

VIZSGA FELADATSOR

VIII A217 SZOFTVERTECHNOLÓGIA

c. tárgyból

2016. január 12.

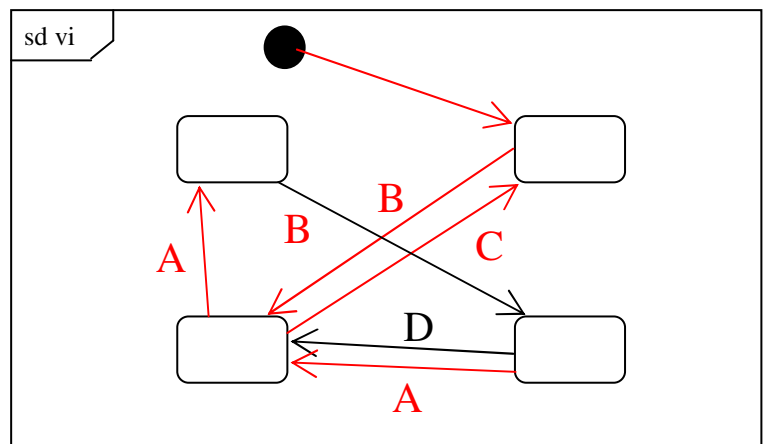
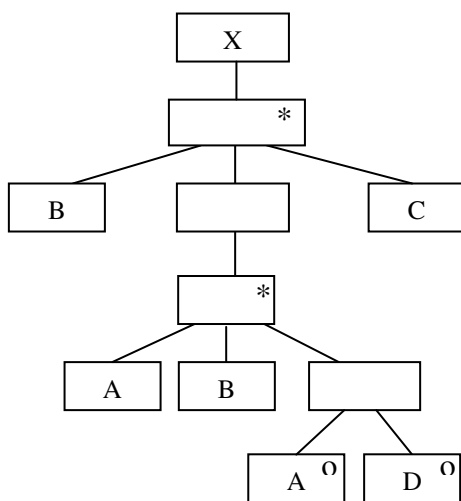
Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

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. Egészítse ki az UML2 állapotgépet úgy, hogy az ELH-nak megfelelő szerkezetet írja le! A kiegészítés során az **állapotgráfon új éleket vehet fel és az élekhez eseményeket írhat**. Bármilyen egyéb az ELH-n vagy az állapotgráfon változtatni tilos! CSAK a nyomtatott ábrába berajzoltakat értékeljük! (6 pont)

Javaslat: Készítsen vázlatot a harmadik lapon és CSAK a végső változatot rajzolja be a nyomtatott ábrába!



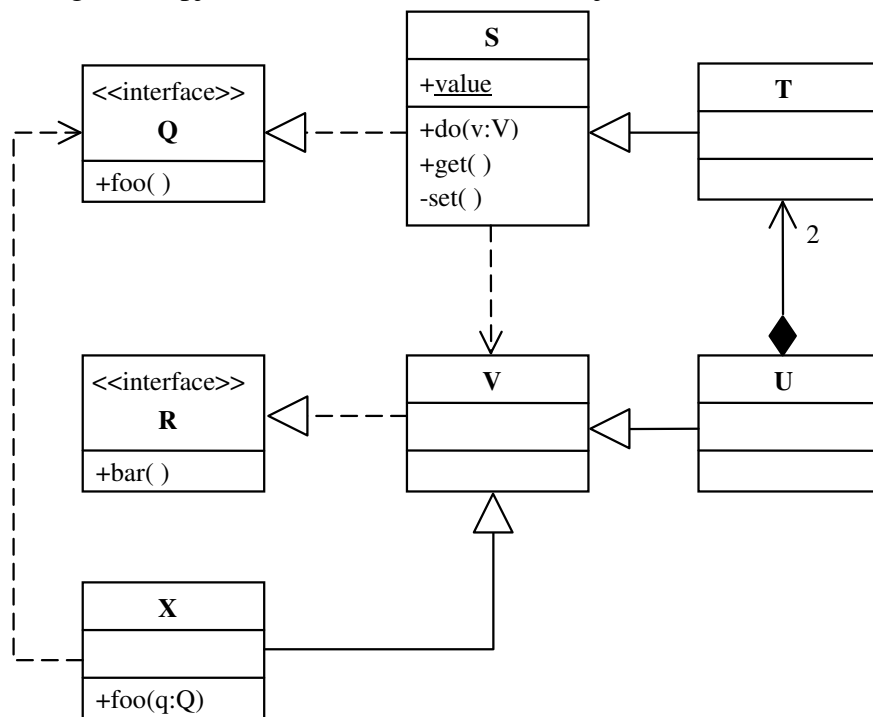
2. Az alábbiak közül mely deklaráció(k) írja(k) le helyesen a fenti ELH-t DTD-ben? (3 pont)

<input type="checkbox"/> $(B, A, B, A D^* C)^*$ <input type="checkbox"/> $(B, (A B (A, D))^*, C)^*$ <input type="checkbox"/> $(B (A, B, (A D)) C)^**$ <input type="checkbox"/> $(B (A, B, (A D))^* C)^*$ <input checked="" type="checkbox"/> $(B, (A, B, (A D))^*, C)^*$	<input type="checkbox"/> $(B, (A, B, (A D)^*), C)^*$ <input type="checkbox"/> $(B, (A B, (A D))^* C)^*$ <input type="checkbox"/> $(B, (A, B (A, D))^*, C)^*$ <input type="checkbox"/> $(B, ((A B), (A D))^*, C)^*$ <input type="checkbox"/> $(B, ((A B)^*, (A D)), C)^*$
--	--

3. Jelölje az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (7 pont)

I	H	Egy metódust el lehet látni egyszerre <i>abstract</i> és <i>final</i> módosítóval is.
I	H	A primitív típusokhoz tartozó csomagoló osztályok (wrapper classes) nem változtathatók (immutable).
I	H	A Java generikus osztályok példányosításakor sosem lehet primitív típus a template-paraméter.
I	H	Egy változó dinamikus típusa nem lehet absztrakt osztály.
I	H	Egy objektum <i>notifyAll()</i> metódusának meghívásakor a végrehajtó szál kilép az objektum monitorából.
I	H	Előfordulhat, hogy két szál (T1 és T2) ugyanazon objektum ugyanazon <i>synchronized</i> metódusát futtatva T1 T2 sorrendben lép be, de T2 T1 sorrendben lép ki.
I	H	Két interfész csak akkor valósítható meg egy osztályban, ha az interfészeknek nincsen közös metódusa.

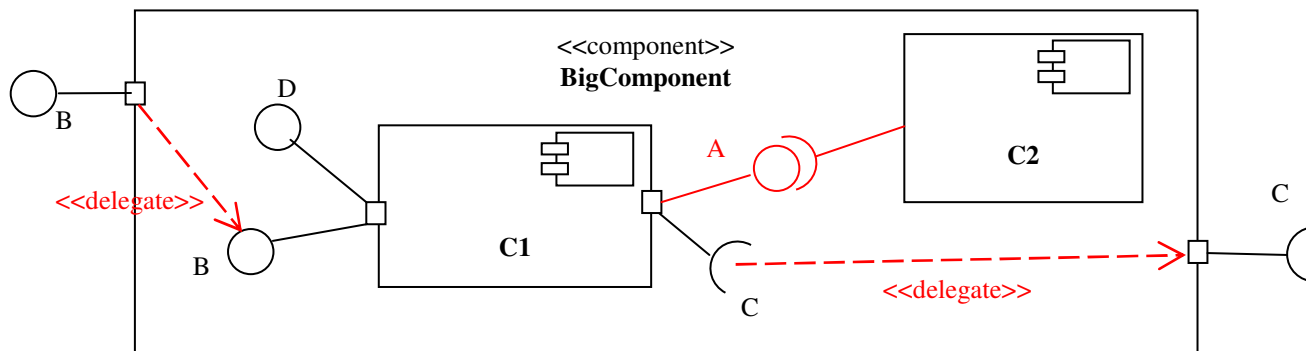
4. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|---|---------|
| 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] X foo(q:Q) metódusa kaphat paraméterül T-t, mert T-nek is van foo nevű metódusa.
- [B] S set() metódusa nem módosíthatja a value attribútumot, mert a láthatóságuk különböző.
- [B] V törlésekor törölni kell két T-t is, mert egy U-nak két T komponense van és U a V leszármazottja.
- [B] S meghívhatja T foo() metódusát, mert T az S leszármazottja.
- [A] X meghívhatja egy Q interfészű objektum foo() metódusát, mert X implementálja Q-t.
- [B] T létrehozhat U osztályú objektumot, mert S létrehozhat V-t és T az S-nek, U a V-nek leszármazottja.
- [B] X foo(q:Q) metódusa meghívhatja a paraméterül kapott S get() metódusát, mert S megvalósítja a Q interfészt.
- [B] X bar() metódusából meghívhatjuk egy Q interfészű objektum foo() metódusát, mert X foo(q:Q) metódusából is hívhatjuk egy Q interfészű objektum foo() metódusát.

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



6. 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. (4 pont)

Az alábbi táblázatba írja be, hogy a változásoktól függően mi történik commit esetén! Ha nincs változás, azt jelölje „---”, jellel!

		repository példány	
		változatlan	változott
munkap éldány	változatlan	---	---
	változott	publish	out of date error

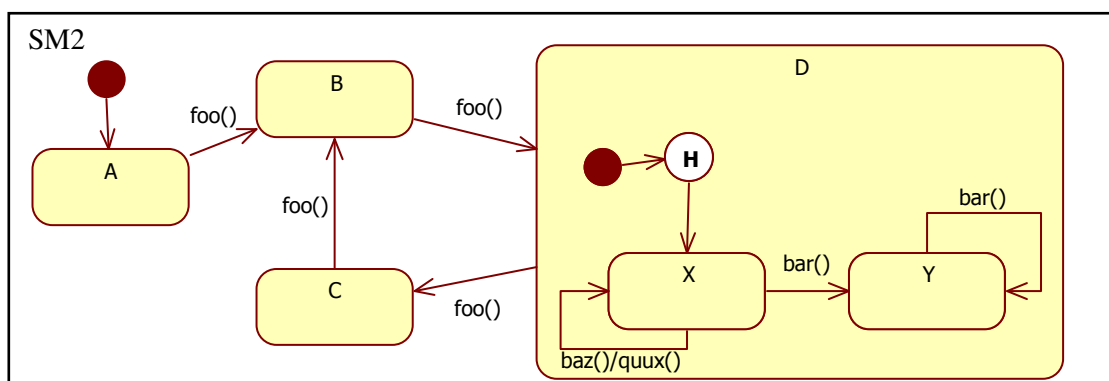
7. Rajzoljon UML2 állapotgépet az alábbi Java osztályhoz! (8 pont)

```
enum ES { A, B, C, D }
```

```
enum FS { X, Y }
```

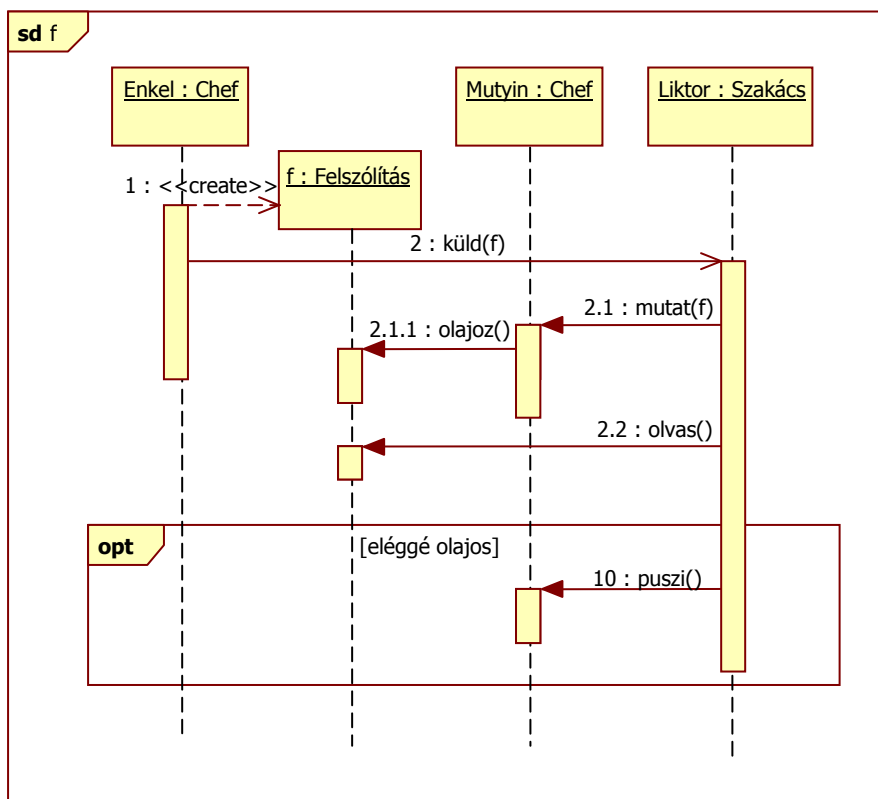
```
public class SM2 {
    private static final HashMap<ES, ES> e = new HashMap<>();
    private static final HashMap<FS, FS> f = new HashMap<>();
    static {
        e.put(ES.A, ES.B); e.put(ES.B, ES.D);
        e.put(ES.C, ES.B); e.put(ES.D, ES.C);
        f.put(FS.X, FS.Y); f.put(FS.Y, FS.Y);
    }
    private ES q = ES.A;
    private FS r = FS.X;

    public void foo() { q = e.get(q); }
    public void bar() { if (q == ES.D) r = f.get(r); }
    public void baz() { if (q == ES.D && r == FS.X) quux(); }
    public void quux() {}
}
```



8. Készítsen UML2 szekvenciadiagramot az alábbi történet alapján! (8 pont)

Zsarátnok városában két étterem üzemel. Az egyiket Margela Enkel, a másikat Gagyinír Mutyin vezeti. Enkelnek nem tetszik, hogy egyik szakácsa, Hombár Liktorkokettál Mutyinnal, ezért ír egy felszólítást Liktornak, majd elküldi neki. Liktorkokettát mutatja Mutyinnak, amit az azon nyomban megolajoz, majd Liktorkokettát átolvassa a felszólítást. Ha az eléggé olajos, akkor Mutyin orrára nyom egy pusztit.



9. 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

Eredmények értékelése:

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