

<b>MI</b>	Név, felvételi azonosító, Neptun-kód: <b>MEGOLDÁS</b>	pont(45) :
-----------	--	------------

Csak felvételi vizsga: <input type="checkbox"/>	csak záróvizsga: <input type="checkbox"/>	közös vizsga: <input type="checkbox"/>
---	---	--

## Közös alapképzéses záróvizsga – mesterképzés felvételi vizsga Mérnök informatikus szak BME Villamosmérnöki és Informatikai Kar

**2013. május 28.**  
**MEGOLDÁSOK**

A dolgozat minden lapjára, a kerettel jelölt részre írja fel nevét, valamint felvételi azonosítóját, záróvizsga esetén Neptun-kódját!

A fenti táblázat megfelelő kockájában jelölje X-szel, hogy csak felvételi vizsgát, csak záróvizsgát, vagy közös felvételi és záróvizsgát kíván tenni!

A feladatok megoldásához csak papír, írószer, zsebszámológép használata megengedett, egyéb segédeszköz és a kommunikáció tiltott. A megoldásra fordítható idő: 120 perc. A feladatok után azok pontszámát is feltüntettük.

A megoldásokat a feladatlpra írja rá, illetve ott jelölje. Teszt jellegű kérdések esetén elegendő a kiválasztott válasz betűjelének bekarikázása. Kiegészítendő kérdések esetén, kérjük, adjon világos, egyértelmű választ. Ha egy válaszon javítani kíván, teszt jellegű kérdések esetén írja le az új betűjelet, egyébként javítása legyen egyértelmű.

A feladatlpra írt információk közül csak az eredményeket vesszük figyelembe. Az áttekinthetetlen válaszokat nem értékeljük.

A vizsga végeztével mindenképpen be kell adnia dolgozatát. Kérjük, hogy a dolgozathoz más lapokat ne mellékeljen.

Felhívjuk figyelmét, hogy illegális segédeszköz felhasználása esetén a felügyelő kollegák a vizsgából kizárják, ennek következtében felvételi vizsgája, illetve záróvizsgája sikertelen lesz, amelynek letételét csak a következő felvételi, illetve záróvizsga-időszakban kísérelheti meg újból.

### Szakirányválasztás

(Csak felvételi vizsga esetén kell kitölteni)

Kérem, az alábbi táblázatban jelölje meg, mely szakirányon kívánja tanulmányait folytatni. A táblázatban a szakirány neve mellett számmal jelölje a sorrendet: 1-es szám az első helyen kiválasztott szakirányhoz, 2-es a második helyen kiválasztotthoz tartozik stb. Nem kell az összes szakirány mellé számot írni, de legalább egy szakirányt jelöljön meg. Egy sorszám csak egyszer szerepeljen.

szakirány neve	gondozó tanszék	sorrend
Alkalmazott informatika szakirány	AAIT	
Autonóm irányító rendszerek és robotok szakirány	IIT	
Hálózatok és szolgáltatások szakirány	TMIT	
Hírközlő rendszerek biztonsága szakirány	HIT	
Intelligens rendszerek szakirány	MIT	
Médiainformatika szakirány	TMIT	
Rendszerfejlesztés szakirány	IIT	
Számításelemélet szakirány	SZIT	
Szolgáltatásbiztos rendszertervezés szakirány	MIT	

<b>AL</b>	Név, felvételi azonosító, Neptun-kód:	pont(15):
	MEGOLDÁS	

1. Az  $f(n)$  és  $g(n)$  függvényekről tudjuk, hogy  $f(n) = O(n^2)$  és  $g(n) = O(n^2)$ . Következik-e ebből, hogy van olyan  $c > 0$  konstans, mellyel minden  $n$ -re teljesül, hogy

$f(n) = c \cdot g(n)$  igen — nem

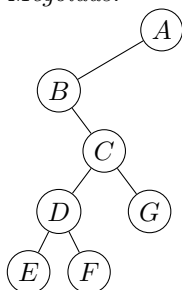
$|f(n)| < c \cdot |g(n)|$  igen — nem

$|f(n)| > c \cdot |g(n)|$  igen — nem

pont(1):

2. Egy 7 csúcsú bináris fa preorder bejárásakor a csúcsok sorrendje:  $A, B, C, D, E, F, G$ , inorder bejárásakor pedig:  $B, E, D, F, C, G, A$ . Rajzolja fel, hogyan nézhet ki a fa!

Megoldás:



pont(1):

3. Az  $1, 2, \dots, n$  számozott pontokon a  $G$  gráf egy olyan csillag, melynek középpontja az  $n$  csúcs, az  $1, 2, \dots, n-1$  csúcsok pedig a végpontjai ( $n > 2$ ). Hány (nem üres) részfa található  $G$ -ben?

Megoldás:  $2^{n-1} + n - 1$

(A középpontot tartalmazók megfelelnek a végpontok részhalmazainak; a középpont nélkül csak 1 pontú fák vannak.)

pont(2):

4. Az alábbi hash-táblán a  $h(x) = x \pmod{11}$  hash-függvény segítségével lineáris próbával hajtsa végre a BESZÚR(3), TÖRÖL(20), TÖRÖL(7) műveleteket és jelölje a változásokat! Ezek után a táblázat melyik mezőjében ér véget a KERES(25) művelet?

0	1	2	3	4	5	6	7	8	9	10
3	23	2	14	4	16		x		x	21

8-ban (a látott értékek: 14-2-23-3-21-x-.)

pont(2):

5. Egy nagyobb társaságnak autóbuzsos kirándulást szervezünk. Olyan beosztást akarunk készíteni, hogy ha  $X$  és  $Y$  barátok, akkor ők biztos egy buszra kerüljenek. Ehhez egyforma méretű ( $n$  személyes) buszok állnak rendelkezésünkre, amelyekkel tudjuk, hogy a feladat megoldható.

Melyik ismert feladattal írható le a probléma, ha az a célunk, hogy minél kevesebb buszt használjunk? (Nem kell a beosztást elkészíteni, csak a feladatot kell megnevezni!)

Megoldás: Ládapakolás (A buszok a ládák, a barátságok összefüggő komponensei a tárgyak, méretük: létszám/ $n$ .)

pont(2):

6. Egy úthálózatot egy irányítatlan gráffal adtunk meg, melyben a csúcsok a városok, az élek a köztük menő utak. A városok egy részénél a városba belépéskor dugódíjat kell fizetni, minden csúcshoz adott, hogy ott mennyit. (Tegyük fel, hogy az utaknak nincs díja.) Azt szeretnénk meghatározni, hogy az  $A$  városból hogyan lehet úgy eljutni a  $B$  városba, hogy minél kevesebbet fizessünk.

Milyen ismert algoritmust, milyen bemeneten futtatva lehet hatékonyan megoldani a problémát?

*Megoldás:* A gráfból csináljunk egy irányított gráfot, amiben minden él súlya 0. Minden csúcstól húzzunk szét két ponttá, az így keletkező él súlya legyen a városi dugódíj. Ezen a gráfon Dijkstra algoritmus megoldja a kérdést.

pont(2):

7. Az alábbi  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $\mathcal{C}$  problémák esetén melyik állítás igaz és melyik nem?

$\mathcal{A}$ : Adott egy  $G$  gráf, valamint adott egy  $k > 0$  egész szám.

Kérdés, hogy van-e  $G$ -ben legalább  $k$  élű út.

$\mathcal{B}$ : Adott egy  $G$  gráf, valamint adott egy  $k > 0$  egész szám.

Kérdés, hogy van-e  $G$ -ben legfeljebb  $k$  élű út.

$\mathcal{C}$ : Adott egy  $G$  gráf.

Kérdés, hogy van-e  $G$ -ben pontosan 2013 élből álló út.

a) Ha  $\mathcal{A}$  polinomiálisan visszavezethető (Karp-redukálható)  $\mathcal{C}$ -re, akkor  $P = NP$ .

igaz — hamis

b) Ha  $P \neq NP$ , akkor  $\mathcal{A}$  nem vezethető vissza polinomiálisan (nem Karp-redukálható)  $\mathcal{B}$ -re.

igaz — hamis

c) Ha  $\mathcal{A}$  polinomiálisan visszavezethető (Karp-redukálható)  $\mathcal{C}$ -re, akkor  $\mathcal{A}$  polinomiálisan visszavezethető (Karp-redukálható)  $\mathcal{B}$ -re.

igaz — hamis

pont(2):

8. Javasoljon olyan adatszerkezetet könyvtári kölcsönzések nyilvántartásához, amely az alábbi négy művelet mindegyikét,  $n$  kölcsönző személy esetén  $O(\log n)$  időben megvalósítja:

KERES(kölcsönző): a személy azonosítója alapján megadja az illető által kikölcsönzött könyvek azonosítóit. Egy személynél egyszerre legfeljebb 8 könyv lehet.

KINÉL(könyv): a könyv azonosítója alapján megadja a kölcsönző személy azonosítóját (ha ki van kölcsönözve).

ÚJ(kölcsönző, könyv): egy új kölcsönzést beilleszt az adatszerkezetbe.

VISSZAAD(kölcsönző, könyv): törli az erre a személyre és könyvre vonatkozó bejegyzést.

Röviden vázolja, hogy a javasolt adatszerkezet miért működik az előírt  $O(\log n)$  időben.

*Megoldás:* Két kiegyensúlyozott keresőfát (vagy 2-3 fát) tartunk. Egyik a kölcsönző azonosítója szerint van rendezve (a csúcsban tárolva a legfeljebb 8 kikölcsönzött könyv), a másik fában a könyvek azonosítója szerint van a rendezés, egy mutatóval minden csúcsban az első fa megfelelő elemére is.

KERES: keresés az első fában, KINÉL: keresés a második fában. Az ÚJ és VISSZAAD egy beszúrás, illetve törlés a második fában. Az első fában, ha ez az első, utolsó könyv az illetőnek, akkor egy csúcs beszúrása, törlése, egyébként a megfelelő csúcsnál levő lista módosítása.

Az első fa  $n$  csúcsú,  $O(\log n)$  idejűek a műveletei. Mivel  $n$  személy esetén legfeljebb  $8n$  könyv lehet kikölcsönözve, a második fában a műveletek  $O(\log(8n)) = O(\log n)$  idejűek.

pont(3):

<b>H</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(7,5):
----------	---	------------

*Figyelem! Ha egy feladatnál több helyes válasz van, minden helyeset be kell jelölni!*

1. Mi a neve annak a sokportos eszköznek, amely jellemzően azonos szabványú hálózatokat vagy gépeket köt össze a második rétegben, s a kereteket igyekszik csak a szükséges irányba továbbítani!

- a) Router (útválasztó).
- b) Repeater (ismétlő).
- c) Gateway (átjáró).
- d) Switch (kapcsoló).
- e) A fentiek közül egyik sem a helyes elnevezés.

Megoldás: **d)**

pont(1):

2. Milyen szerepet játszanak az autonóm rendszerek (AS) az Internet útvonal-irányításában?

- a) Megnehezítik az egységes (konzisztens) kép kialakítását a hálózatról, mert belsejükben szabadon dönthetnek az alkalmazott routing protokollról.
- b) Lehetővé teszik a routerek számára, hogy felhasználói végpontok nagyszámú csoportjait csak egy-egy bejegyzésként kezeljék az útvonaltábláikban.
- c) Megnehezítik a gerincrouterek dolgát, hogy helyesen állapítsák meg a végpontok felé vezető legalkalmasabb utat.
- d) Lehetővé teszik a routing feladatok hierarchikus megosztását a világméretű hálózatban.
- e) A többi válasz közül egyik sem helyes.

Megoldás: **b), d)**

pont(1):

3. Mit jelent *tartalmában* a Classless Inter-Domain Routing?

- a) Osztály nélküli tartományközi útválasztást.
- b) Az A, B és C osztályú címek merev NetID/HostID-re való osztásának a megszüntetését.
- c) AS-ek (autonóm rendszerek) közötti útválasztási módszert.
- d) A többi válasz közül egyik sem helyes.

Megoldás: **b)**

pont(1):

4. Mi a feltétele annak, hogy a rétegzett protokollarchitektúra valamely rétegében a megvalósítást megváltoztassuk?

- a) A szomszéd rétegek hozzájárulása szükséges.
- b) Csak a felette lévő réteg – amelynek szolgáltatást nyújt – hozzájárulására van szükség.
- c) Csak az alatta lévő réteg – amelynek szolgáltatást nyújt – hozzájárulására van szükség.
- d) Csak a mellette lévő réteg – amelynek szolgáltatást nyújt – hozzájárulására van szükség.
- e) A többi válasz közül egyik sem jó.

Megoldás: **e)**

pont(1):

---

5. Milyen szolgáltatás(oka)t *nem* nyújt az UDP protokoll?

- a) Sorrendhelyes átvitel
- b) Portkezelés
- c) Torlódásvezérlés
- d) Hibavédő kódolás a teljes UDP PDU-ra.
- e) A fentiek közül az UDP mindegyik szolgáltatást nyújtja.

Megoldás: a), c)

pont(1):

---

6. Mely tevékenység(ek) jellemzi(k) a CSMA/CD-protokollt?

- a) Az állomás figyeli a csatornát („vívót”).
- b) Ha az állomás nem érzékel adást a csatornán, elkezd küldeni a keretet.
- c) Ha kettő vagy több állomás ad ütközést okozva, mindegyik abbahagyja az adást.
- d) Ütközés után valamekkora (véletlen) késleltetést követően az állomás újból megkísérli az adást.
- e) A többi válasz egyike sem helyes.

Megoldás: a), b), c), d)

pont(1):

---

7. Egészítse ki az alábbi mondatot:

„A névfeloldás az Interneten a hálózati csomópont neve és annak  $\overset{1}{\dots}$  címe között teremt egyértelmű kapcsolatot. Ennek megvalósítását egy hierarchikus név- és címrendszer kialakításával a  $\overset{2}{\dots}$  végzi. Ebben a hierarchikus rendszerben elvileg először a rootnak megy a névfeloldási kérés, az továbbítja a kérést az aktuális tartományba az azt kezelő  $\overset{3}{\dots}$ -nek. A terhelés csökkenthető például szerver-replikációval vagy  $\overset{4}{\dots}$ -eléssel. Ezen utóbbi eljárás azt jelenti, hogy ha a végpont vagy ügynöke felold egy nevet, akkor ennek az eredménye tárolásra kerül az ismételt kérések kiszolgálásának felgyorsítása végett.”

A hiányzó elnevezések – *Megoldás:*

- 1: IP
- 2: DNS
- 3: name server (névszerver)
- 4: cache

pont(1,5):

<b>O</b>	Név, felvételi azonosító, Neptun-kód:  <b>MEGOLDÁS</b>	pont(7,5):
----------	--	------------

*Figyelem! Minden feladatnál csak egy helyes válasz van!*

1. Az alábbi megállapítások közül melyik *hamis* az időosztásos (time-sharing) rendszerekre?

- a) Az időosztásos rendszerek köteget (batch) feladatok futtatására is képesek.
- b) Az időosztásos rendszerek mindig prioritásosak.
- c) Az időosztásos rendszerek kifejlesztésének fő célja az volt, hogy on-line hozzáférést biztosítsanak a felhasználóknak a számítógépek szolgáltatásaihoz.
- d) A rendszer válaszideje fontos tulajdonsága az időosztásos rendszereknek.

Megoldás: **b)**

pont(1):

2. Az alábbi állítások közül melyik *igaz* az operációs rendszerek belső szerkezetével kapcsolatban?

- a) Az operációs rendszerek tipikusan a 7 rétegű ISO/OSI modell szerint épülnek fel.
- b) Az operációs rendszer hardverfügő részei a HAL-ban (Hardware Abstraction Layer) vannak megvalósítva.
- c) Az operációs rendszer magjának (kernel) fő feladata többek között a taszkkezelés és a memóriakezelés megvalósítása.
- d) Az operációs rendszer és a rajta futó alkalmazások függvényhívásokkal kommunikálnak egymással.

Megoldás: **c)**

pont(1):

3. Az alábbi állítások közül melyik *hamis* a szála (thread)?

- a) A szálnak saját verme (stack) van.
- b) A szál magában szekvenciális kódot hajt végre.
- c) Egy operációs rendszerben lévő két tetszőleges szál között lehetséges a kommunikáció közös memória alkalmazásával.
- d) A szálak alkalmazása lehetővé teszi egy folyamat számára, hogy az kihasználjon több rendelkezésre álló végrehajtóegységet (skalázhatóság).

Megoldás: **c)**

pont(1):

4. Az alábbi állítások közül melyik *hamis* folyamatokra (process)?

- a) A folyamat végrehajtás alatt álló program.
- b) Az operációs rendszeren belül a folyamatok szülő-gyermek viszonyban vannak.
- c) A szülő folyamat mindig korábban áll le (terminálódik), mint a gyermek folyamat.
- d) A folyamatok elindítása és leállítása erőforrás-igényes.

Megoldás: **c)**

pont(1):

---

5. Az alábbi, holtponttal (deadlock) kapcsolatos állítások közül melyik *igaz*?

- a) A holtpontban lévő feladatok mindegyike futásra kész állapotban (READY) van, de nem kap CPU-t, mert az állandóan foglalt.
- b) A holtpont kialakulásának szükséges feltétele az, hogy a feladatok szemaforokat használjanak a kölcsönös kizárás megvalósítására.
- c) A holtpont megelőzése során a rendszer minden erőforrásigény kielégítése előtt mérlegeli, hogy a rendszer biztonságos állapotban marad-e.
- d) A holtpont megelőzhető, ha ez erőforrásokat sorszámozzuk, és azokat az azokért versengő folyamatok csak a sorszámuk sorrendjében foglalják le.

Megoldás: **d)**

pont(1):

---

6. Az alábbi, a lapszervezéssel kapcsolatos kérdések közül melyik *hamis*?

- a) A lapszervezés során a fizikai memóriát azonos méretű fizikai memóriakeretekre osztjuk.
- b) A laptáblát is a fizikai memóriában tároljuk többnyire.
- c) A lapszervezés esetén nincs belső tördelődés.
- d) Az érvényesség (valid) bit a laptáblában azt jelzi, hogy az adott laphoz rendeltek-e fizikai memória keretet.

Megoldás: **c)**

pont(1):

---

7. Az alábbi, az állományok permanens táron történő tárolásával kapcsolatos állítások közül melyik *igaz*?

- a) A permanens táron a fizikai információátvitel egysége a fájl.
- b) A láncolt listás tárolás támogatja az egy fájlon belüli információ gyors közvetlen elérését.
- c) Az indexelt tárolás esetén a fájl egy részének tárolására szolgáló blokk tárolókapacitásának egy részét a fájl következő részeit tartalmazó blokk azonosítására használjuk (indexeljük).
- d) Az indexelt tárolás esetén az egy fájlhoz tartozó index blokkok számát nem tudjuk előre, így biztosítani kell egy fájlhoz tartozó index blokkok számának növelését és csökkentését.

Megoldás: **d)**

pont(1):

---

8. Az alábbi két állítás közül melyik *igaz*?

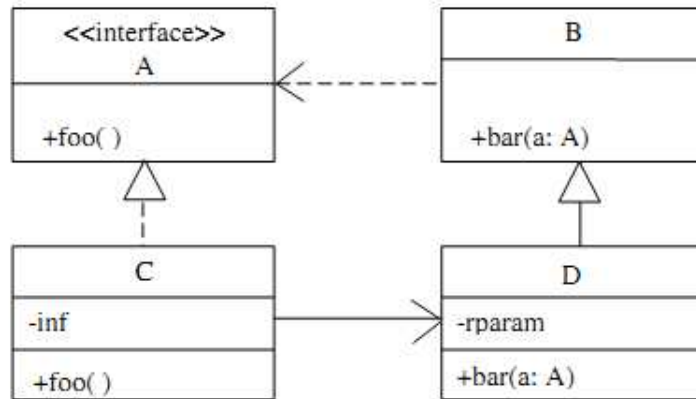
- a) A legrégebbi lap (FIFO) algoritmus laptábla-stratégia alkalmazása során a fizikai memóriakeretek számának növelése növekvő laphiba gyakoriságot eredményez (Bélády-anomália).
- b) A legrégebben nem használt (LRU) laptábla-stratégia közelíti meg legjobban a optimális algoritmus, de a megvalósítása nehéz, ezért annak csak közelítéseit szokták a gyakorlatban alkalmazni.

Megoldás: **a)**

pont(0,5):

<b>S1</b>	Név, felvételi azonosító, Neptun-kód: <b>MEGOLDÁS</b>	pont(5):
-----------	--	----------

1. Az alábbi UML2 diagram alapján – a kulcs felhasználásával – jellemezze az állítást!



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

D osztályú objektum `bar(a:A)` metódusa meghívhatja egy C `foo()` metódusát, mert D örökli az A-ra irányított asszociációt B-től.

Megoldás: **C**

pont(1):

2. Nevezzen meg egy dinamikus verifikációs technikát!

Megoldás: tesztelés

pont(1):

3. A szoftverfejlesztés melyik fázisának célja „a követelményeket kielégítő rendszer magas absztrakciós szintű formális leírása”?

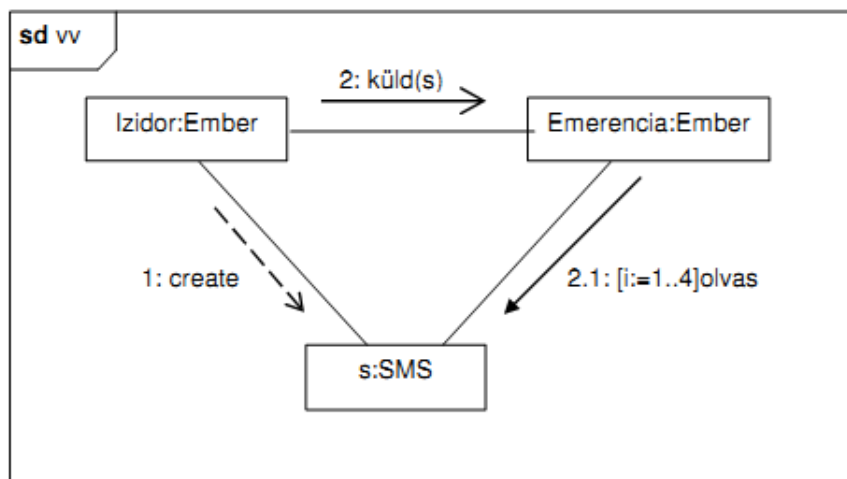
Megoldás: specifikáció

pont(1):



4. Izidor ír egy szerelemes SMS-t, amit elküld Emerenciának. Emerencia az SMS-t azonnal négyszer elolvassa. Rajzoljon UML2 kommunikációs diagramot!

Megoldás:

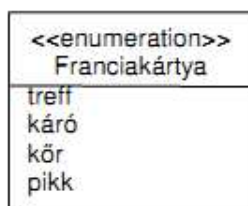


pont(1):

5. Definiálja UML2-ben az alábbi felsorolást (enumeráció)!

Franciakártya = [treff | káró | kőr | pikk]

Megoldás:



pont(1):

<b>S2</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(5):
-----------	---	----------

1. Adja meg két-három pontban, miben és hogyan segítenek a *tervezési* minták a szoftvertervezés során!  
Figyelem: *ne* a tervezési minta definícióját adja meg!

*Megoldás:*

- Növelik a rendszer karbantarthatóságát, módosíthatóságát;
- növelik az egyes részek újrafelhasználhatóságát;
- segítenek megtalálni a nem maguktól értetődő osztályokat.

pont(1):

- 
2. Milyen két általános problémát old meg a Singleton (Egyke) tervezési minta?

*Megoldás:* Kikényszeríti, hogy egy adott osztályból csak egyetlen objektumot lehessen létrehozni, és ehhez globális hozzáférést biztosít.

pont(1):

- 
3. Mutasson egy C++, Java vagy C# kódrészletet a Singleton tervezési minta implementálására, és mutasson példát a mintának megfelelő osztály használatára!

*Megoldás:* Egyszerű C++ nyelvű megoldás:

```
class Singleton {
public:
    static Singleton* Instance()
    {
        if (_instance == 0) {
            _instance = new Singleton;
        }
        return _instance;
    }
    void Print() { ... }
protected:
    Singleton() {}
private:
    static Singleton* _instance;
};

Singleton* Singleton::_instance = NULL;
```

Példa használatra:

```
int main()
{
    Singleton::Instance()->Print();
    ...
}
```

pont(1):

---

4. Jellemezze az előző pontban megadott megoldást, adja meg a megoldás kulcsgondolatait!

*Megoldás:*

A Singleton osztály az `_instance` statikus tagváltozóban tárolja az egyetlen példányra mutató pointert. Ennek kezdeti értéke NULL. Az egyetlen példányhoz hozzáférni az `Instance` statikus tagfüggvénnyel lehet. Első híváskor ez létrehozza az új objektumot, és eltárolja az `_instance` tagban. A későbbi hívások során már ezzel tér vissza. Az osztály konstruktora védett (`protected`), így garantált az, hogy kívülről, az `Instance` tagfüggvény megkerülésével ne lehessen további példányokat létrehozni. Az egyetlen példányhoz a globális hozzáférést az `Instance` statikus tagfüggvény biztosítja.

pont(1):

---

5. Mutasson példát ASP.NET inline script alkalmazására! A HTML részeket is adja meg!

*Megoldás:* Példa 5 faktoriálisának kiszámítására és az aktuális szerveridő kiírására:

```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN">
<html>
<head>
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    5! = <% int n = 1;
      for (int i = 1; i <= 5; i++)
        n = n*i;
      Response.Write(n.ToString());
    %>
  <br>
  Idő: <%=DateTime.Now.ToString() %>
  </div>
</form>
</body>
</html>
```

pont(1):

<b>AD</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(5):
-----------	---	----------

1. Mit garantálhat a relációs sémák normalizáláson alapuló sématervezési módszere?

- a) Az adatbázis konzisztenciáját.
- b) Egyes adatbázis-műveletek válaszidejét.
- c) Rekurziómentes adatbázis-lekérdezéseket.
- d) A relációk redundanciamentességét.
- e) A fizikai adatbázis egyszerű karbantarthatóságát.

Megoldás: b)

pont(1):

2. Melyik relációs sémafelbontásnak nincs gyakorlati jelentősége?

- a) Függőségőrző, veszteségmentes, részsémák nem BCNF-ek.
- b) Függőségőrző, nem veszteségmentes, részsémák BCNF-ek.
- c) Nem függőségőrző, veszteségmentes, részsémák BCNF-ek.

Megoldás: b)

pont(1):

3. Egy adatbázisban 16 millió rekord között kell a keresést támogatni a rekord egyediséget biztosító kulcsértéke alapján. Mekkora legyen legalább a vödörös hash vödörkatalógusának mérete (byte-ban), ha a kereséshez rendelkezésre álló idő 1 msec? A rekordok 250 byte-osak, egy blokk nettó mérete 4000 byte, minden kulcs 30 byte, a mutatók 4 byte-osak, az átlagos blokkelérési idő 1 msec. A vödörkatalógus befér a memóriába, a hash-függvény egyenletesen szór.

Megoldás: 4 millió byte

pont(1):

4. Vizsgálja meg a táblázatos teszttel az  $R(ABCDEF)$  séma  $R1(ABC)$ ,  $R2(CDE)$ ,  $R3(ABFD)$ ,  $R4(EAD)$  felbontását az

$$F = \{AB \rightarrow CDE, B \rightarrow FA, CD \rightarrow BE, E \rightarrow AD, EF \rightarrow A\}$$

függéshalmaz mellett!

Megoldás: veszteségmentes

pont(1):

5. Egy relációs adatbázis tervezéséhez mintaadatokat kaptunk az egyik táblából, ld. alább. A különböző kódok garantáltan különböző értékeket jelölnek.

Vizsgálja meg, hogy biztosan igazak-e az  $R(ABCD)$  sémán a funkcionális függőségekre vonatkozó következő állítások:

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c1	d2
a2	b2	c2	d1

$A \rightarrow B$  igaz — nem igaz

$B \not\rightarrow C$  igaz — nem igaz

$BC \rightarrow D$  igaz — nem igaz

pont(1):