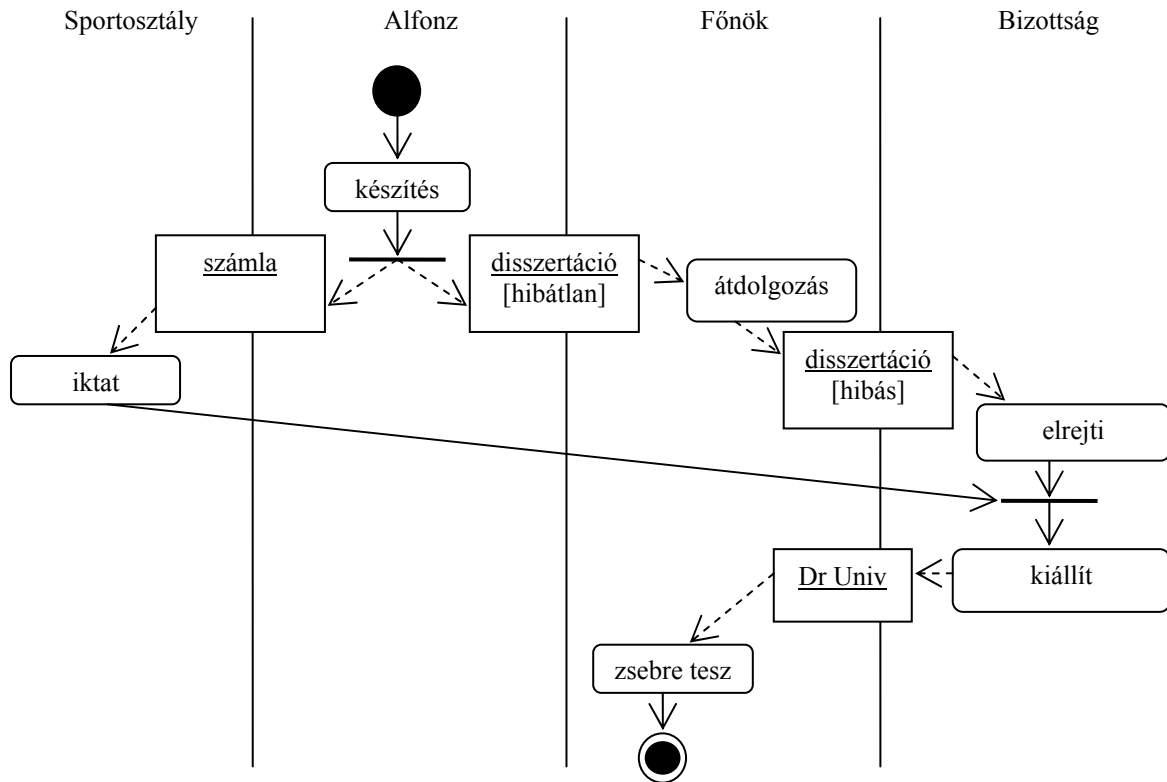
7. Egy programban két Java szálunk van, *t1* és *t2*. Mindkettő ugyanazt a Qux objektumot ismeri (*q*). A *t1* meghívja a *q.foo()*, majd 1 mp múlva *t2* a *q.bar()* metódusát. A metódusok lefutása után a szálak megállnak (végetérnek). Milyen sorrendben zajlnak le a számozott sorok, és az egyes sorok hatására milyen állapotba kerülnek a szálak? (7 pont)

```
public class Qux {
    synchronized void foo() throws InterruptedException { //1
        Thread.sleep(10000); // 10 mp-ig alszik //2
        wait(); //3
    } //4
    synchronized void bar() { //5
        notifyAll(); //6
    } //7
}
```

Esemény (sor)	t1	t2
0	Runnable	Runnable
1	- '' -	- '' -
2	Timed waiting (sleep)	- '' -
5	- '' -	Blocked
3	Waiting	Runnable
6	Blocked	- '' -
7	Runnable	Terminated
4	Terminated	- '' -

8. Készítsen UML 2 aktivitás-diagramot (activity diagram) az alábbi leírás alapján! Jelölje az action-object flow-t is! Használja a kövéren szedett kifejezéseket! (9 pont)

Senki **Alfonz** opponens **disszertációt készít**. A hibátlan disszertációt elküldi a **főnökének átdolgozásra**, és ezzel párhuzamosan **számlát ad a sportosztálynak**. A főnök az átdolgozás során néhány helyesírási hibát tesz a disszertációba, és átadja a **bizottságnak, ahol elrejtik**. A sportosztály **iktatja** a számlát, és ezt jelzi a bizottság felé. A bizottság, miután végzett a rejtéssel és megkapta a sportosztály jelzését, **kiállítja** a doktori **bizonyítványt**, amit elküld a főnöknek, aki a címet legjobb tudása szerint **zsebre teszi**. A folyamatnak itt vége szakad.



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból

2012. december 18.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. Adott az alábbi (hibás) Java kódrészlet.

```
public class Valami extends Exception implements Serializable {
    private transient int foo;
    public boolean bar;
    protected static String baz;
    public Valami(String s) { baz = s; }
}

public class Main {
    public void serialize(String file, Set<Valami> sv) {
        try {
            FileOutputStream os = new FileOutputStream(file);
            ObjectOutputStream oos = new ObjectOutputStream(os);
            oos.writeObject(sv);
            os.close();
        } catch (Exception e) { e.printStackTrace(); }
    }
}
```

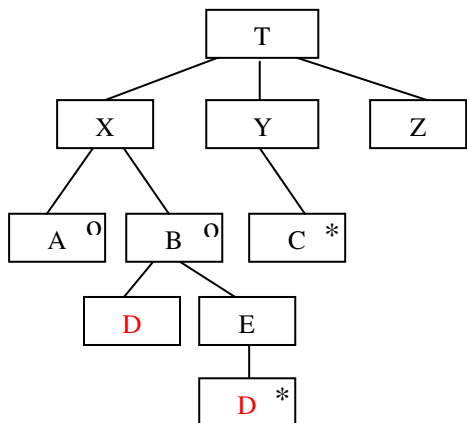
Jellemezze az alábbi állításokat a kulcs felhasználásával! (8 pont)

- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

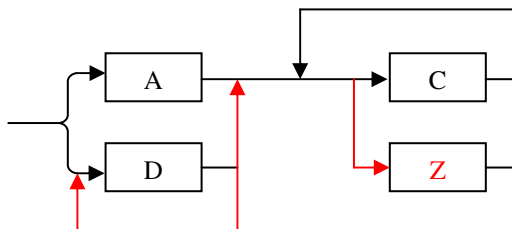
- [B]** A **Main** osztály nem tud sorosítani, mert a **Main** osztály nem implementálja a **Serializable** interfészt.
- [E]** A **serialize** metódus **sv** paramétere nem sorosítható, mert interfész típusú változót nem lehet sorosítani.
- [E]** A **Valami** konstruktorában a **baz** attribútumnak egyenlőségjellel nem adható értékül az **s** paraméter, mert csak a következő forma lenne helyes: **baz = new String(s)**
- [A]** Egy **Valami** típusú objektum **foo** attribútuma nem kerül sorosításra, mert **private** láthatóságú attribútum nem sorosítható.
- [D]** A **Main** osztály **serialize** metódusa tartalmaz hibát, mert hiányzik belőle egy **oos.close()** hívás.
- [B]** A **Valami** osztály egy példánya nem sorosítható a **writeObject** metódussal, mert **Valami** az **Exception** leszármazottja.
- [B]** **protected** módosítóval ellátott attribútum nem sorosítható, ezért a **Valami** osztályban definiált **baz** attribútum nem kerül sorosításra.
- [B]** A **FileOutputStream** konstruktorhívása hibás, mert nem **File** típusú paramétert kap.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total ≥ 0

2. Egészítse ki az alábbi ELH-t és állapottáblát úgy, hogy mindkettő ugyanazt a szerkezetet írja le! A kiegészítés során az **ábrák szerkezetén**, az azokba **beírt, berajzolt elemeken változtani tilos!** Az ELH-ban csak a két üresen maradt blokkot kell kitölteni! Az állapottáblát egészítse ki! Az induló állapot legyen az ① ! (ELH 2 pont, állapottábla 4 pont)



	A	C	D	Z
①	③		②	
②		③	②	④
③		③		④
④				



A szintaxis gráfot egészítse ki úgy, hogy az is a fenti szerkezet írja le! (3 pont)

3. Az alábbiak közül mely deklarációk szerepelhetnek a 2. példában adott szerkezetet leíró DTD-ben? (2 pont)

- | | | | | | |
|--------------------------|----------|------------------|--------------------------|----------|---------------|
| <input type="checkbox"/> | T | (X+Y+Z) | <input type="checkbox"/> | Y | (C+) |
| <input type="checkbox"/> | T | ((A D*), C*, Z) | <input type="checkbox"/> | D | (E+) |
| <input type="checkbox"/> | X | (A+B) | <input type="checkbox"/> | Y | (C*) |
| <input type="checkbox"/> | X | (A B) | <input type="checkbox"/> | B | (D, E*) |
| <input type="checkbox"/> | T | (X, Y, Z) | <input type="checkbox"/> | B | (D, E) |
| <input type="checkbox"/> | T | ((A B), C+, Z) | <input type="checkbox"/> | B | (D+) |
| <input type="checkbox"/> | T | ((A B), C*, Z) | <input type="checkbox"/> | C | (Y*) |
| <input type="checkbox"/> | X | (A (D, E D*)) | | | |

4. Legyen az alábbi A osztályunk.

```
class A {
    protected int j;
    public int foo(int i) {return(j);}
}
```

Legyen egy B osztály, amely A-ból származik, és metódusai az alábbi táblázatban találhatóak. Jelölje meg az(oka)t a metódus(oka)t, amely(ek) megsérti(k) a Meyer-féle nyit-zár (open-close) elvet! (2 pont)

- public int foo1(int i) {return(j);}
- public double foo(double d, int i) {return(d*j);}
- public int foo(int i) {return(j*i);}**
- public int foo(double d) {return((int)(j*d));}
- public double foo2(int i) {return(3.0*i);}

5. Sorolja fel a Scrum agilis módszertan legfontosabb dokumentumait, anyagait (artifacts) (3 pont)

Product backlog (termék teendőlista) **Sprint backlog (futam teendőlista)**

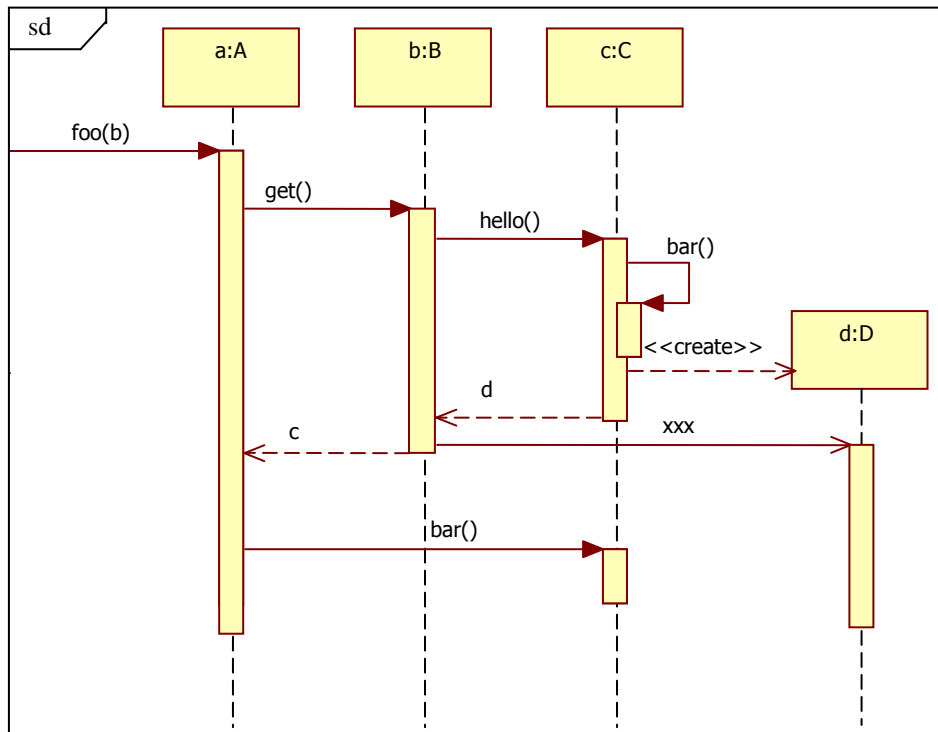
Burndown chart (eredménykimutatás).....

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

6. Az alábbi Java kódrészletek alapján készítsen UML 2 szekvenciadiagramot! A diagramot a *START*-tal jelölt megjegyzéstől kezdje! Tételezze fel, hogy a D.xxx() metódus aszinkron! (10 pont)

```
public interface X {
    void bar();
}
public class A {
    static protected double qux(double d){
        return 2*d;
    }
    public void foo(B b) {
        C c = b.get();
        c.bar();
    }
}
public class B {
    C c;
    public void set(C x) {
        c = x;
        c.bar();
    }
    public C get() {
        c.hello().xxx();
        return c;
    }
}
}
```

```
public class C {
    public void bar() {
    }
    public X hello() {
        bar();
        return new D();
    }
}
public class D implements X {
    public void xxx() {}
}
public class Main {
    public static void main(String args[]) {
        A a = new A();
        B b = new B();
        C c = new C();
        b.set(c);
        // START
        a.foo(b);
        // END
    }
}
```



7. Jellemezzünk egy stringet az alábbi műveletekkel!

CRT() új (üres) stringet hoz létre
LGTH(s) az s string karaktereinek számát adja
TAIL(s) az s string első karakterének levágása után maradó stringet adja
XTEND(s,x) az s string végére rakja az x karaktert
TOP(s) az s string első karakterét mutatja meg
PALIN(s) igaz, ha az s string palindróma

Egy string palindróma, ha az elejétől olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

Az alábbi táblázatban található kifejezések közül jelölje meg azokat, amelyek algebrai axiómák BAL oldalán állhatnak! (6 pont)

- | | |
|--|--|
| <input type="checkbox"/> TOP(TAIL(s)) | <input type="checkbox"/> PALIN(TAIL(CRT())) |
| <input type="checkbox"/> LGTH(TOP(CRT())) | <input type="checkbox"/> LGTH(TAIL(s)) |
| <input type="checkbox"/> XTEND(TOP(s)) | <input checked="" type="checkbox"/> PALIN(XTEND(s, x)) |
| <input type="checkbox"/> TAIL(XTEND(CRT())) | <input type="checkbox"/> LGTH(TOP(s)) |
| <input checked="" type="checkbox"/> PALIN(CRT()) | <input checked="" type="checkbox"/> TOP(XTEND(s, x)) |

8. A felülvizsgálat (review, walkthrough stb.) során minden egyes „akció elem”-hez (hiba, ellentmondás, javaslat stb.) a következőket kell előírni (6 pont) :

- the responsible person
- the action to be taken
- the problem severity and type of bug

9. A RUP (Rational Unified Process) egyik munkafolyamatában (workflow) szerződés (contract) készítését javasolja illetve írja elő. Melyik munkafolyamatban esedékes szerződés készítése ? Kik között kell szerződést készíteni ? Milyen fontosabb pontjai vannak a szerződésnek ? (4 pont)

Munkafolyamat (workflow): **analízis**.....

Szerződő felek: **operációk és az operációk felhasználói**.....

Szerződés fontosabb pontjai: **Responsibilities, Pre-conditions, Post-condition, Types, Crossrefs, Exceptions, Output**

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2013. január 8.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. Jelölje, hogy az oszlopokban szereplő Java nyelvi elemek milyen módosítókkal rendelkezhetnek! (7 pont)

	konstruktor	statikus metódus	példány metódus	példány attribútum
abstract	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
final	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
protected	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
static	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
synchronized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
transient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
volatile	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Mi a Rational Unified Process (RUP) életciklus modelljének utolsó fázisa? Milyen tevékenységek tartoznak ebbe a fázisba? (4 pont)

utolsó fázis: **.transition**

tevékenységek: **manufacturing, delivering, training**

3. Jellemezzünk egy stringet az alábbi műveletekkel!

- ADD(s, x)** az s string végére rakja az x karaktert.
- LGTH(s)** az s string karaktereinek számát adja.
- NEW()** új (üres) stringet hoz létre.
- OCN(s, x)** az s stringben található x karakterek előfordulásának számát adja.
- TAIL(s)** az s string legrégebbi karakterének levágása után maradó stringet adja.

Az alábbi kifejezésekhez adja meg, hogy azok algebrai axiómák BAL oldalán állhatnak-e vagy sem! (5 pont)

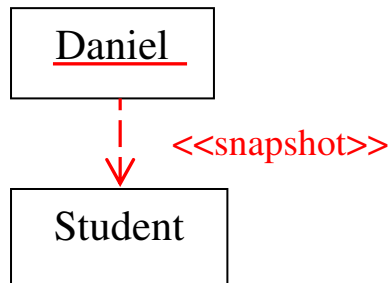
Igen	Nem		Igen	Nem	
<input type="checkbox"/>	<input type="checkbox"/>	OCN(TAIL(s), x)	<input type="checkbox"/>	<input type="checkbox"/>	ADD(NEW(), x)
<input type="checkbox"/>	<input type="checkbox"/>	LGTH(OCN(s, x))	<input type="checkbox"/>	<input type="checkbox"/>	OCN(ADD(s, y), x)
<input type="checkbox"/>	<input type="checkbox"/>	OCN(ADD(s, x), x)	<input type="checkbox"/>	<input type="checkbox"/>	OCN(NEW(), 0)
<input type="checkbox"/>	<input type="checkbox"/>	TAIL(OCN(s, x))	<input type="checkbox"/>	<input type="checkbox"/>	LGTH(TAIL(s))
<input type="checkbox"/>	<input type="checkbox"/>	LGTH(ADD(NEW(), x))	<input type="checkbox"/>	<input type="checkbox"/>	ADD(s, LGTH(s))

4. Egy UML2 modelben legyen egy Student osztályunk. Daniel a Student osztály valós idejű példányának UML2-beli modellje.

Kinek a példánya Daniel? (2 pont): **.Instance specification**

Kinek a példánya Student? (1 pont): **Class**.....

Az alábbi (nem korrekt!) részletet kiegészítve javítsa az ábrát és jelölje be a Student és Daniel közötti kapcsolatot! (2 pont)



5. Az alábbi táblázatban a fontosabb szoftver architektúra típusokat adtuk meg, minden típushoz egy indexet rendelve. A jobboldali táblázatban szoftver architektúrákkal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik architektúra típushoz kapcsolhatóak leginkább! (A fogalom előtti mezőbe írja be a típus indexét!) Ha a fogalom egyik típushoz sem társítható egyértelműen, akkor a mezőbe tegyen X-et! (3 pont)

1	esemény alapú implicit hívást használó (event-based implicit invocation)
2	csővezeték+szűrők (pipes and filters)
3	tábla (blackboard)
4	interpreter
5	szolgáltatás orientált architektúra (SOA)

5	XML
1	szórás (broadcast)
2	batch (kötegelés)
2	adatfolyam (data flow)
1	előfizet (subscribe)
X	grafikus interfész (GUI)

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

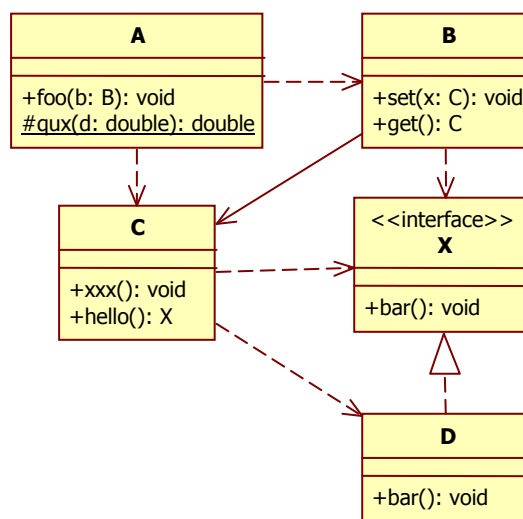
6. Készítsen Java metódust, amelyik két, X típust tartalmazó halmazt kap paraméterül, és visszaadja a két halmaz uniójának elemszámát. Az eredeti halmazok nem változhatnak meg, az elemekhez egyedileg nem szabad hozzáférni! Ahol lehet, használja a collection framework osztályait és interfészeit! Nem használhat default konstruktort! Törekedjen minél általánosabb megoldásra! (4 pont)

```
int unionSize(Set<X> s1, Set<X> s2) {
    Set<X> r = new HashSet<X>(s1);
    r.addAll(s2);
    return r.size();
}
```

7. A Java kódrészletek alapján rajzoljon UML 2 osztálydiagramot! (8 pont)

```
public interface X {
    void bar();
}
public class B {
    C c;
    public void set(C x) {
        c = x;
        c.xxx();
    }
    public C get() {
        c.hello().bar();
        return c;
    }
}
public class D implements X {
    public void bar() {}
}
```

```
public class C {
    public void xxx() {
    }
    public X hello() {
        xxx();
        return new D();
    }
}
public class A {
    static protected double qux(double d){
        return 2*d;
    }
    public void foo(B b) {
        C c = b.get();
        c.xxx();
    }
}
```



8. A szoftver fejlesztés Ward-Mellor modelljében mi áll a (2 pont)

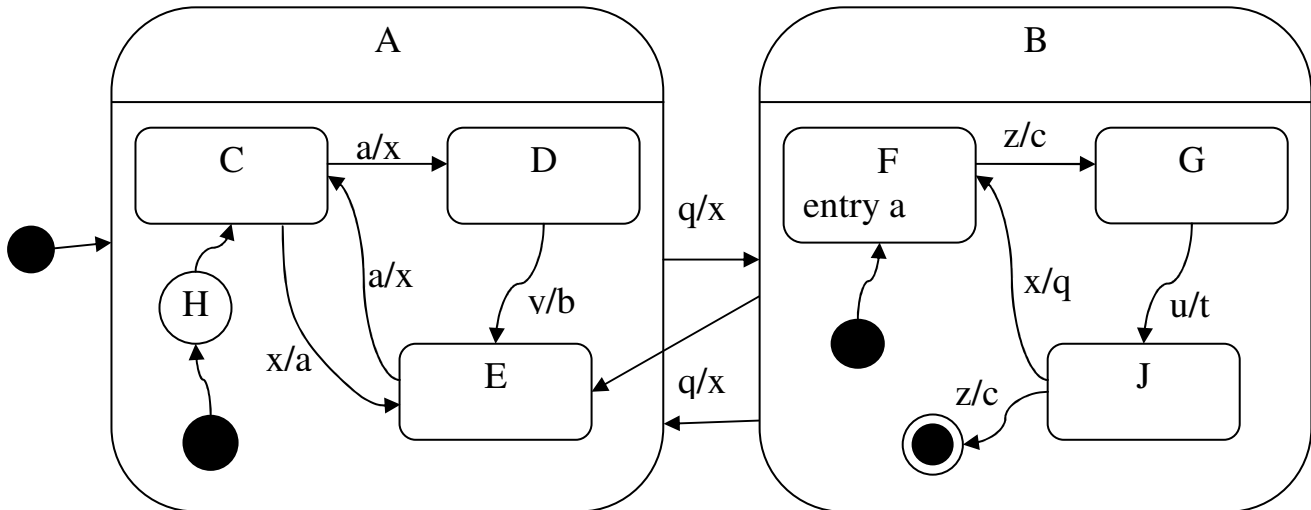
vízszintes tengelyen: **implementációs függőség**

függőleges tengelyen: **absztrakció**

Egy mondatban fogalmazza meg a vízszintes tengelyen szereplő fogalom jelentését! (2 pont)

A leíráshoz használt fogalom mennyire "szoftveres"

9. A következő UML2 állapotdiagram alapján minősítse az állításokat! (7 pont)

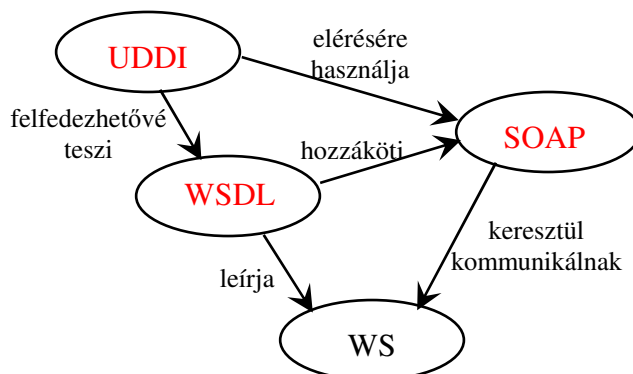


Igaz	Hamis	Állítás
<input type="checkbox"/>	<input type="checkbox"/>	D állapotból 2 lépésben visszaérhet D-be
<input type="checkbox"/>	<input type="checkbox"/>	J állapotból egyetlen esemény hatására D állapot következhet
<input type="checkbox"/>	<input type="checkbox"/>	F állapotból „q” esemény hatására H állapotba kerül
<input type="checkbox"/>	<input type="checkbox"/>	B-ből A-ba való váltáskor végrehajtható az „a” tevékenység

A kezdés után az **x, q, z, q** esemény-szekvencia hatására

Igaz	Hamis	Állítás
<input type="checkbox"/>	<input type="checkbox"/>	Kétszer lefut az „x” tevékenység
<input type="checkbox"/>	<input type="checkbox"/>	E állapotba kerülünk
<input type="checkbox"/>	<input type="checkbox"/>	Kétszer lefut az „a” tevékenység

10. A szolgáltatás orientált architektúrát bemutató ábrán mik a név nélküli elemek? (3 pont)



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2013. január 15.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám >= 0

1. Subversion-ben a commit (check in) végrehajtása attól függ, hogy az utolsó letöltés (check out/update) óta módosult-e a letöltött és/vagy a repositoryban tárolt változat.

Az alábbi táblázatba írja be, hogy a változásoktól függően mi történik commit esetén (4 pont)

		repository példány	
		nem változott	változott
munkap éldány	változott	publish	out of date error
	változatlan	-	-

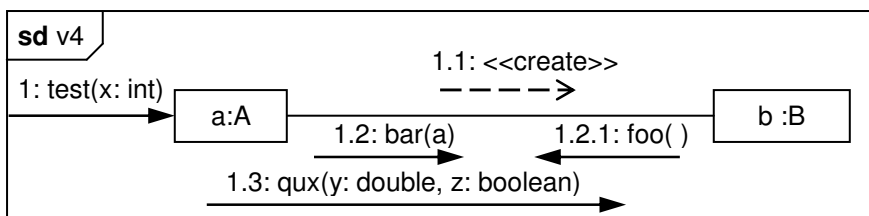
2. Legyen egy *X* osztályunk, *aaa()* és *bbb()* metódusokkal jellemezve. Egy kliens meghívja az *aaa()* metódust. Az *aaa()* futása közben egy másik kliens meghívja a *bbb()* metódust. Az alábbi táblázatba írja be, hogy a különböző UML2 szemantikák esetében mi a követett eljárás (policy)! (5 pont)

szemantika neve	eljárás (policy)
sorrendi (Sequ)	kizárt – nem fordulhat elő
őrzött (Guarded)	befejezi aaa-t, majd elkezd bbb-t
konkurrens (Conc)	félbeszakítja aaa-t és elkezd bbb-t

Miben különbözik, ha a másik kliens is az *aaa()* metódust hívja? (1 pont)

semmiben

3. Adja meg, hogy az alábbi UML2 kommunikációs diagramon leírt viselkedés megvalósításához minimálisan milyen példány attribútumokat szükséges definiálni az osztályokban! (3 pont)



osztály	nem kell attribútum	attribútum típusa
A	□	
B	□	

4. Milyen általános kiterjesztő technikákat (general extension mechanisms) alkalmaz az UML2? (3 pont)

constraint, stereotype, tagged value

5. Adott az alábbi (hibás) Java kódrészlet.

<pre>public class ThreadSafe { private long x; private long y; private Object o = new Object(); public void setX(long v) { x = v; y = 1-x; } public long getXY() { return x+y; } public synchronized void setY(long w) { o.notify(); y = w; x = 1-y; } public void finalize() { System.out.println("destructor called"); } }</pre>	<pre>public class MyThread implements Runnable { private boolean done = false; private ThreadSafe ts; public MyThread(ThreadSafe ts) { this.ts = ts; } public void exit() { done = true; } public void run() { long i = 0; while (!done) { ts.setX(i++); } ts.finalize(); ts.getXY(); } } public class Main { public static void main(String[] args) { MyThread mt = new MyThread (new ThreadSafe()); mt.start(); } }</pre>
---	---

Jellemezze az alábbi állításokat a kulcs felhasználásával! (8 pont)

- A** - csak az első tagmondat igaz (+ -)
- B** - csak a második tagmondat igaz (- +)
- C** - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
- D** - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
- E** - egyik tagmondat sem igaz (- -)

- [B]** Egy **ThreadSafe** objektum belső állapota mindig konzisztens marad a **setX** metódus több szálból egyszerre történő hívása során is, mert a metódus törzse nem tartalmaz feltételes elágazást.
- [C]** A **ThreadSafe** típusú objektumok használata esetén a **getXY** metódus nem mindig 1-gyel tér vissza, mert egyszerre több szál is be tud lépni a metódusba.
- [E]** Az **o.notify()**-t **synchronized** metódusból kell hívni, ezért az **o.notify()** hívás a **setY** metódusban nem dob **IllegalMonitorStateException** kivételt.
- [A]** A **main** metódusban az **mt.start()** hívás hibás, mert a szálat az **mt.run()** hívással kell elindítani.
- [A]** A **MyThread** szál megállítására használt **done** attribútum hiányosan van definiálva, mert a **transient** kulcsszóval jelezni kell, hogy az attribútum értékét más szál is módosíthatja.
- [E]** A **MyThread.run** metódusában a **ts.finalize()** hívás meghívja a garbage collectort és felszabadítja a **ts** objektumot, ezért az ezt követő **ts.getXY()** hívás **NullPointerException** kivételt fog dobni.
- [B]** A **ThreadSafe** típusú objektumok többszálú használata esetén az **x+y==1** invariáns mindig érvényes marad, mert egy **MyThread** szál csak egyetlen **ThreadSafe** objektumot használ.
- [E]** A **MyThread** konstruktorában a **ts** paraméter neve nem egyezhet meg a **ts** attribútum nevével, mert így a **this.ts = ts** utasításnak semmi hatása nincs.

6. Felsoroltunk szoftverrel kapcsolatos "hibákat". A hibák mellett jelölje be, hogy az melyik kategóriába tartozik! (4 pont)

	failure	error	bug	fault
A ciklusfeltétel hibás	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Alábecsültük a példányosítás erőforrás-igényét	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hiányzik a "synchronized" kulcsszó	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sok objektumnál nagyon lassan kapjuk az eredményt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Elmaradt a kritikus kódok felülvizsgálata	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Az első futásnál hibás eredményt kapunk	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

7. Adott az alábbi dekorált XML leírás.

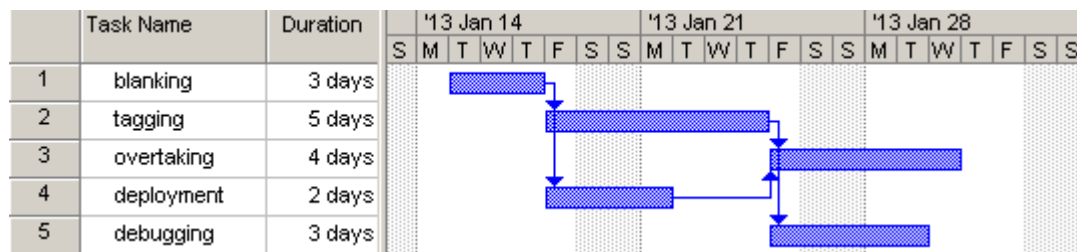
```
<!DOCTYPE y [
  <!ELEMENT y (a+, b)>
  <!ELEMENT a ANY>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA |y)*>
]>
<y>
  <a><y><a>
    ②
    <b></b></y>
  a/</a><b>/b</b>
</y>
```

Mi állhat ② helyében, hogy az XML érvényes legyen? (6 pont)

- <c>/b</c>b
- <a><c>b</c>b
- b/bb
- <c>b</c></y>
- <a>a/b
- <y><c>b</c></y>
- <c>b</c>
- <y><c>b</c></y>

8. Egy ma (január 15., kedd) induló projekt feladatait, azok hosszát és függéseit a mellékelt táblázat foglalja össze. Rajzoljon Gantt-diagramot ! (szombaton és vasárnap nem dolgozunk !) (3 pont)

szám	feladat	nap	függés
1	blinking	3	
2	tagging	5	1
3	overtaking	4	2, 4
4	deployment	2	1
5	debugging	3	2

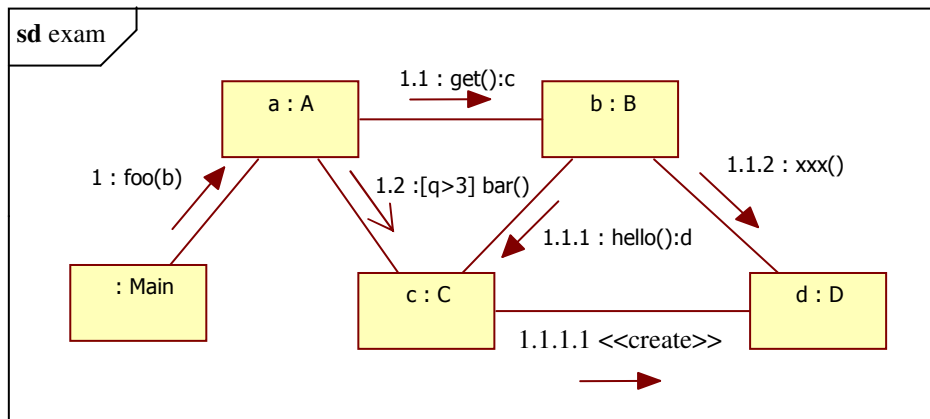


Hányadikán kezdhethetjük legkésőbb a „deployment”-et, ha a véghatáridőt tartani akarjuk (2 pont)? **jan 23.-án**

9. Az alábbi Java kódrészletek alapján készítsen UML 2 kommunikációs diagramot! A diagramot a START-tal jelölt megjegyzéstől kezdje! Tételezze fel, hogy a C.bar() metódus aszinkron! (7 pont)

```
public class A {
    double q;
    static protected double qux(double d){
        q = d/2;
        return 2*d;
    }
    public void foo(B b) {
        C c = b.get();
        if (q > 3) c.bar();
    }
}
public class B {
    C c;
    public void set(C x) {
        c = x;
        c.bar();
    }
    public C get() {
        c.hello().xxx();
        return c;
    }
}
}
```

```
public class C {
    public void bar() {
    }
    public X hello() {
        return new D();
    }
}
public class D implements X {
    public void xxx() {}
}
public class Main {
    public static void main(String args[]) {
        A a = new A();
        B b = new B();
        C c = new C();
        b.set(c);
        // START
        a.foo(b);
    }
}
public interface X {
    void xxx();
}
}
```



10. Az UML2-ben definiált **Sequence** gyűjteménynek (kollekciónak) adja meg a tulajdonságait! (4 pont)

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

igen	nem	nem jellemző	tulajdonság
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	egyedi (unique)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	minősített (qualified)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	rendezett (ordered)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	delegált (delegated)

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból
2013. május 28.

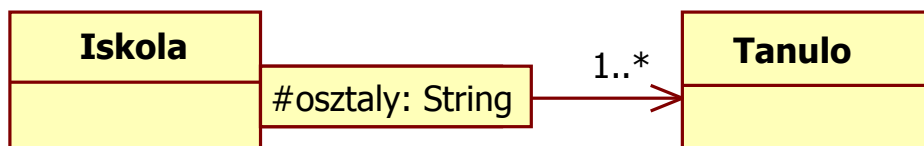
Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. Jelölje az alábbi, Java nyelvre vonatkozó állítások igazságtartalmát! (6 pont)

igen	nem	
<input type="checkbox"/>	<input type="checkbox"/>	konstruktorok nem lehet láthatósága
<input type="checkbox"/>	<input type="checkbox"/>	private tag nem szerializálódik
<input checked="" type="checkbox"/>	<input type="checkbox"/>	statikus tag nem szerializálódik
<input checked="" type="checkbox"/>	<input type="checkbox"/>	lehet olyan private tag, aminek többször is lehet értéket adni
<input type="checkbox"/>	<input type="checkbox"/>	privát metódust csak privát metódusból lehet hívni
<input type="checkbox"/>	<input type="checkbox"/>	statikus metódusban használható a this változó
<input checked="" type="checkbox"/>	<input type="checkbox"/>	final metódusban használható a this változó
<input type="checkbox"/>	<input type="checkbox"/>	statikus metódus nem lehet private
<input type="checkbox"/>	<input type="checkbox"/>	final metódus nem lehet statikus
<input checked="" type="checkbox"/>	<input type="checkbox"/>	final metódus nem lehet abstract
<input type="checkbox"/>	<input type="checkbox"/>	absztrakt osztálynak nem lehet final metódusa
<input type="checkbox"/>	<input type="checkbox"/>	két interfész csak akkor valósítható meg egy osztályban, ha az interfészeknek nincsen közös metódusa

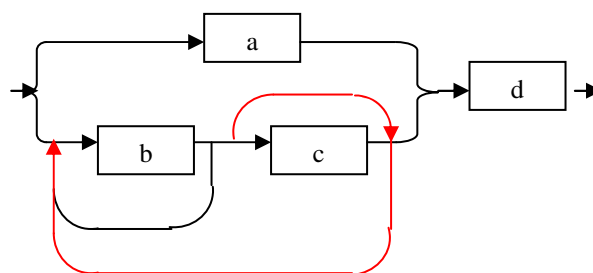
2. Az iskola és a tanulók viszonyát az alábbi UML ábrával írjuk le. Ez alapján definiálja az Iskola osztályt Java nyelven, a kollekció keretrendszer elemeinek felhasználásával! (4 pont)



```
import java.util.*;
public class Iskola {
    protected Map<String, List<Tanulo>> x;
}
```


3. A szintaxis gráfot egészítse ki úgy, hogy az a DTD-vel definiált adatszerkezetet írja le! (4 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x ((a | (b+, c?)+), d)>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
  <!ELEMENT d (#PCDATA)>
]>
```



4. Egy objektum metódusa **szekvenciálisan** kohézív, ha a metódus ... (3 pont)

- adatszekvenciával (tömb, lista) tér vissza
- kódjában csak utasítás-szekvenciák vannak (pl. nincs benne iteráció)
- paramétere egy vagy több adatszekvencia (tömb, lista)
- több funkciót hajt végre, ezek mind az inicializáláshoz kapcsolódnak
- több, egymást követő részfunkciót lát el**
- nevében benne van a szekvencia (sequence) (pl. read_input_sequence)

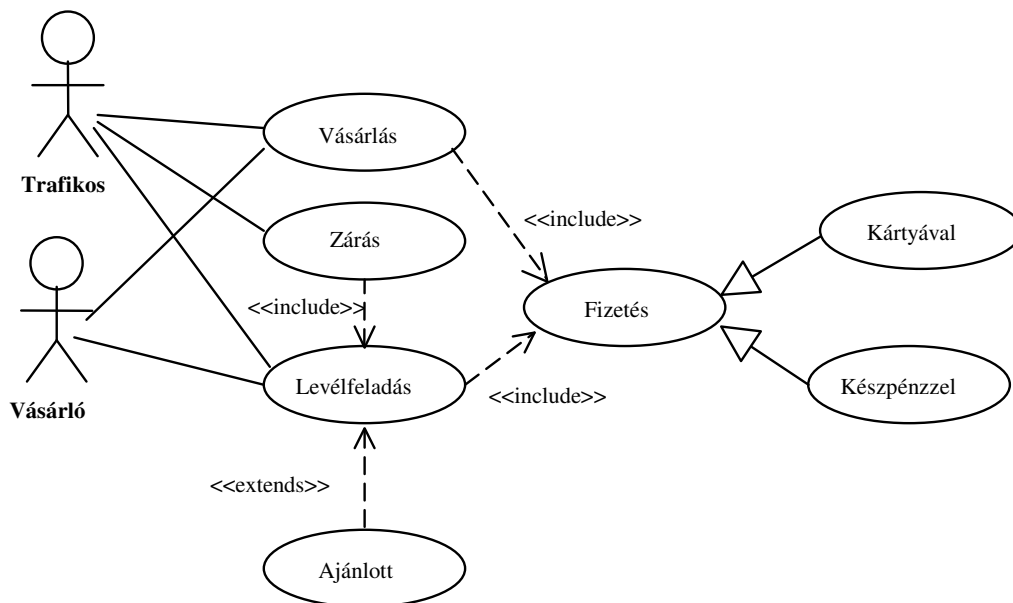
Egy objektum metódusa **procedurálisan** kohézív, ha a metódus ... (3 pont)

- egy nagyon fontos procedúrát hajt végre
- kódjában további metódus és procedúra hívások vannak
- különböző adattípusra (osztályra) hajtja végre ugyanazt a műveletet**
- nevében legalább két ige (pl. read_and_validate_input) áll
- nem tartalmaz adat-deklarációkat, csak kódot
- egy adattípusra (osztályra) különböző műveleteket (procedúrákat) hajt végre

5. Jelölje az állítások mellett 1-5-ig, hogy minimálisan melyik CMM szinttől igazak ! Ha az állítás nem értelmezhető, akkor tegyen X-et ! (4 pont)

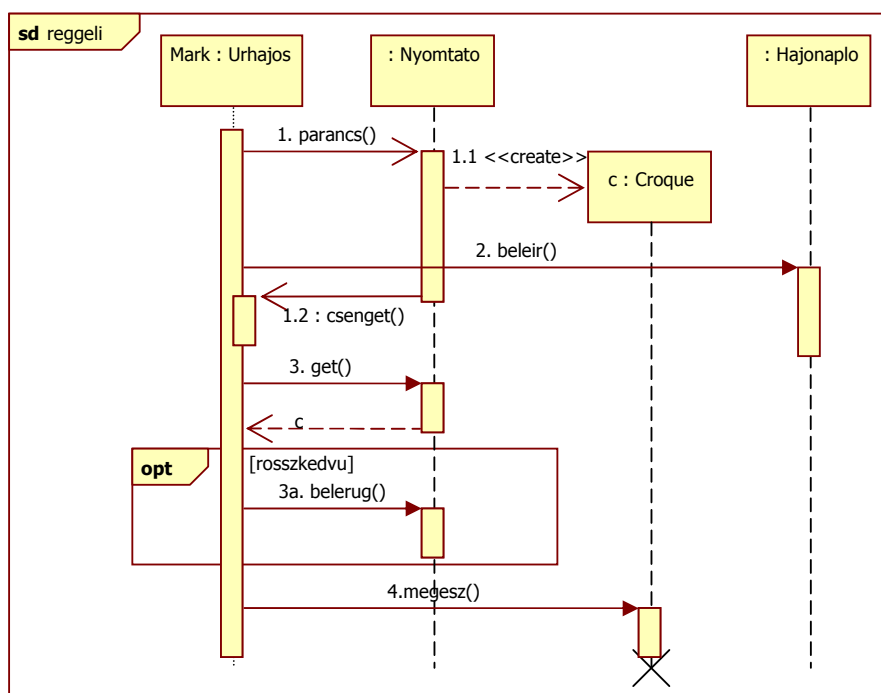
3	Minden projekt a szervezet szabványos fejlesztési és karbantartási folyamatának egy jóváhagyott, személyre szabott verzióját követi.
X	Minden termék szabvány szerinti (pl. ISO 9126) megfelelőségét ellenőrzik.
X	A folyamat számszerű mérése lehetővé teszi az ipari szabványok továbbfejlesztését.
2	Fejlett projekt-menedzsment technikákat és eszközöket alkalmaznak a napi gyakorlatban

7. A nemzeti dohányárudában a trafikos közreműködésével lehet – cigarettát, alkoholt stb. – vásárolni és levelet föladni. Készpénzzel vagy SZÉP-kártyával fizethet a vásárló. Egyes trafikokban lehetőség van ajánlott leveleket is föladni. Záráskor a napi forgalmi adatokat a trafikos levélben feladja a nemzeti adóhivatal részére. Rajzoljon UML2 use-case diagramot ! (8 pont)



8. Rajzoljon UML2 szekvenciadiagramot az alábbi leírás alapján! (8 pont)

Mark Watney, a marson ragadt űrhajós reggeli rutinja a következő. Az ételnyomtatónak parancsot ad, hogy készítsen egy Croque Monsieur-t. Amíg az étel készül, Mark beleír a hajónaplóba. Írás közben a nyomtató csenget (amikor befejezte az ételgyártást). Mark, mikor végzett az írással, kiveszi az ételt és az utolsó morzsáig megeszi. Ha rossz passzban van, akkor az étel kivétele után (de mielőtt megenné) belerúg a nyomtatóba.



9. Rajzoljon UML 2 időzítési diagramot az alábbi Java kódrészlet alapján az **x**, **y** és **z** objektumokra! (8 pont)

```
enum XS { A, B, C }
enum YS { M, N }
enum ZS { P, Q }
```

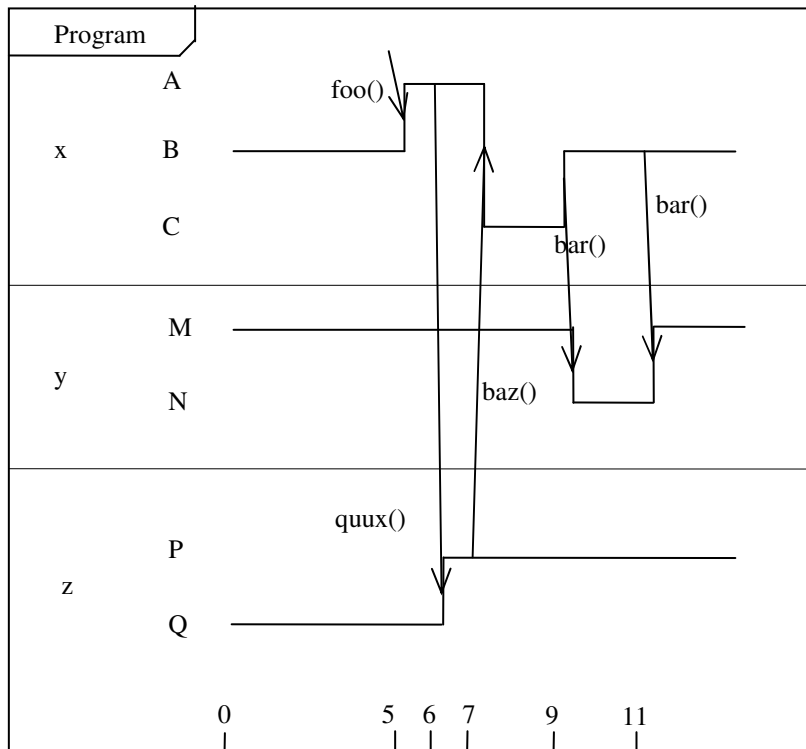
```
class Base {
    static void sleep(int seconds) {
        try {
            Thread.sleep(seconds*1000);
        } catch (InterruptedException e) { }
    }
}
```

```
class X extends Base
{
    XS xs = XS.B;
    void foo() {
        xs = XS.A; sleep(1); Program.z.quux();
    }
    void baz() {
        xs = XS.C; sleep(2); Program.y.bar();
        xs = XS.B; sleep(2); Program.y.bar();
    }
}
```

```
class Y extends Base
{
    YS ys = YS.M;
    void bar() {
        ys = ys == YS.M ? YS.N : YS.M;
    }
}
```

```
class Z extends Base
{
    ZS zs = ZS.Q;
    void quux() {
        zs = zs == ZS.P ? ZS.Q : ZS.P;
        sleep(1); Program.x.baz();
    }
}
```

```
public class Program extends Base {
    static X x = new X();
    static Y y = new Y();
    static Z z = new Z();
    public static void main(String[] args) {
        sleep(5);
        x.foo();
    }
}
```



10. Az üzleti folyamatok lefutásának egyik esete az orkesztráció (orchestration). Mi a lényege ? (2 pont)

egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2013. június 11.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

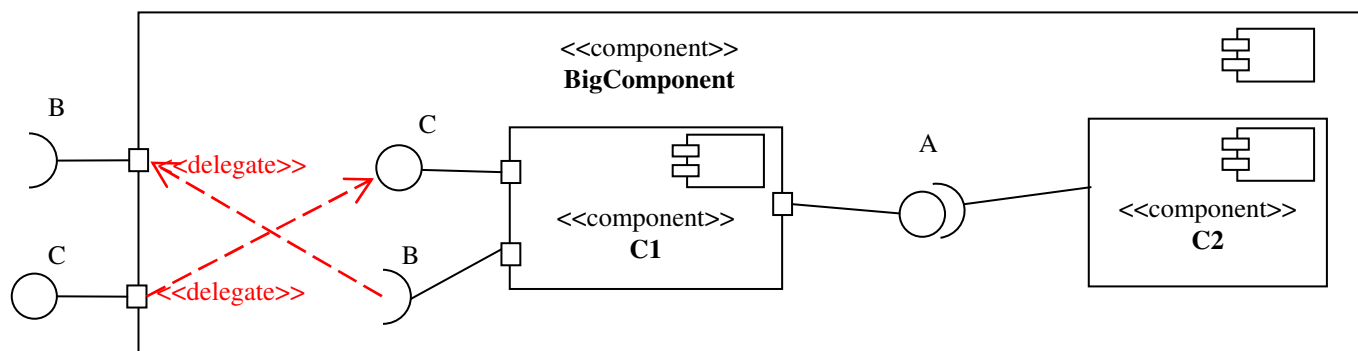
1. Készítse el azt az állapottáblát, amely megfelel az alábbi DTD-vel definiált adatszerkezetnek! A tábla az 1. állapotban kezdődik, és segítségül megadtunk két blokkot. Kötőjellel (-) jelölje, ha egy adatelem, egy állapotban nem fogadható el! (5 pont)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE x [
  <!ELEMENT x      ((a,b+) | (c,b?)) *>
  <!ELEMENT a      (#PCDATA)>
  <!ELEMENT b      (#PCDATA)>
  <!ELEMENT c      (#PCDATA)>
]>
```

	a	b	c
1	2	-	3
2	-	4	-
3	2	1	3
4	2	4	3

2. Rajzolja be alábbi UML2 diagramba a hiányzó kapcsolatokat! (3 pont)



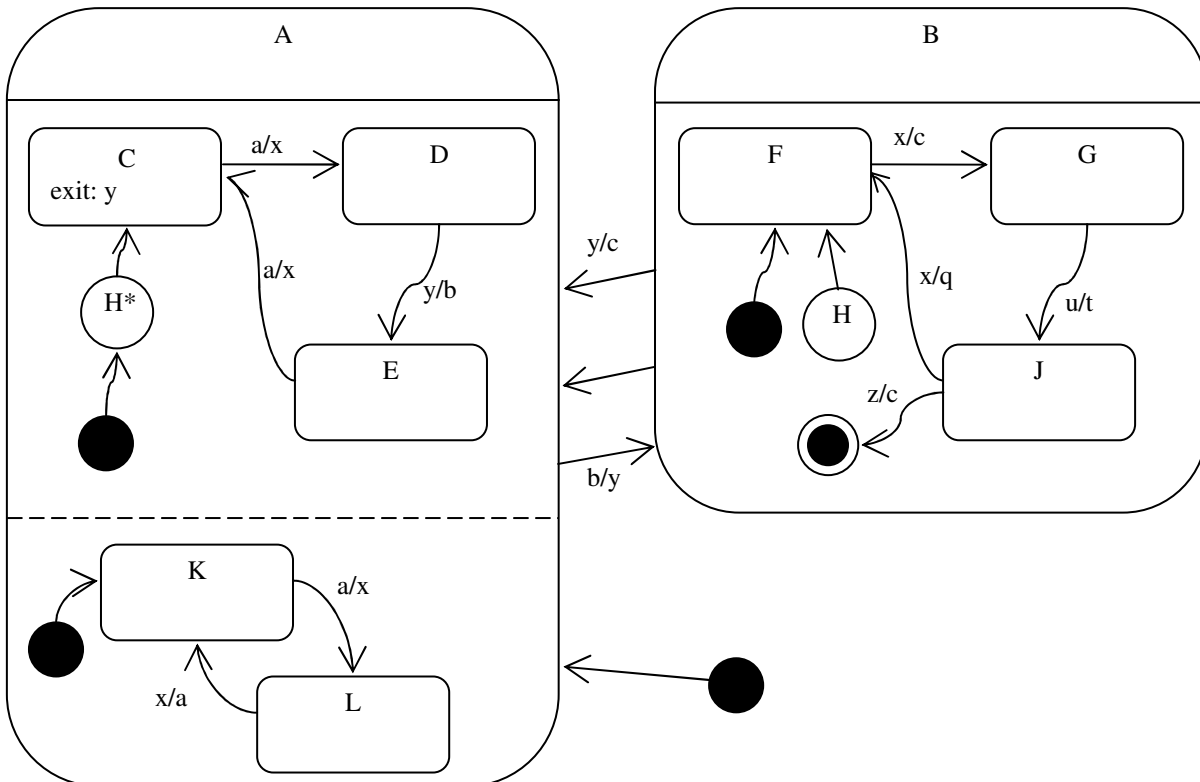
3. Töltse ki a táblázatot a Java gyűjtemény-keretrendszer osztály és interfészneveivel, amelyre igazak a táblázat peremén található állítások! Egy dobozba egy interfész és egy (az interfészt megvalósító) osztály nevét írja be! (3 pont)

	Elemek egyediek (unique)	Elemek nem egyediek (non unique)
Elemek rendezettek (ordered)	SortedSet, TreeSet	List, ArrayList, Vector, LinkedList
Elemek nem rendezettek (unordered)	Set, HashSet	---

Húzza alá azt a metódust, amelyik a fenti osztályok példányain meghívható! (1pont)

join, sleep, wait, interrupt, notify

4. A következő UML állapotdiagram alapján minősítse az állításokat! Csak a rubrikába tett jelzést vesszük figyelembe! (8 pont)



Igaz	Hamis	Állítás
<input type="checkbox"/>	<input checked="" type="checkbox"/>	H állapotból bármely esemény bekövetkeztekor F állapotba jutunk.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	B állapot elhagyásakor a 'c' tevékenység pontosan egyszer hajtódik végre.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	G-ből két lépésben eljuthatunk L-be.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	A J állapottal egyidőben lehetünk K-ban is.

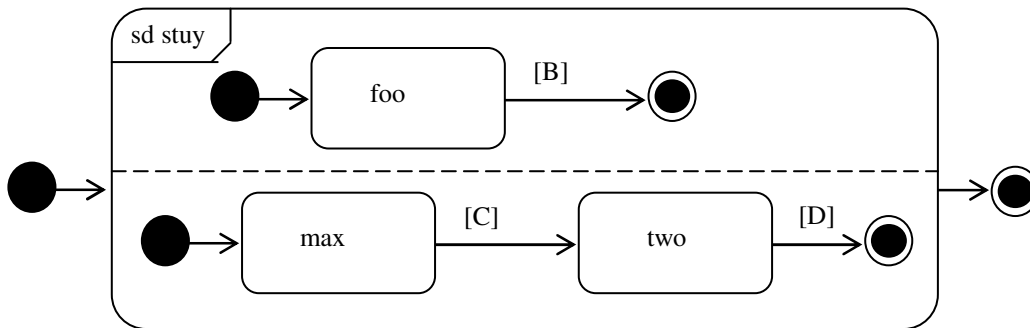
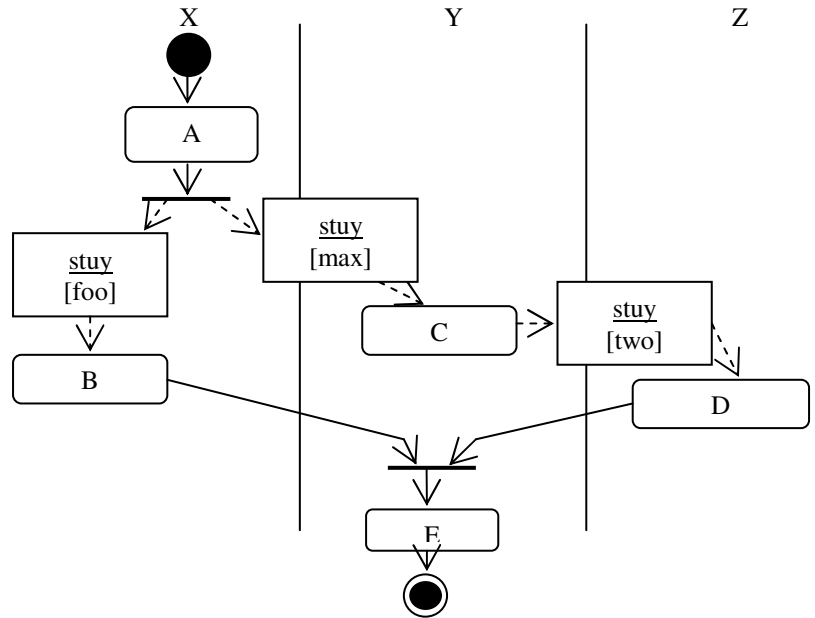
A kezdés után a következő esemény-szekvencia hatására: **a, b, x, y, a, b**

Igaz	Hamis	Állítás
<input type="checkbox"/>	<input checked="" type="checkbox"/>	pontosan kétszer fut le az 'x' tevékenység.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	a végén G állapotba kerülünk.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	pontosan kétszer fut le a 'c' tevékenység.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	pontosan kétszer érintettük az L állapotot.

5. Jelölje az állítások mellett 1-5-ig, hogy minimálisan melyik CMM szinttől igazak ! Ha az állítás nem értelmezhető, akkor tegyen X-et ! (4 pont)

2	Fejlett projekt-menedzsment technikákat és eszközöket alkalmaznak a napi gyakorlatban.
X	Koreográfia (choreography) elvén szervezik a technológiai folyamatokat.
X	A projektekben rendszeresen alkalmazzák az agilis programozást (pl. Scrum).
X	A vezetők képesek a termékek minőségének közvetlen, számszerű ellenőrzésére.

6. Adott a mellékelt – object flow-val kiegészített – aktivitás-diagram (activity diagram) ! Rajzolja meg azon objektumok UML2 state-chartját, amelyeknek az ábra alapján több állapota is van! (7 pont)



7. Jellemezzünk egy stringet az alábbi műveletekkel!

- END(s, x)** az s string végére rakja az x karaktert.
- CHR(s, x)** az s stringben található x karakterek előfordulásának számát adja.
- LGTH(s)** az s string karaktereinek számát adja.
- CRT()** új (üres) stringet hoz létre.
- TAIL(s)** az s string legrégebbi karakterének levágása után maradó stringet adja.

Az alábbi kifejezésekhez adja meg, hogy azok algebrai axiómák BAL oldalán állhatnak-e vagy sem! (5 pont)

Igen	Nem		Igen	Nem	
<input type="checkbox"/>	<input type="checkbox"/>	CHR(TAIL(s), x)	<input type="checkbox"/>	<input type="checkbox"/>	END(CRT(), x)
<input type="checkbox"/>	<input type="checkbox"/>	TAIL(CHR(s, x))	<input type="checkbox"/>	<input type="checkbox"/>	LGTH(TAIL(s))
<input type="checkbox"/>	<input type="checkbox"/>	CHR(END(s, x), x)	<input type="checkbox"/>	<input type="checkbox"/>	CHR(CRT(), 0)
<input type="checkbox"/>	<input type="checkbox"/>	CHR(END(s, y), x)	<input type="checkbox"/>	<input type="checkbox"/>	LGTH(CHR(s, x))
<input type="checkbox"/>	<input type="checkbox"/>	TAIL(CHR(s, x))	<input type="checkbox"/>	<input type="checkbox"/>	LGTH(TAIL(s))
<input type="checkbox"/>	<input type="checkbox"/>	LGTH(END(CRT(), x))	<input type="checkbox"/>	<input type="checkbox"/>	END(s, LGTH(s))

8. Sorolja fel a tesztelés során végrehajtandó „kiértékelés” (evaluation) folyamatnak a be és kimeneteit! (4 pont)

Bemenetek: **test results, expected results**.....

Kimenetek: **errors, error statistics**.....

9. Jelölje be az alábbi táblán, hogy az egyes szerződéses feltételek megszegése esetén melyik fél a hibás! (2 pont)

	kliens	szerver
invariáns (invariant)		X
utófeltétel (postcondition)		X

10. Adott az alábbi **Student** generikus Java osztály, amelyet tárgy (C) típusal lehet paraméterezni.

```
public class Student<C> implements Cloneable {
    private String name; // név
    private String neptun; // neptunkód
    private Set<C> courses; // felvett tárgyak
    public Student(String na, String ne) {
        name = na;
        neptun = ne;
        courses = new HashSet<C>();
    }
    public void setName(String s) { name = s; }
    public void addCourse(C c) { courses.add(c); }
    public void delCourse(C c) { courses.remove(c); }
}
```

Implementálja meg Javában a fenti **Student** osztály következő metódusait! Nem használható ciklus és a paraméterek állapotát nem módosíthatja!

```
public boolean equals(Object o) {
```

```
//Két hallgató akkor egyezik, ha a nevük és a neptunkódjuk is azonos. Feltételezheti,
//hogy a metódus csak Student dinamikus típusú objektumot kap paraméterül. (2 pont)
```

```
Student<C> s = (Student<C>)o;
return (s.name.equals(name)&& s.neptun.equals(neptun));
```

```
}
```

```
public Object clone() {
```

```
//visszatér egy olyan Student példánnyal, amelynek tartalma azonos az eredetivel, de a
//set, add és del kezdetű metódusok hívásakor csak a hívott objektum tartalma változik
//(deep clone). (3 pont)
```

```
Student<C> s = new Student<C>(name, neptun);
s.courses = new HashSet<C>(courses);
return s;
```

```
}
```

```
static <C> int numOfSharedCourses(Student<C> s1,
Student<C> s2) {
```

```
//visszatérési értéke a két hallgató közös tárgyainak darabszáma. (3 pont)
```

```
HashSet<C> hs = new HashSet<C>(s1.courses);
hs.retainAll(s2.courses);
return hs.size();
```

```
}
```

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

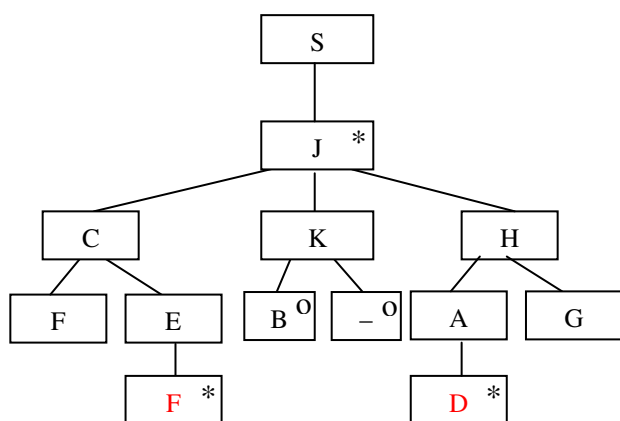
c. tárgyból

2013. június 18.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. Egészítse ki az alábbi ELH-t és állapottáblát úgy, hogy mindkettő ugyanazt a szerkezetet írja le! A kiegészítés során az **ábrák szerkezetén**, az azokba **beírt, berajzolt elemeken változtani tilos!** Az ELH-ban csak a két üresen maradt blokkot kell kitölteni! Az állapottáblát egészítse ki! Az induló állapot legyen az ①! (ELH 2 pont, állapottábla 3 pont)



	F	G	B	D
①	③			
②		①		②
③	③	①	②	②

2. Mire használjuk a JUnit-ban a `@AfterClass` annotációval megjelölt metódust? (2 pont)

A tesztosztály közös lezárása, erőforrások felszabadítása.....

Mit történt, ha a JUnit teszt eredménye *error*? (2 pont)

A tesztmetódus általa nem kezelt kivételt dobott

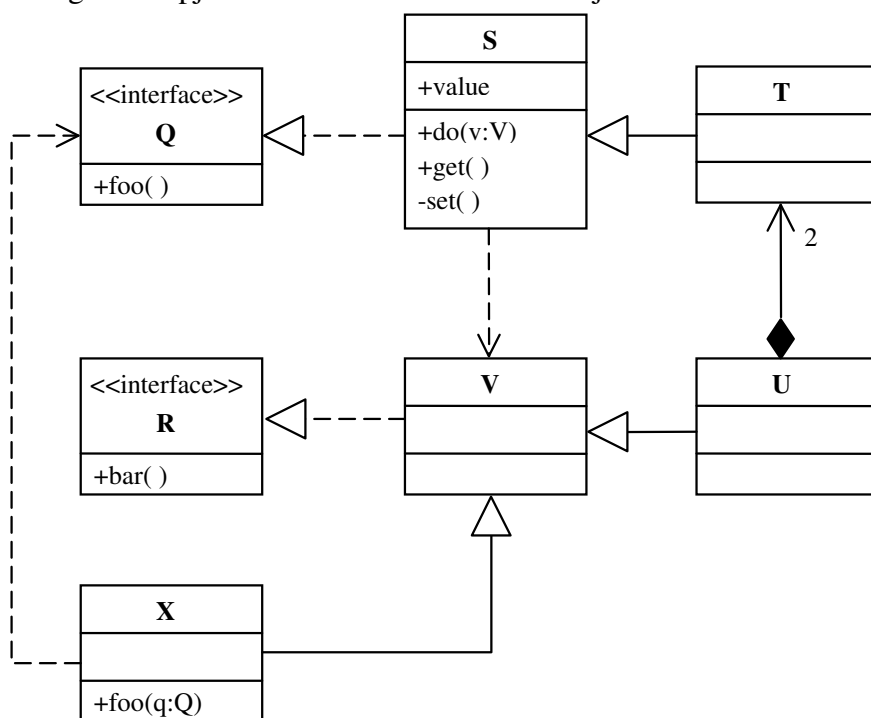
3. A szabványos Java nyelvben az alábbiak közül mely hívások hatására hagyja el biztosan a szál a futó állapotát? (4 pont)

- | | |
|--|---|
| <input type="checkbox"/> yield() | <input type="checkbox"/> sleep(1000) |
| <input checked="" type="checkbox"/> wait() | <input checked="" type="checkbox"/> stop() |
| <input type="checkbox"/> notify() | <input checked="" type="checkbox"/> egy másik élő szálra történő join() hívás |
| <input type="checkbox"/> notifyAll() | <input type="checkbox"/> Thread.killThread() |

4. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (3 pont)

- I** **H** egy szál egyszerre csak egy objektum monitorában tartózkodhat.
- I** **H** szálak nem képesek saját magukat közvetlenül *waiting* állapotból *notify*-jal felébreszteni.
- I** **H** egy változó statikus típusa nem lehet a változó dinamikus típusának leszármazottja.

5. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- [B]** T létrehozhat U osztályú objektumot, mert S létrehozhat V-t és T az S-nek, U a V-nek leszármazottja.
- [C]** X foo(q:Q) metódusa kaphat paraméterül T-t, mert T-nek is van foo() metódusa.
- [B]** X foo(q:Q) metódusa meghívhatja a paraméterül kapott S get() metódusát, mert S megvalósítja a Q interfészt.
- [B]** V törlésekor törölni kell két T-t is, mert egy U-nak két T komponense van és U a V leszármazottja.
- [A]** T nem függ U-tól, mert T nem függ V-től sem.
- [A]** X meghívhatja egy Q interfészes objektum foo() metódusát, mert X implementálja Q-t.
- [B]** X bar() metódusából meghívhatjuk egy Q interfészes objektum foo() metódusát, mert X foo(q:Q) metódusából is hívhatjuk egy Q interfészes objektum foo() metódusát.
- [B]** S set() metódusa nem módosíthatja a value attribútumot, mert a láthatóságuk különböző.

6. A kockázat tervezése során milyen stratégiákat választunk ? (3 pont)

A kockázat valószínűségének csökkentés

A bekövetkező káros hatások minimalizálása

A nem kívánatos eseményt követően felmerülő tevékenységek tervezése

7. Mit jelent a CMM ? (3 pont)

Capability Maturity Model

Jelölje meg az igaz állításokat ! (4 pont)

- a CMM egy szervezet által készített összes szoftver minőségét értékeli.
- a CMM szint kifejezi a szervezet vezetésének minőségét is.**
- CMM minősítést csak jó szoftverek kaphatnak.
- a CMM egy adott szoftver termék fejlettségét, érettségét vizsgálja.
- a CMM egy szervezetben zajló szoftver fejlesztési folyamatot értékeli.**
- egy jó szoftvertől elvárható a magas CMM szint.

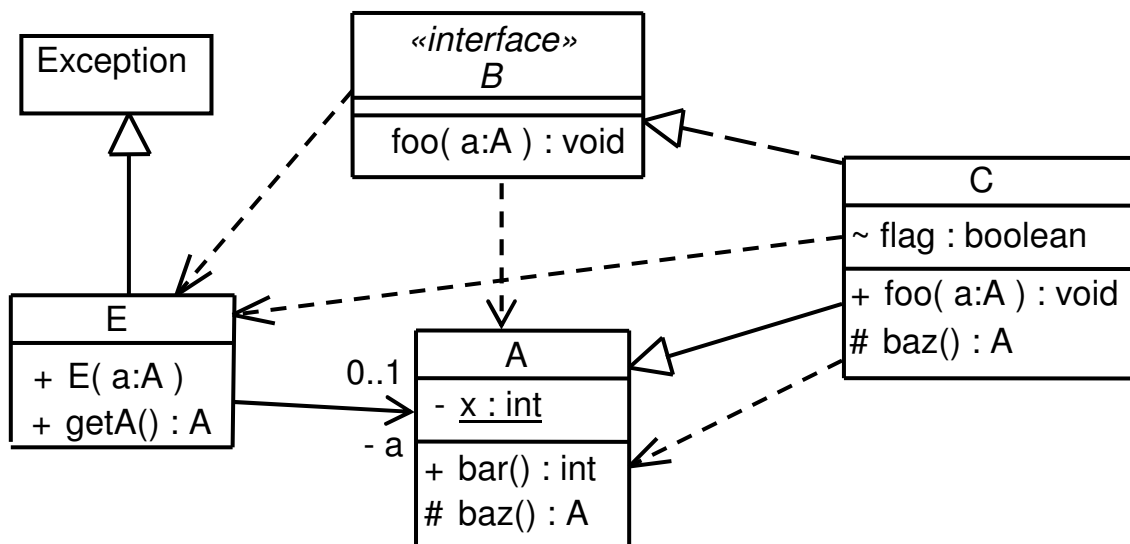
8. Rajzoljon UML 2 osztálydiagramot az alábbi Java kódrészlet alapján! (6 pont)

```
class A {
    private static int x = 0;
    public int bar() { return ++x; }
    protected A baz() {
        return new A(); } }

class E extends Exception {
    private A a;
    public E(A a) { this.a = a; }
    public A getA() { return a; } }

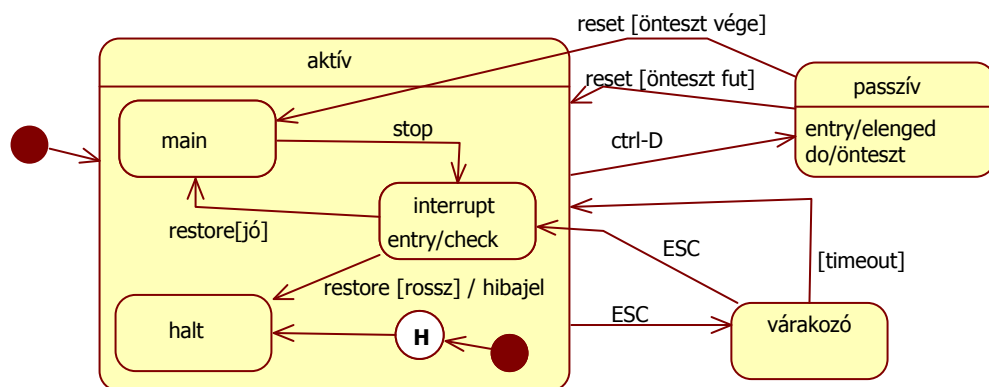
interface B {
    void foo(A a) throws E; }

class C extends A implements B {
    boolean flag;
    public void foo(A a) throws E {
        flag = !flag;
        if(a.bar()>5&&flag) throw new E(baz());
    }
    protected A baz() { return new C(); } }
```



9. Rajzoljon UML2 állapot diagramot! (8 pont)

A Dowswin program futása során aktív, passzív és várakozó lehet. Aktivitás közben lehet main, interrupt és halt helyzetben. Main-ből a stop hatására interrupt-ba megy. Az interrupt-ba belépéskor lefuttat egy memory check-et. Az interrupt-ból a restore hatására lép ki. Ha a belépéskor végrehajtott memory check jó volt, akkor main-be lép, ha nem, akkor halt-ba. Halt-ba lépés közben hibajelzést küld a konzolra. Aktívból ctrl-D hatására passzíválódik. Passzív állapotba lépéskor elengedi az erőforrásokat, majd egy hosszantartó öntesztelést végez. Reset hatására újra aktivizálódik. Ha az öntesztelés befejeződött, akkor main-be lép, ha nem, akkor a ctrl-D érkezése előtti helyzetbe tér vissza. Aktivitásból várakozásba megy át, ha ESC érkezik. A várakozás végetér egy újabb ESC előfordulásakor. Ekkor a működést interrupt-ban folytatja. A várakozás egy timeout letelte után szintén befejeződik. Ez esetben a várakozást megelőző helyzetben folytatódik a működés. A program halt helyzetből indul.



10. Mi a “fan-out”? (2 pont)

- egy adott modul (osztály) döntési hatáskörébe tartozó modulok (osztályok) száma
- egy adott modulban (osztályban) használt paraméterek száma
- egy adott modul (osztály) vezérlési hatáskörébe tartozó modulok (osztályok) száma
- egy adott modul (osztály) által használt modulok (osztályok) száma
- egy adott modult (osztályt) használó más modulok (osztályok) száma

Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

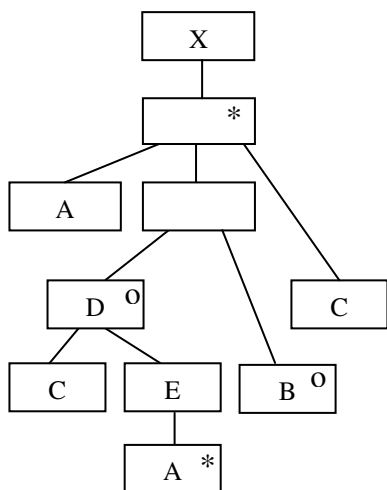
c. tárgyból

2014. január 7.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. Egészítse ki az **állapottáblát** úgy, hogy az ELH-nak megfelelő szerkezetet írja le! A kiegészítés során **csak az állapottábla üres blokkjaiba írhat**, bármi egyebet az ELH-n vagy az állapottáblán változtatni tilos! Az induló állapot legyen az ① ! (5 pont)



	A	B	C	
①	②			
②		④	③	
③	③		①	
④			①	

Az alábbiak közül mely deklarációk hiányoznak az ELH-t leíró DTD-ből? (3 pont)

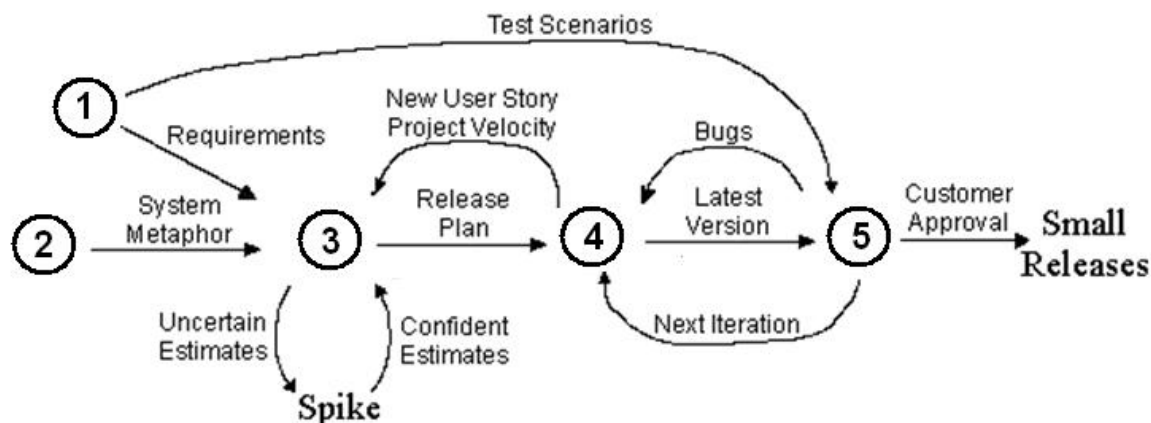
- X (A, Q, C) *
- Q (D|B)
- D (C, E)
- D (A) +
- E (A) *
- Y (C, A*|B?)
- Y ((C, A) *|B)
- X (A, (D|B), C) *
- D (C, E, A*|B)
- X (A, D?, C) *

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<!DOCTYPE X [
  <!ELEMENT A (#PCDATA)>
  <!ELEMENT B (#PCDATA)>
  <!ELEMENT C (#PCDATA)>
]>
```

2. JUnit-tal tesztelni akarjuk a nem kezelt kivételt. Az alábbi lehetőségek közül mely tesztmetódusok futnak le sikeresen, ha a program nem kezelt ArithmeticException (pl. nullával osztáskor) kivételt dob? Jelölje X-szel a helyes választ! (2 pont)

	@Test(throwable=ArithmeticException) public void test() {program, ami kivételt dob}
	@Test public void test()throws ArithmeticException {program, ami kivételt dob}
X	@Test(expected=ArithmeticException.class) public void test() {program, ami kivételt dob}
	JUnit-tal nem kezelt kivétel nem tesztelhető

3. Az Extrém Programozás (XP) projektet írja le az alábbi (hiányos) ábra. Adja meg azoknak az elemeknek a számát és nevét, amelyek a Rational Unified Process (RUP) alapelveihez közvetlenül kapcsolódnak (3 pont)



száma	XP elem neve	RUP alapelv
1	User stories	Use case vezérelt
2	Architectural spike	Architektúra központú
4	Iteration	Iteratív, inkrementális

4. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (9 pont)

- I H *final* osztálynak nem lehet *abstract* metódusa.
- I H minden generikus osztály használható *Object* paraméterezéssel.
- I H generikus osztály paramétere nem lehet primitív típus.
- I H primitív típus tömbje is a primitív típusok közé számít.
- I H ha egy szál véget ért, akkor *start()* metódushívással újraindítható.
- I H egy szál csak akkor hajthat végre *wait()* metódushívást, ha a hívott objektum monitorában tartózkodik.
- I H ha egy szál *notify()* hívás hatására hagyja el a *WAIT* állapotot, akkor *RUNNABLE* állapotba kerül.
- I H *String* objektum tartalma bármikor megváltoztatható.
- I H egy osztály statikus metódusát csak statikus metódusból lehet meghívni.

5. Ha metódus meghívásakor a kliens oldalon hibázunk, az mely szerződéses feltételek megsértését jelenti? (2 pont)

előfeltétel (precondition)

6. Rajzoljon UML2 állapotdiagramot a Q osztályhoz! (5 pont)

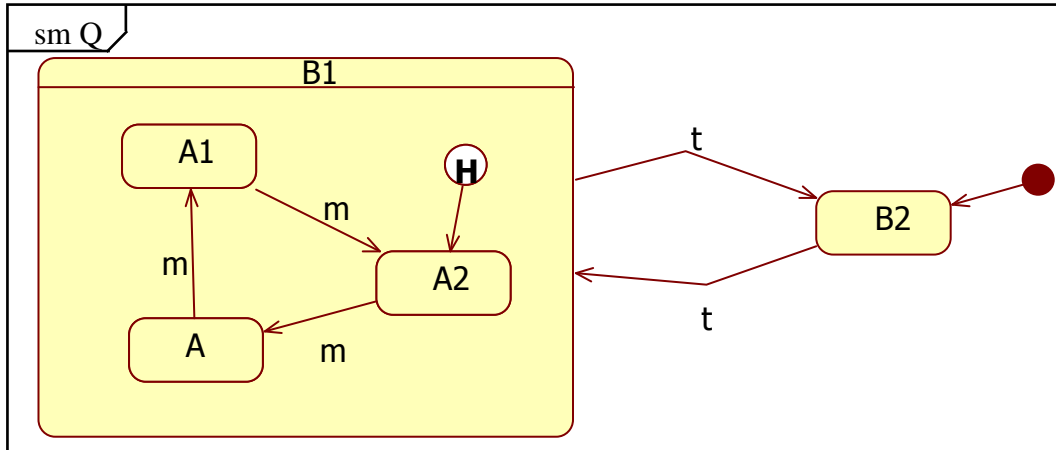
```

enum AS { A1, A2, A3 }
enum BS { B1, B2 }

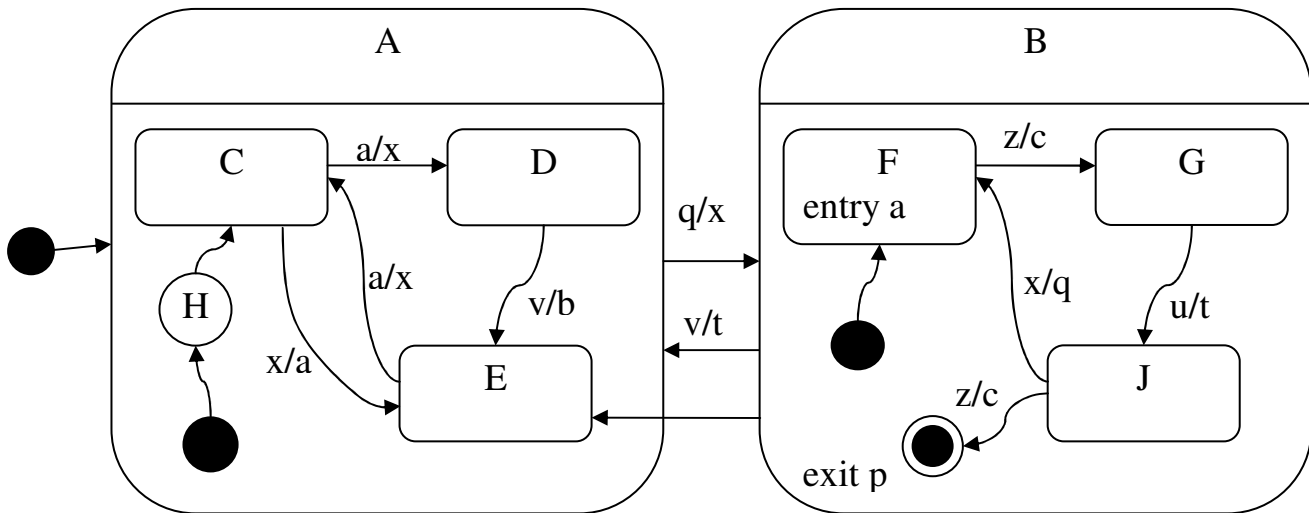
class Q {
private AS as = AS.A2;
private BS bs = BS.B2;

public void t() {
if (bs == BS.B1) {bs = BS.B2;}
else {bs = BS.B1;}
}
}

public void m() {
if (bs == BS.B1) {
switch (as) {
case A1: as = AS.A2; break;
case A2: as = AS.A3; break;
case A3: as = AS.A1; break;
}
}
}
}
    
```



7. A következő UML2 állapotdiagram alapján minősítse az állításokat! (8 pont)



Igaz	Hamis	Állítás
<input type="checkbox"/>	<input type="checkbox"/>	D állapotból 2 lépésben visszaérhet D-be
<input type="checkbox"/>	<input checked="" type="checkbox"/>	J állapotból „p” esemény hatására E állapotba kerül
<input type="checkbox"/>	<input checked="" type="checkbox"/>	F állapotból „v” esemény hatására H állapotba kerül
<input checked="" type="checkbox"/>	<input type="checkbox"/>	A-ból B-be való váltáskor mindig végrehajtódik az „a” tevékenység

A kezdés után az **x, q, z, u, z** esemény-szekvencia hatására

Igaz	Hamis	Állítás
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Egyszer lefut a „p” tevékenység
<input type="checkbox"/>	<input checked="" type="checkbox"/>	C állapotba kerülünk
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Az E állapotot kétszer érintjük
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Kétszer lefut az „a” tevékenység

8. Az alábbi állapotgéppel specifikált programot az interpreter architektúra szerint kívánjuk megvalósítani.

	A	B	C
D	D/E	D/-	F/G
F	D/-	D/E	D/-

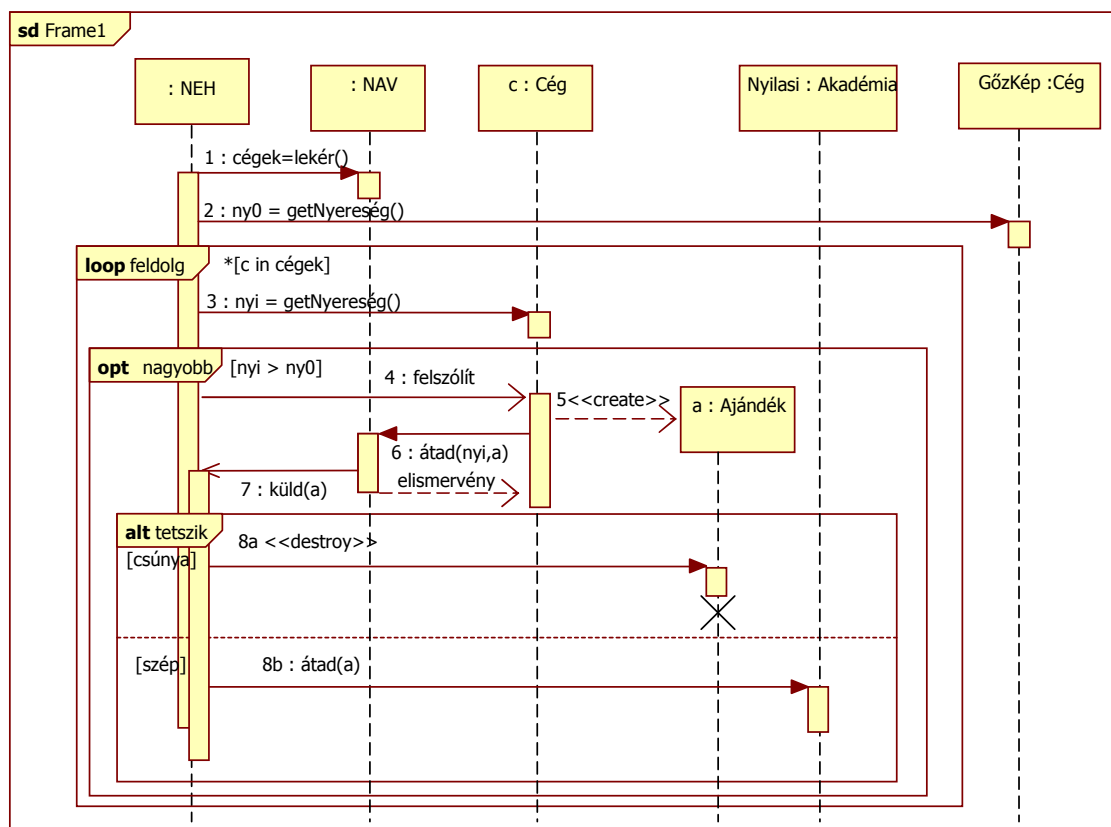
Specifikálja a

“pseudocode” utasításait ! (3 pont): **D/E, D/-, F/G**.....

“control state” lehetséges értékeit ! (2 pont): **[D|F]**.....

9. Készítsen UML2 szekvenciadiagramot az alábbi történet alapján! (8 pont)

A Nemzeti Egyensúly Hivatal (NEH) lekéri a NAV-tól a bekötött pénztárgép cégek listáját. A listán szereplő cégektől elkéri az éves eredményt, és amelyik cégnél a GőzKép ZRt-ét meghaladó nyereség keletkezett, levélben felszólítást küld. A cégnek a felszólítás hatására azonnal kis ajándékot kell készítenie, és az ajándékkal együtt át kell adnia a nyereségét a NAV-nak. A NAV az ajándékot azonnal továbbküldi a NEH-nek, majd a cég felé elismervénnyel nyugtázza az adomány átvételét. Ha az ajándék csúnya, a NEH megsemmisíti, ha szép, akkor a Nyilasi Akadémiának adja át.



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5