

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

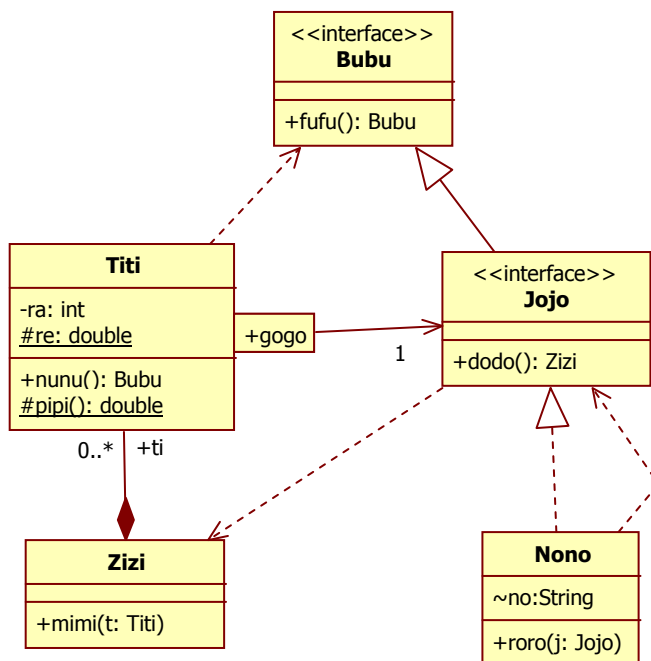
c. tárgyból

2014. január 21.

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

1. 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] Zizi mimi(t:Titi) függvénye nem hívhatja meg a t paraméter nunu() függvénye által visszaadott Jojo interfészű objektum dodo() függvényét, mert az sértené a Demeter-törvényt.
- [A] Titi pipi() függvénye nem szorozhatja össze a ra és re attribútum értékét, mert privát változóhoz csak privát függvény férhet hozzá.
- [B] Zizi a ti.nunu() hívás eredményén hívhat roro(j:Jojo) metódust, mert Nono implementálja a Bubu interfészt.
- [B] Egy Titi objektum csak egy Jojo interfészű objektumot ismer, mert közöttük lévő asszociáció Jojo oldalán 1-es számosság szerepel.
- [E] Nono roro(j:Jojo) függvénye kaphat paraméterül Bubu interfészű objektumot, mert Jojo implementálja a Bubu interfészt.
- [E] Nono dodo() függvénye nem hozhat létre Zizi objektumot, mert Nono nem ismeri Zizi-t.
- [B] Zizi függ Bubu-tól, mert Titi függ Bubu-tól.
- [B] Nono dodo() függvénye nem módosíthatja a no attribútum értékét, mert Javában a String típus immutábilis.

2. Az Agilis Kiáltvány ("Agile Manifesto") a klasszikus fejlesztés hagyományos fogalmait újakra cseréli. Mik az új fogalmak? (4 pont)

Klasszikus fejlesztés	Agilis fejlesztés
folyamatok és eszközök (processes and tools)	Individuals and interactions
precíz dokumentációk (comprehensive documentation)	Working software
szereződés szerinti tárgyalás (contract negotiation)	Customer collaboration
terv követése (following a plan)	Responding to change

3. Tekintsük a következő Java kódot! Jelölje be a táblázatban, hogy a fejlécben szereplő változók által mutatott objektumok a program mely pontján válhatnak először a garbage collector áldozatává! (3 pont)

<pre>class X { private String f = "fff"; public String g = "ggg"; public void gee() { System.out.println(f+g); this.g = null; } @Override public void finalize() { this.f = null; } }</pre>	<pre>public class Program { public static void main(String[] args) { X x1 = new X(); X x2 = new X(); x1.finalize(); /*A*/ x2.gee(); /*B*/ x2 = null; /*C*/ } /*D*/ }</pre>
---	--

	x1	x1.f	x1.g	x2	x2.f	x2.g
/*A*/	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
/*B*/	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
/*C*/	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
/*D*/	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

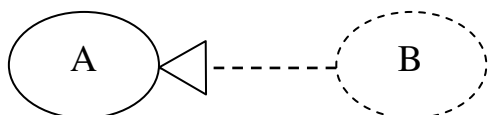
4. Adott a következő BNF specifikáció.

```
szam ::= 0|1|2|3|4|5|6|7|8|9
betu ::= A|B|C|D|E|F
cwrgl ::= <szam>|<szam><cwrgl>
qyfl ::= <betu>|<betu><qyfl>
pof ::= <cwrgl>[<qyfl>]
laax ::= <qyfl><cwrgl>
vizsga ::= <pof>|<laax>
```

Adja meg, hogy a következő kifejezések megfelelnek-e a vizsga specifikációnak! (5 pont)

Igen	Nem		Igen	Nem	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	3AF44B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	06209552783
<input checked="" type="checkbox"/>	<input type="checkbox"/>	BBDA28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	9357F
<input checked="" type="checkbox"/>	<input type="checkbox"/>	7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	CAA
<input type="checkbox"/>	<input checked="" type="checkbox"/>	D	<input type="checkbox"/>	<input checked="" type="checkbox"/>	DACCC84E
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ABCBA125	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2014EEC021BB

5. Mi **A** és **B** az alábbi UML diagramon? (4 pont)



A		B	
<input type="checkbox"/>	operáció	<input checked="" type="checkbox"/>	kollaboráció
<input type="checkbox"/>	állapot	<input type="checkbox"/>	metódus
<input checked="" type="checkbox"/>	use-case	<input type="checkbox"/>	beágyazott állapot
<input type="checkbox"/>	processz	<input type="checkbox"/>	feltételes use-case

6. Rajzoljon UML2 állapotdiagramot a W osztályhoz! (7 pont)

```

enum XS { A, B }
enum YS { P, Q }
enum ZS { M, N }

class W {
  private XS xs = XS.A;
  private YS ys = YS.P;
  private ZS zs = ZS.M;

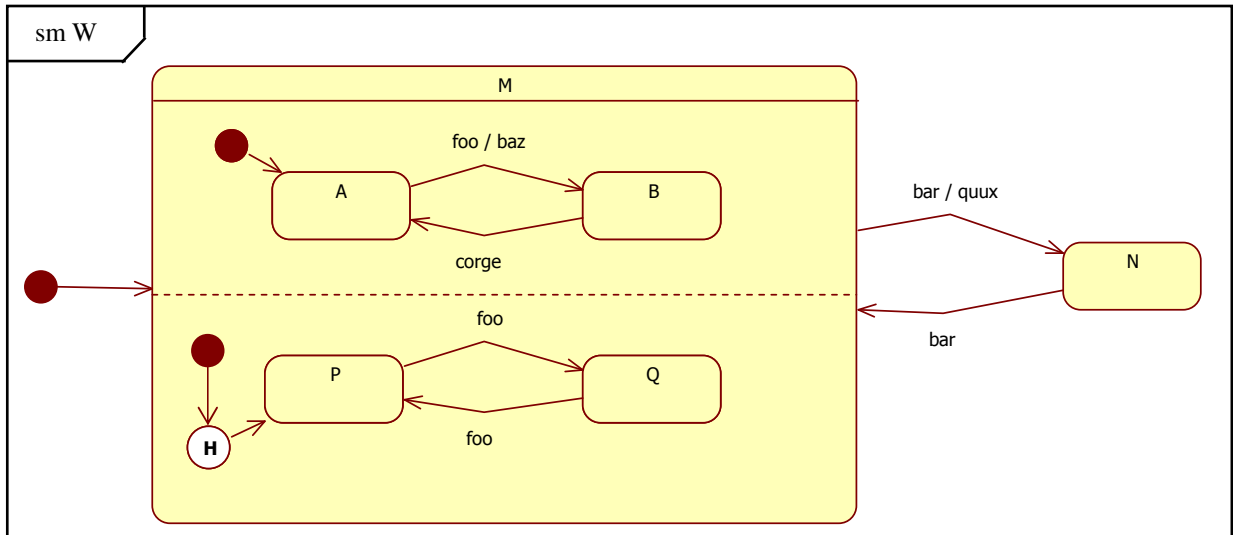
  public void foo() {
    if (zs == ZS.M) {
      if (xs == XS.A) {
        xs = XS.B;
        baz();
      }
      if (ys == YS.P) ys = YS.Q;
      else ys = YS.P;
    }
  }

  public void bar() {
    if (zs == ZS.M) {
      zs = ZS.N;
      quux();
    } else {
      zs = ZS.M;
      xs = XS.A;
    }
  }

  public void corge() {
    if (zs == ZS.M) {
      if (xs == XS.B) xs = XS.A;
    }
  }

  private void baz() { }
  private void quux() { }
}

```



7. Az UML2-ben a gyűjteményeknek (kollekcióknak) két fontos tulajdonsága van: rendezettség (ordered) és egyediség (unique). Írja be a táblázatba az UML2 kollekciók nevét ! (4 pont)

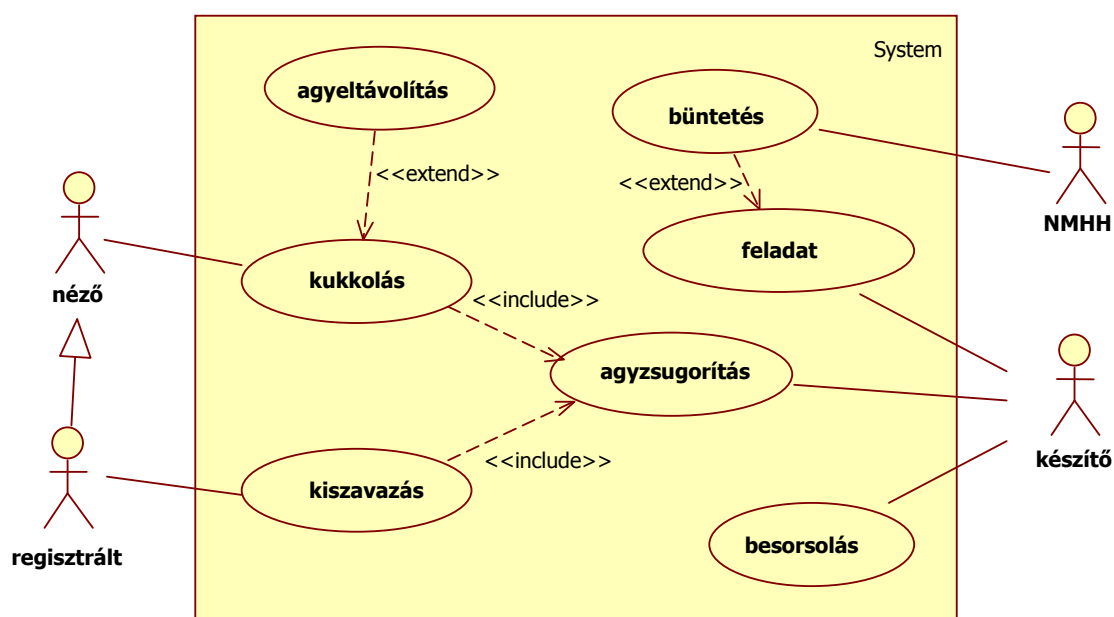
rendezett	egyedi	UML2 kollekció neve
igen	igen	Ordered Set
igen	nem	Sequence
nem	nem	Bag
nem	igen	Set

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

- I H** minden primitív típusnak van csomagoló (wrapper) osztálya.
- I H** primitív típus is lehet generikus osztály template-paramétere.
- I H** ha egy szál véget ért, nem lehet újraindítani.
- I H** szálakat a *run()* metódus meghívásával indíthatunk.
- I H** szerializálás körkörös hivatkozású adatszerkezeten (pl. gyűrű) kivételt dob.
- I H** minden objektumnak van *wait()* metódusa.
- I H** nincs olyan várakozó szál, amelyik egyből RUNNABLE állapotú lesz a *notifyAll()* hatására.
- I H** szálakon a *join()* metódust csak az indításuk sorrendjében szabad meghívni.

9. Az alábbi történet alapján készítsen UML2 használati eset (use-case) diagramot!! (7 pont)

A HelóVilág c. tévéműsor folyamában a nézők kukkolhatnak. Ez egyes esetekben agyeltávolításban végződik. A regisztrált nézők a kukkolás mellett kiszavazásban is részt vehetnek. Mind a kiszavazás, mind a kukkolás során a rendszer végrehajtja az agyzsugorítást. A rendszert a készítő is használják: egyfelől besorsolhatnak idiotákat, másfelől feladatokat adhatnak meg, harmadrészt pedig aktív résztvevői az agyzsugorításnak. Ha a feladat túl rücskösre sikerül, akkor büntetéssel jár, ennek mértékét az NMHH határozza meg.



Eredmények értékelése:

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