VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból

2012. június 12.

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. Egy szoftver projektben az alábbi kockázatokat azonosítottuk. Elemezze a kockázatokat ! Tíznél kevesebb szót használjon! (4 pont)

	valószínűség	hatás
• A megrendelő késni fog a követelményeket is befolyásoló dokumentumok átadásával.	valószínű	uralható
• A fejlesztés elhúzódása közben az alkalmazásszerver egy lényegesen megújított változata jelenik meg. A betanulás időveszteséget okozhat.	valószínűtlen	elhanyagolható

Egy átlagos fejlesztés során 1 programozó hozzávetőlegesen mennyi idő alatt tud 1 funkciópontnyi (FP) programot elkészíteni ? (2 pont)

- ☐ 1 nap
- ☐ 1 hét
- ☐ 1 hónap
- ☐ 3 hónap

---

2. A dobozban adott egy Test osztály. Mit ír ki a mellette álló kódrészlet, ha az őt befoglaló kódról feltételezzük, hogy hibátlan? (4 pont)

```
FileOutputStream os =
      new FileOutputStream("test.sr");
ObjectOutputStream oos =
      new ObjectOutputStream(os);
Test t = new Test();
t.foo(1,2,3,4);
oos.writeObject(t);
t.foo(6,7,8,9);
oos.writeObject(t);
oos.close();
FileInputStream is =
      new FileInputStream("test.sr");
ObjectInputStream ois =
      new ObjectInputStream(is);
Test t2 = (Test)ois.readObject();
Test t3 = (Test)ois.readObject();
t2.bar();
t3.bar();
```

```
class Test implements
      java.io.Serializable {
public int a;
public static int b;
public transient int c;
private int d;
void foo (int i1, int i2,
      int i3, int i4){
      a= i1; b=i2; c=i3; d=i4;}
void bar() {
  System.out.println(a+""+b+""+c+""+d);}
}
```

1	7	0	4															
1	7	0	4															

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)

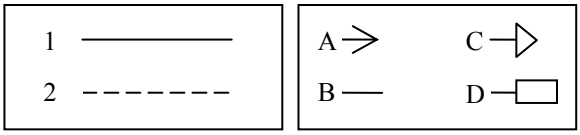
a még hiányzó alárendelt szoftver komponenst helyettesíti (pl. under construction weblap)

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 töltse ki a táblázatot. A rubrikákba azt kell beírni, hogy a két osztály közé 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	-

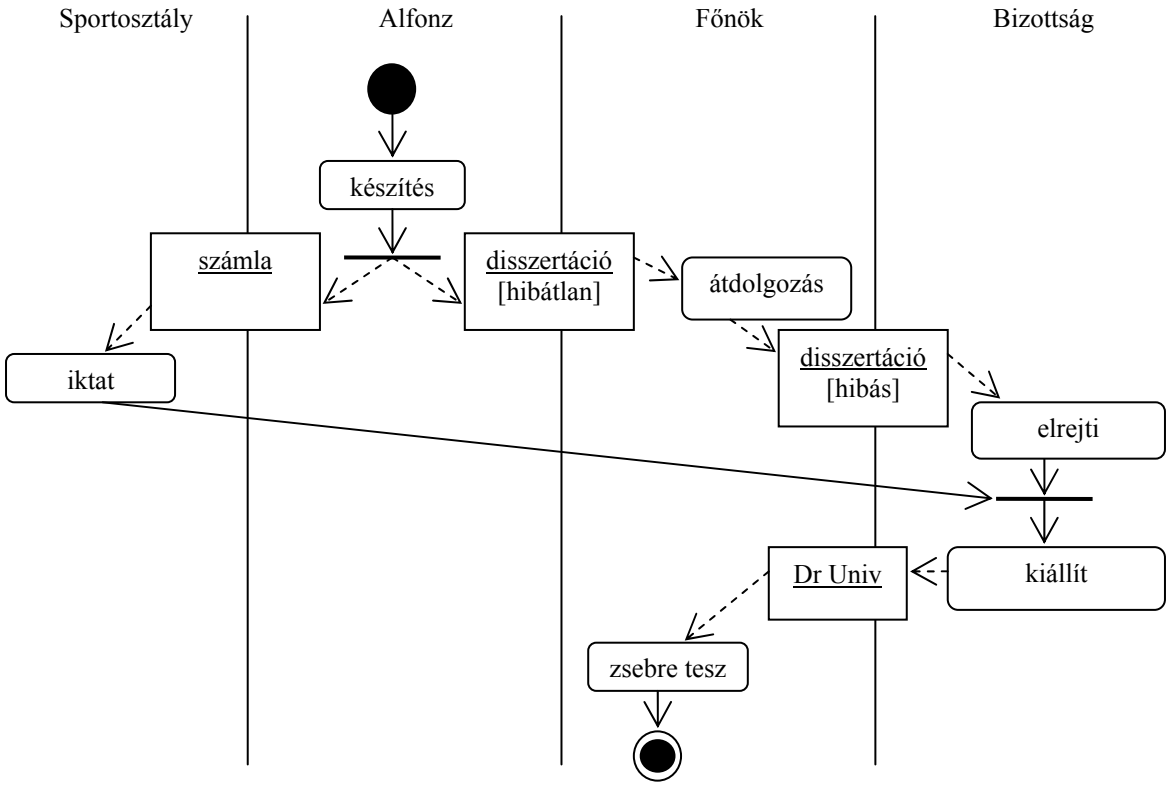
---

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 InteruptedException {   //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 **disszertáció**t **készít**. A hibátlan disszertációt elküldi a **főnök**ének **átdolgozás**ra, és ezzel párhuzamosan **számlá**t ad a **sportosztály**nak. 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 **bizonyítvány**t, 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