

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)

- I H *for* (*S x : z*) fejlécű *for* ciklusban a *z* referencia csak tömbre vagy a JDK-val szállított gyári kollekciók példányaira referálhat.
- I H egy szál egyszerre csak egy objektum monitorában tartózkodhat.
- I H *synchronized* kulcsszó használatával elkerüljük a deadlock kialakulását.
- I H szálak nem képesek saját magukat közvetlenül *waiting* állapotból *notify*-jal felébreszteni.
- 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 egy változó statikus típusa nem lehet a változó dinamikus típusának leszármazottja.
- I 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)

Tételezzük fel, hogy az A folyamat tartalmazza a B folyamatot.

Hogyan ábrázoljuk ezt az adatfolyam modellben és a use-case diagrammon? (2 pont)

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

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");  
  
        foo =  
                                (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)

Á

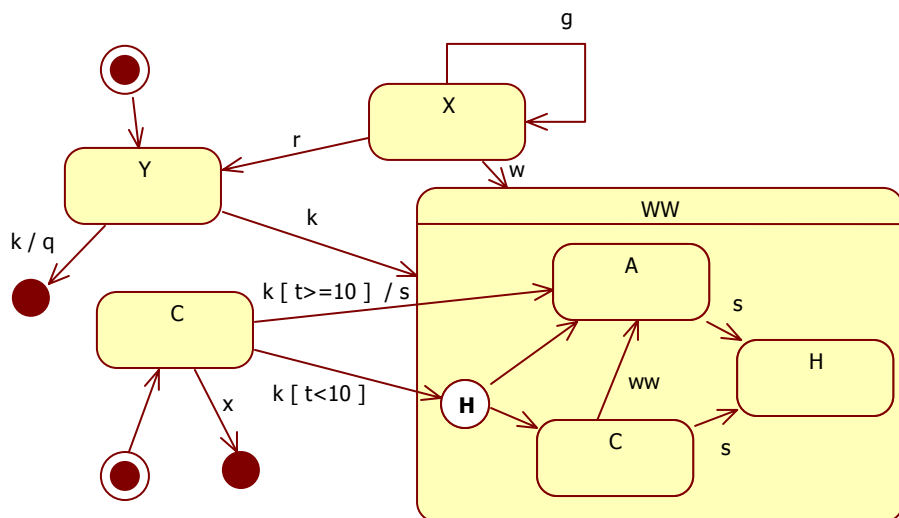
ÁÁ

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

6. Mik a konfigurációs menedzsment fő folyamatai? (4 pont)

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)



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

1.
2.
3.
4.
5.

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		
unique		
unique, ordered		
qualified		
qualified, ordered		

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