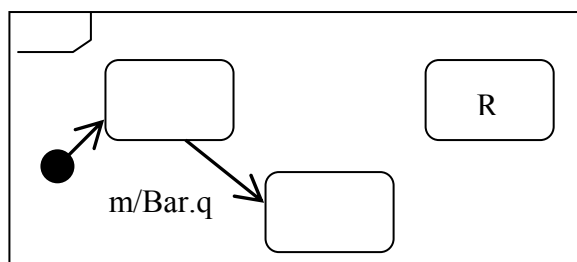
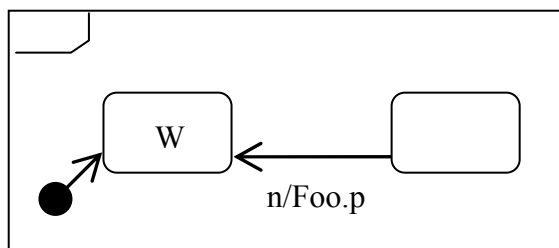
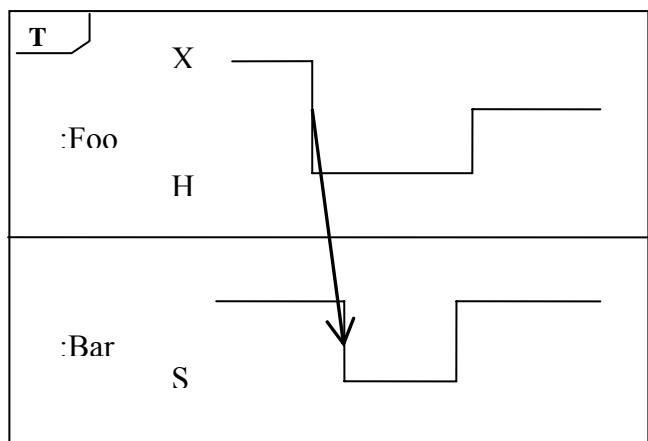


# VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból  
2012. január 17.

**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. Izidor elkészített három – szemantikailag összefüggő, hibátlan – UML2 diagramot. Miközben vacsorázott, a kis huncut Zsilip (Izidor öccse) kiradírozott néhány elemet (szöveg, vonal, nyíl stb.) Segítségül Izidornak visszaállítani az ábrákat! Rajzold be a hiányzó elemeket és feliratokat, úgy, hogy az ábrák helyesek és összefüggőek legyenek! (5 pont)



2. Sorrendben adja meg, hogy a *foo* metódus teljes lefutása során a futtató szál milyen állapotokat vesz fel! (Segítségül megadtuk az induló állapotot.) Tételjeze fel, hogy a metódus meghívásakor egy másik szál ugyanazon objektum *bar* metódusát hajtja végre! (5 pont).

```
synchronized void bar() {
    ...
}
synchronized public void foo() {
    try {
        this.wait();
    } catch (Exception e) { }
}
```

1	RUNNABLE (RUNNING)
2	
3	
4	
5	
6	
7	

3. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (4 pont)

I H Egy változó dinamikus típusának nem lehet *abstract* metódusa.

I H Egy változó statikus típusának nem lehet *final* metódusa.

I H Egy *interface* típusnak csak egyetlen közvetlen őse lehet.

I H Az alábbi kódrészlet futtatása végén a *b* változó értéke *true*.

```
String s1 = new String("Hello");
String s2 = s1;
s2 += " world";
boolean b = (s1 == s2);
```

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

4. Adott az alábbi dekorált XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE z [
  <!ELEMENT z (f+,x)>
  <!ELEMENT f ANY>
  <!ELEMENT x (#PCDATA)>
  <!ELEMENT k (#PCDATA ① z)*>
]>
<z>
  <f><z><f>
    ②
    <x></x></z>
    f/</f><x>/x</x>
</z>
```

Mit írna ① helyére, hogy a DTD jól formált legyen? (2 pont)

- +
- \*
- |
- ”,” (vessző)

Feltételezve, hogy a DTD jól formált, mi állhat ② helyében, hogy az XML érvényes legyen? (4 pont)

- <x><k>/x</k></f>x</f>
- <z><k></x></f>x</k></z>
- <x><k></x></f>x</k></f>
- <x>x/x</x></f>x</f>
- <k><x></f>x</k></z>
- <f><k>x</k></f>x</f>
- <f><x>f</x>/x</f></f>
- <z><k>x</k></z>

5. A mellékelt állapotgéppel specifikált programot az interpreter architektúra szerint kívánjuk megvalósítani. Az állapottábla alapján specifikálja a

	A	B	C
S1	S1/X	S1/X	S2/Y
S2		S2/Y	S3/Y
S3	S1/X	S3/Y	

“control state” értékeit! (1 pont): .....

“pseudocode” utasításait! (3 pont): .....

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

6. A kulcs felhasználásával jellemezze az alábbi Java kódrészlettel kapcsolatos állításokat ! (4 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 (- -)

```
public class X {
    private long l;
    private static void foo() { System.out.println("hello"); bar(); }
    public void bar() { System.out.println("world"); }
    public void bar(X x) { System.out.println("world"); x.l=10; }
    protected X copy() { return new X(); }
    public void do() { bar(copy()); }
}
```

- A *foo()* metódusban hibás a *bar()* hívása, mert privát láthatóságú metódusból nem hívhatunk publikusat.
- A *do* metódus fordításkor hibát jelez, mert publikus metódus paraméterének nem lehet protected metódus visszatérési értékét adni.
- X* osztály nem példányosítható, mert nincs paraméter nélküli konstruktora.
- X* csak egyetlen másik osztálynak lehet a közvetlen őse, mert a Javában nincs többszörös öröklés.

Blank 0 pont, minden találat 1 pont, minden rossz válasz -0.5 pont, de total >= 0

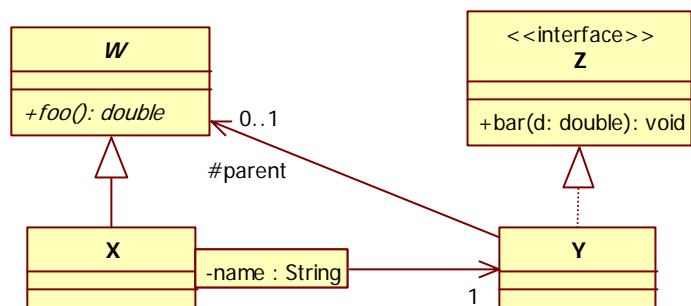
7. Jellemezzünk egy (K,V) párokból álló Map-et – ahol K egy kulcs, V egy 0-nál nagyobb egész érték – az alábbi műveletekkel! Adja meg az algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveletet. (6 pont)

- NEW()** új (üres) map-et hoz létre.
- PUT(m,(k,v))** az *m* map-hez hozzáveszi a *k,v* párt, ha *k* nem szerepelt a map-ben. Ha a map-ben már van *k*,akkor a hozzá tartozó *v*-t az új *v*-vel helyettesíti, ha az kisebb a map-ben szereplőnél.
- GET(m,k)** megadja az *m* map-ben a *k* kulcshoz tartozó *v*-t. Ha a map-ben a megadott *k* nem szerepel, akkor az eredmény 0.
- MIN(m)** az *m* map-ben szereplő legkisebb *v*. (üres map esetén nulla)

8. Adott a következő Scrum product backlog (teendők listája). Feltételezve, hogy a munka pontosan a terveknek megfelelően haladt, rajzolja fel a végső burndown chart-ot (napi eredménykimutatás)! (6 pont)

	H	K	Sz	Cs	P
brummogás	5	4	5	1	
tekerés			11	2	6
számolás	3	6	4	4	
okoskodás	2			3	4

9. Izidor karácsonyra kapta az alábbi szép UML2 osztálydiagramot, azzal, hogy írja meg az ábrán látható osztályok és interfészek Java forráskódját. Izidor még nem készült fel a Szofttech vizsgára, ezért nem megy neki a dolog. Segíts neki, és készítsd el te a kódot! Minden, az ábrából következő módszert és attribútumot vegyél fel! Ha szükséges, használd a szabványos Java API osztályait is! Minden osztály publikus. A metódusok törzse legyen üres! (10 pont)



Eredmények értékelése:

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