

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból

2015. június 2.

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

A feladatsor mentes az aktuálpolitikai témákra való utalásoktól és nem célja, hogy az államigazgatás bármely szervébe vetett közbizalmat megingassa, a tisztviselők iránti közmegebecsülést hátrányosan befolyásolja. A feladatsor szereplői nem valós személyek és bármilyen hasonlóság a valósággal csak a véletlen műve.

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 adatalem, egy állapotban nem fogadható el! (5 pont)

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

	a	b	c
1	2	4	-
2	3	-	-
3	2	-	1
4	-	3	-

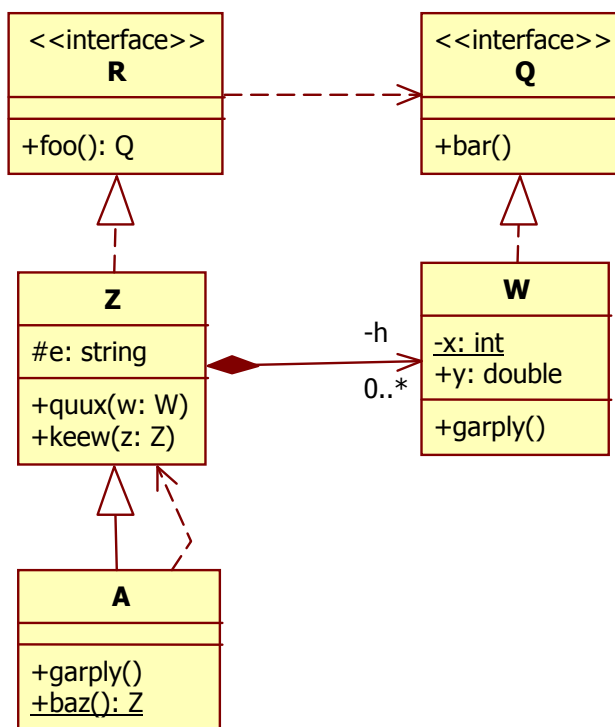
2. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (8 pont)

- I H** A t szál belépett x objektum szinkronizált foo metódusába. Az x foo metódusát semmilyen körülmények között nem hajthatja végre másik szál, amíg t vissza nem tért a metódusból.
- I H** Az A osztálynak leszármazottja B. B-nek minden olyan metódust meg kell valósítania, ami A-ban *abstract*.
- I H** *Abstract* osztálynak lehet *final* módosítójú metódusa.
- I H** Ha egy iterátor *hasNext()* metódusa *false*-szal tért vissza, az iterátor *remove()* metódusát meghívva minden esetben kivétel dobódik.
- I H** *Final* osztály csak akkor valósíthatja meg a *Serializable* interfészt, ha implementálja a *void write(InputStream is)* metódust.
- I H** A *String* osztálynak nincs olyan metódusa, amely az objektum tartalmát megváltoztatja.
- I H** Egy kollekció elemeit iterátorral érjük el. A kollekciót nem lehet úgy módosítani, hogy utána az iterátor a *hasNext()* metódushívásra ne dobjon kivételt.
- I H** A Java generikus osztályok példányosításakor sosem lehet primitív típus a template-paraméter.

3. A táblázatban felsorolt dokumentumok ellenőrzésére a V-modell milyen ellenőrzést (tesztelést) ír elő ? (3 pont)

Specifikáció	Rendszerteszt (System test)
Architektúra tervek	Integrációs teszt (Integration test)
Követelmények	Validálás (Validation)

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



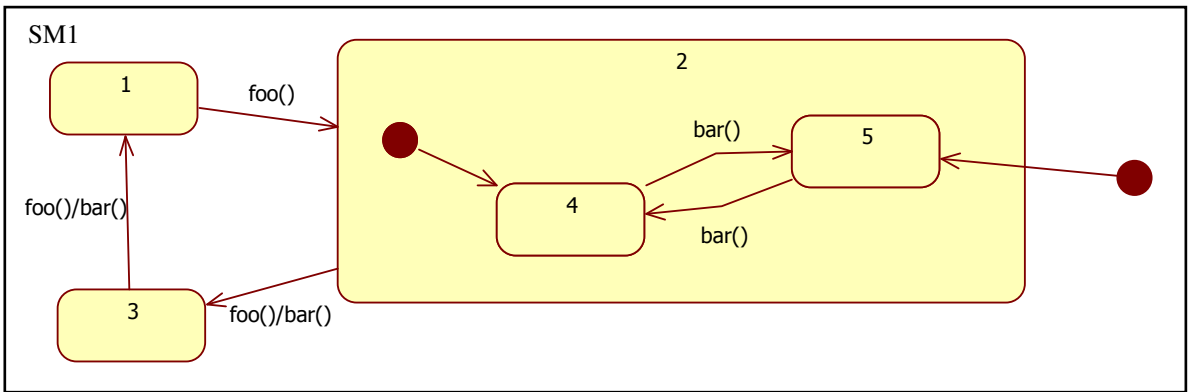
- A** - csak az első tagmondat igaz (+ -)
B - csak a második tagmondat igaz (- +)
C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
E - egyik tagmondat sem igaz (- -)

- [C]** Z foo() függvénye nem hozhat létre A objektumot, mert A nem implementálja a Q interfészt.
[C] Van olyan garply() függvény, amely nem módosíthatja W x attribútumát, mert x statikus.
[E] A baz() függvénye nem példányosíthat W objektumot, mert W absztrakt.
[B] Z keew(). függvénye nem kaphat paraméterül A objektumot, mert Z nem ismeri A-t.
[D] Z quux() függvénye módosíthatja a paraméterül kapott w y attribútumát, mert y publikus..
[B] A és Z interfésze megegyezik, mert mindkettő implementálja az R interfészt.
[B] Nincs olyan garply() függvény, amely módosíthatja az e attribútumot, mert e nem statikus.
[C] W bar() függvénye nem hívhatja meg A garply() függvényét, mert W nem ismeri Z-t.

5. Rajzoljon UML2 állapotgépet az alábbi Java osztályhoz! (6 pont)

```
public class SM1 {
    private static final int a[] = { 2, 3, 1 };
    private static final int b[] = { 5, 4 };
    private int s = 2;
    private int p = 5;

    public void foo() { s = a[s-1]; if (s == 2) p = 4; else bar(); }
    public void bar() { if (s == 2) { p = b[p-4]; } }
}
```



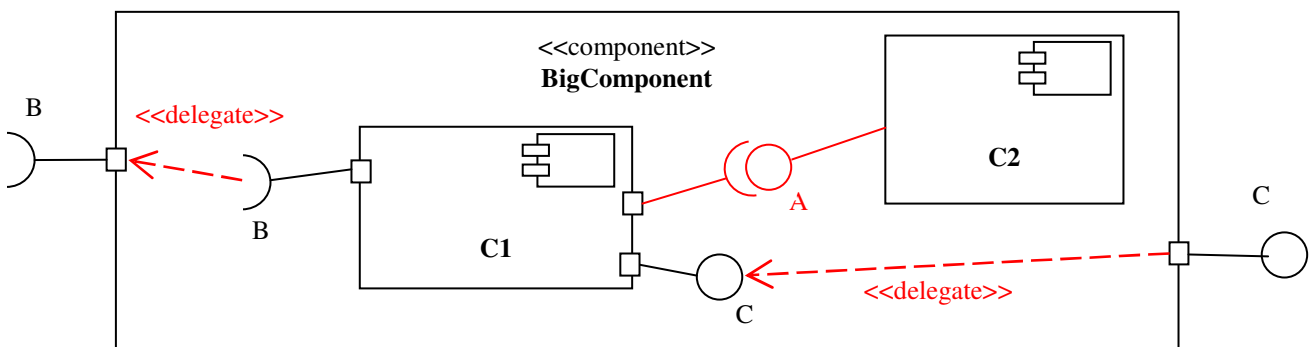
6. Mit jelent a CMM ? (2 pont)

Capability Maturity Model

Jelölje az állítások igazságtartalmát! (6 pont)

igen	nem	
<input type="checkbox"/>	<input type="checkbox"/>	a CMM egy adott szoftver termék fejlettségét, érettségét vizsgálja.
<input type="checkbox"/>	<input type="checkbox"/>	a CMM egy szervezet által készített összes szoftver minőségét értékeli.
<input type="checkbox"/>	<input type="checkbox"/>	a CMM szint kifejezi a szervezet vezetésének minőségét is.
<input type="checkbox"/>	<input type="checkbox"/>	a CMM egy szervezetben zajló szoftver fejlesztési folyamatot értékeli.
<input type="checkbox"/>	<input type="checkbox"/>	Koreográfia (choreography) elvén szervezik a technológiai folyamatokat.
<input type="checkbox"/>	<input type="checkbox"/>	Minden termék szabvány szerinti (pl. ISO 9126) megfelelőségét ellenőrzik.

7. Egészítse ki az alábbi UML2 diagramot a hiányzó részletekkel, figyelemmel arra, hogy C1 komponens használja a C2 komponens által megvalósított A interfészt! (3 pont)

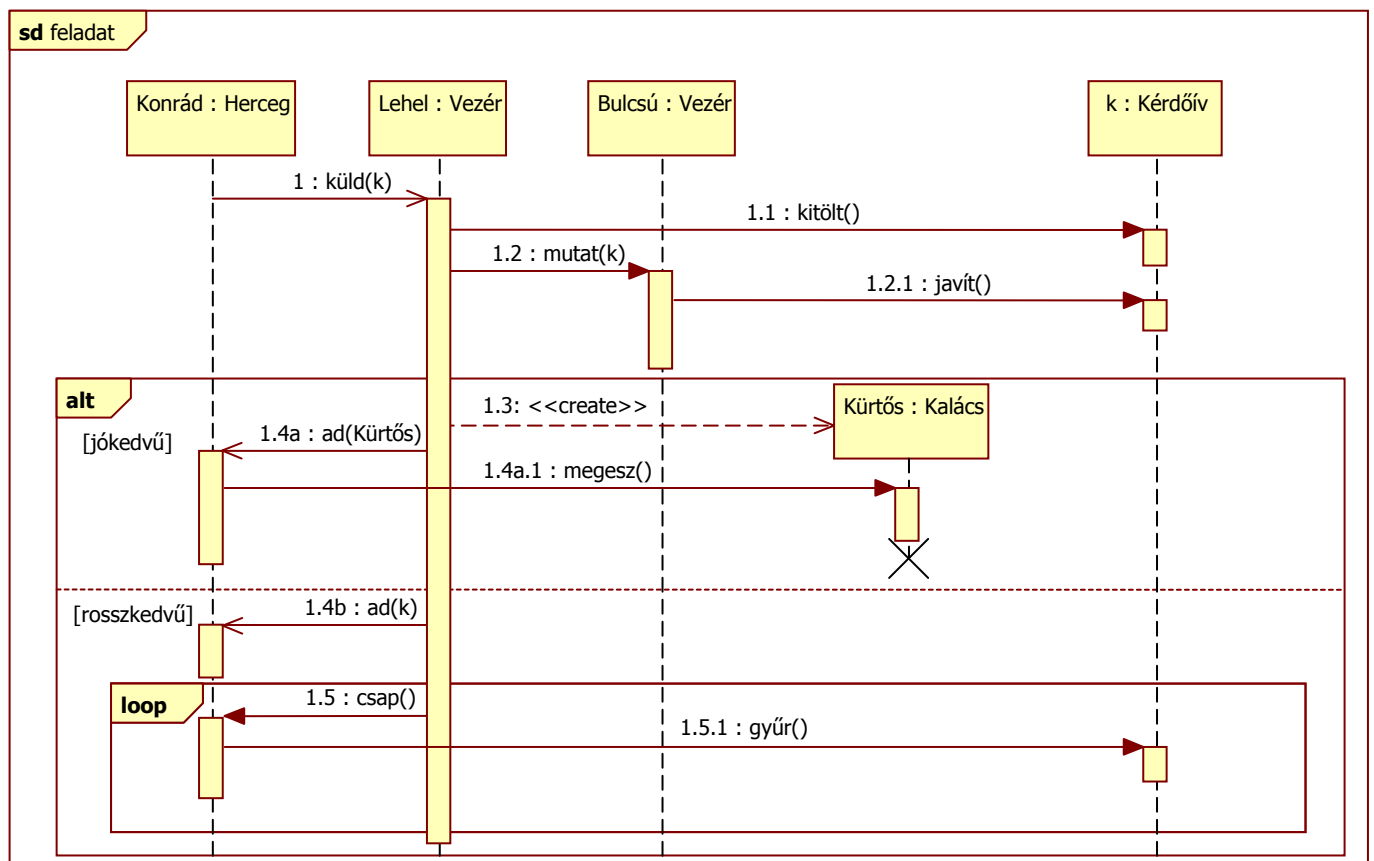


8. Az üzleti folyamatok lefutásának egyik esete az orkesztráció (orchestration). Mi a lényege ? (2 pont)

egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.

9. Rajzoljon UML2 szekvenciadiagramot az alábbi történet alapján! (7 pont)

Konrád herceg konzultációs kérdőívet küld Lehel vezérnek. Lehel kitölti, majd átnyújtja Bulcsúnak, hogy ellenőrizze a helyesírást, az eredményt pedig megvárja. Bulcsú két rovást kijavít. Ha Lehel jókedvű lesz, akkor süt egy kürtőskalácsot, és odaadja Konrádnak, aki a kalácsot megeszi. Ha Lehel rosszkedvű, akkor visszaadja Konrádnak a kérdőívet, majd Lehel többször is fejbecsapja a herceget, aki minden csapásnál gyűr egyet az íven.



Eredmények értékelése:

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