

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból
2012. június 5.

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! (5 pont)

- I** **H** Statikus attribútumot csak statikus metódusból lehet elérni.
- I** **H** Példánymetódust kivétel nélkül csak példánymetódusból hívhatunk.
- I** **H** Egy osztály lehet akkor is absztrakt, ha nincs absztrakt metódusa.
- I** **H** A *java.lang.String* osztálynak vannak a string értékét (tartalmát) megváltoztató metódusai.
- I** **H** A primitív típusokhoz tartozó csomagoló osztályok (wrapper classes) nem változtathatók (immutable)

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total ≥ 0

2. Adja meg, hogy az alábbi állítások melyik szabványos Java kivételosztályra igazak! Ha többre is igaz, akkor az öröklési hierarchiában legmagasabban levő (ős)osztályt jelölje meg! (6 pont)

A	Throwable	B	Exception	C	InterruptedException
D	RuntimeException	E	NullPointerException	F	Error

- [A]** Minden kivételként eldobható osztály ősoosztálya.
- [F]** JVM szintű rendszerhiba
- [F]** Elkapása nem kötelező, nehéz rá felkészülni, nehéz lekezelni.
- [B]** Programhiba
- [D]** Elkapása nem kötelező, "zajt" vinne a kódba.
- [A]** Az *initCause* metódusával be lehet állítani az okozó kivételt

Blank 0 pont, minden találat 1 pont, minden rossz válasz -1 pont, de total ≥ 0

3. Minek az ellenőrzésére irányulnak az alábbi tesztelések? (4 pont)

Rendszerteszt (System test)	Specifikáció
Validálás (Validation)	Követelmények

4. A `Java.util.Stack<E>` osztály metódusai közötti kohézió milyen típusú? (2 pont)

Kommunikációs (ugyanazon adatszerkezeten operálnak)

5. Mi a szerepe a JUnit-ban az alábbi annotációknak? (2 pont)

@BeforeClass	Tesztosztály közös inicializálása – erőforrások lefoglalása
@After	Egyedi tesztet lefutása után erőforrások elengedése

Egy JUnit teszt eredménye *pass* vagy *fail* vagy *error* lehet. A *pass* a tesztben foglalt állítás beteljesülését jelenti. Mi a jelentése a másik két eredménynek? (2 pont)

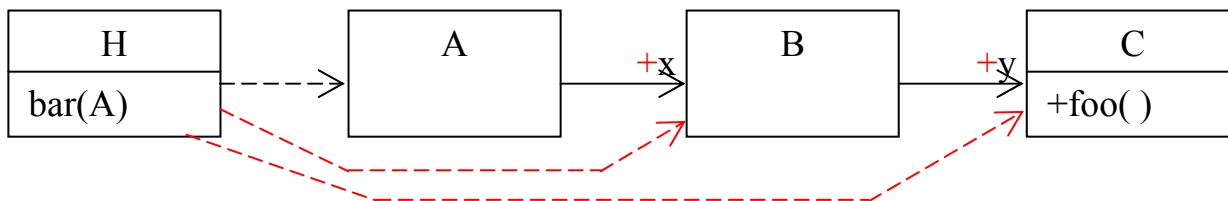
fail	Az állítás nem teljesül.
error	A tesztmetódus általa nem kezelt kivételt dob.

6. Az UML2 Activity diagram egy másik UML2 diagram speciális esetének tekinthető. Melyik ez a diagram? Hasonlítsa össze a két diagramot! (3 pont)

Activity diagram	Állapot..... diagram
state – tevékenységet hajt végre	state (eseményre vár)
transition – a tevékenység befejezésekor (belső) automatikusan	transition (külső esemény hatására)

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

7. Adott az alábbi UML2 diagram, ahol a H osztály *bar(A)* metódusát a következőképp implementáljuk:
`public void bar(A a) {a.x.y.foo();}`

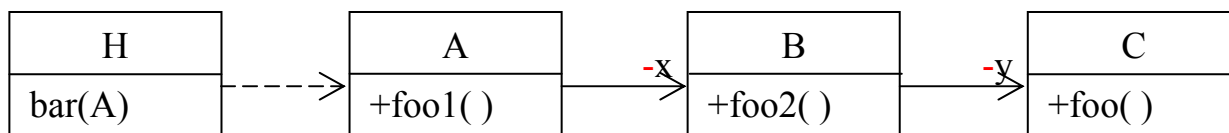


A *bar(A)* metódus implementációjára figyelemmel egészítse ki a diagramot ! (2 pont)

Milyen tervezési problémákat ismer fel? (2 pont)

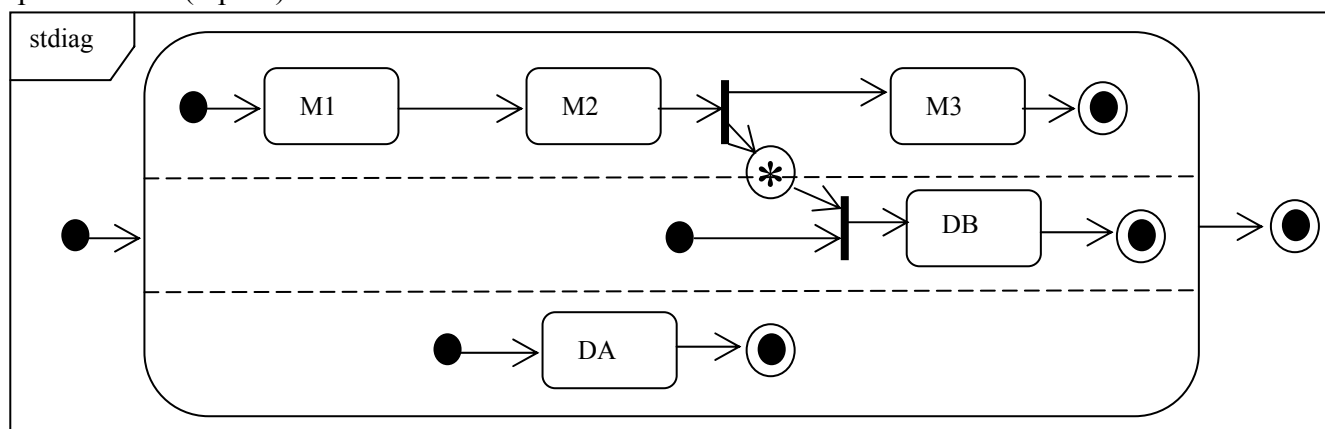
Ellentétes a Demeter-törvénnyel (H össze van csatolva mindegyik osztállyal, A, B gyengén kohézív.)

Rajzolja fel a jól tervezett változat UML2 diagramját, és adja meg a metódus(ok) implementációját! (6 pont)

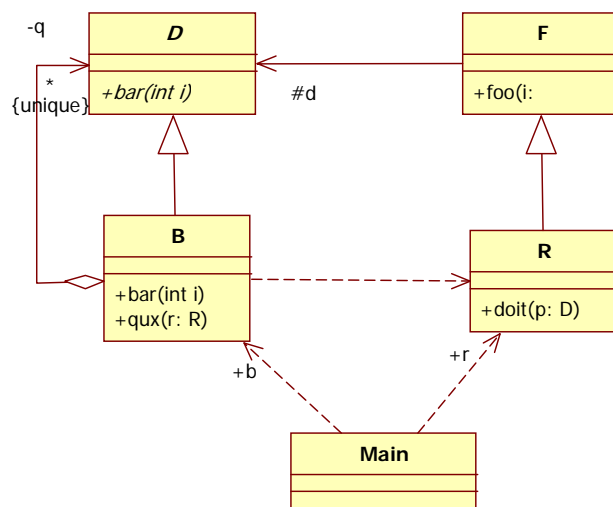
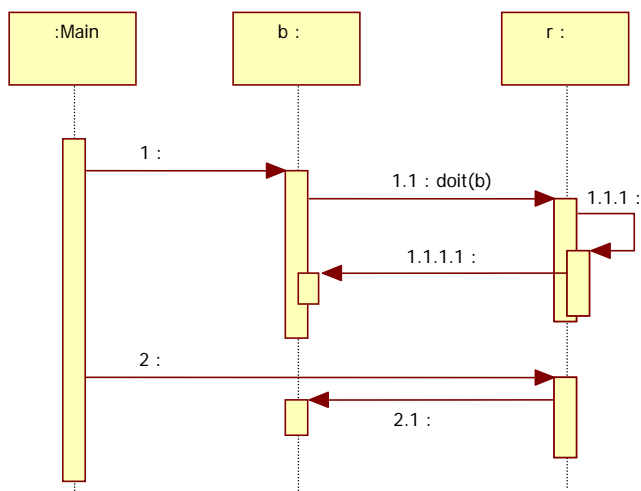


```
public void bar(A a) {a.foo1();}
public void foo1() {x.foo2();}
public void foo2() {y.foo();}
```

8. Egy tantárgy teljesítéséhez három mérést (M1, M2, M3) kell elvégezni, és két házi dolgozatot kell beadni (DA, DB). A méréseket szigorúan szám szerinti sorrendben kell elvégezni, a két dolgozat tetszőleges sorrendben készíthető el. A DB dolgozat csak az M2 mérés elvégzését követően készíthető el. Rajzoljon UML2 állapot modellt! (8 pont)



9. Java nyelven implementálja az alábbi osztály és szekvenciadiagramon megtervezett osztályokat és metódusokat! A Main osztály működése a main metódusban valósul meg, ami a futása elején létrehoz egy B és egy R objektumot (ez az ábrán helyhiány miatt nincs jelölve). (8 pont)



```

abstract class D {
    public abstract void bar(int i);
}
class B extends D {
    private Set<D> q;
    public void bar(int i) {...}
    public void qux(R r) {
        r.doit(this);
    }
}
class F {
    protected D d;
    public void foo(int i) {
        d.bar(i);
    }
}
class R extends F {
    public void doit(D p) {
        d = p;
        foo(3);
    }
}
public class Main {
    public static void main(String[] args) {
        B b = new B();
        R r = new R();
        b.qux(r);
        r.foo(7);
    }
}
  
```

Eredmények értékelése:

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