

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból

2012. december 18.

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. Adott az alábbi (hibás) Java kódrészlet.

```
public class Valami extends Exception implements Serializable {
    private transient int foo;
    public boolean bar;
    protected static String baz;
    public Valami(String s) { baz = s; }
}

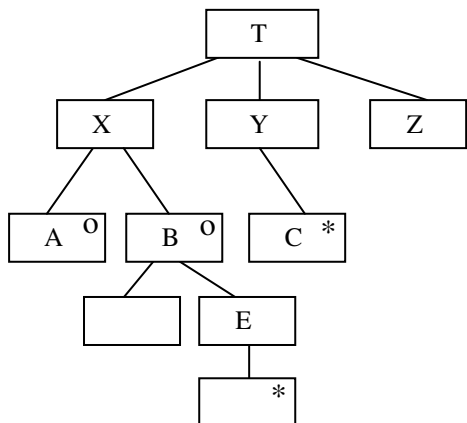
public class Main {
    public void serialize(String file, Set<Valami> sv) {
        try {
            FileOutputStream os = new FileOutputStream(file);
            ObjectOutputStream oos = new ObjectOutputStream(os);
            oos.writeObject(sv);
            os.close();
        } catch (Exception e) { e.printStackTrace(); }
    }
}
```

Jellemezze az alábbi állításokat a kulcs felhasználásával! (8 pont)

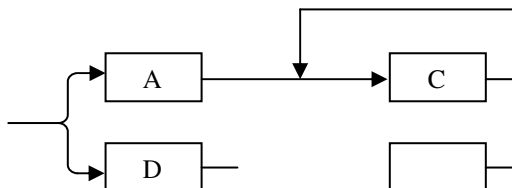
- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

- A **Main** osztály nem tud sorosítani, mert a **Main** osztály nem implementálja a **Serializable** interfészt.
- A **serialize** metódus **sv** paramétere nem sorosítható, mert interfész típusú változót nem lehet sorosítani.
- A **Valami** konstruktorában a **baz** attribútumnak egyenlőségjellel nem adható értékül az **s** paraméter, mert csak a következő forma lenne helyes: **baz = new String(s)**
- Egy **Valami** típusú objektum **foo** attribútuma nem kerül sorosításra, mert **private** láthatóságú attribútum nem sorosítható.
- A **Main** osztály **serialize** metódusa tartalmaz hibát, mert hiányzik belőle egy **oos.close()** hívás.
- A **Valami** osztály egy példánya nem sorosítható a **writeObject** metódussal, mert **Valami** az **Exception** leszármazottja.
- protected** módosítóval ellátott attribútum nem sorosítható, ezért a **Valami** osztályban definiált **baz** attribútum nem kerül sorosításra.
- A **FileOutputStream** konstruktorhívása hibás, mert nem **File** típusú paramétert kap.

2. Egészítse ki az alábbi ELH-t és állapottáblát úgy, hogy mindkettő ugyanazt a szerkezetet írja le! A kiegészítés során az **ábrák szerkezetén**, az azokba **beírt, berajzolt elemeken változtani tilos!** Az ELH-ban csak a két üresen maradt blokkot kell kitölteni! Az állapottáblát egészítse ki! Az induló állapot legyen az ① ! (ELH 2 pont, állapottábla 4 pont)



	A	C	D	
①			②	
②		③	②	④
③				
④				



A szintaxis gráfot egészítse ki úgy, hogy az is a fenti szerkezet írja le! (3 pont)

3. Az alábbiak közül mely deklarációk szerepelhetnek a 2. példában adott szerkezetet leíró DTD-ben? (2 pont)

- | | | | | | |
|--------------------------|---|-----------------|--------------------------|---|---------|
| <input type="checkbox"/> | T | (X+Y+Z) | <input type="checkbox"/> | Y | (C+) |
| <input type="checkbox"/> | T | ((A D*), C*, Z) | <input type="checkbox"/> | D | (E+) |
| <input type="checkbox"/> | X | (A+B) | <input type="checkbox"/> | Y | (C*) |
| <input type="checkbox"/> | X | (A B) | <input type="checkbox"/> | B | (D, E*) |
| <input type="checkbox"/> | T | (X, Y, Z) | <input type="checkbox"/> | B | (D, E) |
| <input type="checkbox"/> | T | ((A B), C+, Z) | <input type="checkbox"/> | B | (D+) |
| <input type="checkbox"/> | T | ((A B), C*, Z) | <input type="checkbox"/> | C | (Y*) |
| <input type="checkbox"/> | X | (A (D, E D*)) | | | |

4. Legyen az alábbi A osztályunk.

```
class A {
    protected int j;
    public int foo(int i) {return(j);}
}
```

Legyen egy B osztály, amely A-ból származik, és metódusai az alábbi táblázatban találhatóak. Jelölje meg az(oka)t a metódus(oka)t, amely(ek) megsérti(k) a Meyer-féle nyit-zár (open-close) elvet! (2 pont)

- public int foo1(int i) {return(j);}
- public double foo(double d, int i) {return(d*j);}
- public int foo(int i) {return(j*i);}
- public int foo(double d) {return((int)(j*d));}
- public double foo2(int i) {return(3.0*i);}

5. Sorolja fel a Scrum agilis módszertan legfontosabb dokumentumait, anyagait (artifacts) (3 pont)

.....

.....

.....

.....

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

6. Az alábbi Java kódrészletek alapján készítsen UML 2 szekvenciadiagramot! A diagramot a *START*-tal jelölt megjegyzéstől kezdje! Tételezze fel, hogy a D.xxx() metódus aszinkron! (10 pont)

```
public interface X {
    void bar();
}
public class A {
    static protected double qux(double d){
        return 2*d;
    }
    public void foo(B b) {
        C c = b.get();
        c.bar();
    }
}
public class B {
    C c;
    public void set(C x) {
        c = x;
        c.bar();
    }
    public C get() {
        c.hello().xxx();
        return c;
    }
}
```

```
public class C {
    public void bar() {
    }
    public X hello() {
        bar();
        return new D();
    }
}
public class D implements X {
    public void xxx() {}
}
public class Main {
    public static void main(String args[]) {
        A a = new A();
        B b = new B();
        C c = new C();
        b.set(c);
        // START
        a.foo(b);
        // END
    }
}
```

7. Jellemezzünk egy stringet az alábbi műveletekkel!

CRT()	új (üres) stringet hoz létre
LGTH(s)	az s string karaktereinek számát adja
TAIL(s)	az s string első karakterének levágása után maradó stringet adja
XTEND(s,x)	az s string végére rakja az x karaktert
TOP(s)	az s string első karakterét mutatja meg
PALIN(s)	igaz, ha az s string palindróma

Egy string palindróma, ha az elejétől olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

Az alábbi táblázatban található kifejezések közül jelölje meg azokat, amelyek algebrai axiómák BAL oldalán állhatnak! (6 pont)

- | | |
|---|---|
| <input type="checkbox"/> TOP(TAIL(s)) | <input type="checkbox"/> PALIN(TAIL(CRT())) |
| <input type="checkbox"/> LGTH(TOP(CRT())) | <input type="checkbox"/> LGTH(TAIL(s)) |
| <input type="checkbox"/> XTEND(TOP(s)) | <input type="checkbox"/> PALIN(XTEND(s, x)) |
| <input type="checkbox"/> TAIL(XTEND(CRT())) | <input type="checkbox"/> LGTH(TOP(s)) |
| <input type="checkbox"/> PALIN(CRT()) | <input type="checkbox"/> TOP(XTEND(s, x)) |

8. A felülvizsgálat (review, walkthrough stb.) során minden egyes „akció elem”-hez (hiba, ellentmondás, javaslat stb.) a következőket kell előírni (6 pont) :

9. A RUP (Rational Unified Process) egyik munkafolyamatában (workflow) szerződés (contract) készítését javasolja illetve írja elő. Melyik munkafolyamatban esedékes szerződés készítése ? Kik között kell szerződést készíteni ? Milyen fontosabb pontjai vannak a szerződésnek ? (4 pont)

Munkafolyamat (workflow):

Szerződő felek:

Szerződés fontosabb pontjai:

Eredmények értékelése:

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