

## Први колоквијум из Објектно оријентисаног програмирања II

- 1) (30 поена) Одговорити концизно (по једна или две реченице) и прецизно на следећа питања:
- а) Да ли је на језику *Java* могуће: (1) написати глобалну функцију, (2) направити објекат класног типа на стеку, (3) приступити локалној променљивој примитивног типа преко референце, (4) написати више конструктора у једној класи, (5) написати деструктор.
  - б) Када се извршавају статички, а када нестатички иницијализациони блокови и којим редоследом (уколико их има више) се извршавају једни, односно други?
  - в) Да ли се заштићеним (*protected*) члановима класе *p1.A* може приступати из: (1) класа које нису изведене из класе *A* и налазе се у истом пакету *p1*; (2) класа које нису изведене из класе *A* и налазе се у потпакету *p1.pp*; (3) класа које су изведене из класе *A* и налазе се у пакету *p2*?

- 2) (укупно 70 поена) Саставити на језику *Java* следећи пакет класа:

- (30 поена) **Парцела** има једнозначан аутоматски генерисан целобројан идентификатор и задату површину (у квадратним метрима). Могу да јој се дохвате садржани подаци, као и једнословна врста. Може да јој се одреди принос у периоду задатом бројем година. Може да се састави текстуални опис у облику *врста-идент [површ]*.
- **Шума** је парцела која има задате површину коју заузима стабло, принос по зрелом стаблу и период сазревања од засада до сече изражен у целом броју година. Ознака врсте је **S**. Принос се остварује тек за период који почиње засадом и траје барем један период сазревања. Сматра се да се после сазревања шума сече и одмах сади нова, тако да се у сваком завршеном периоду сазревања оствари пуни принос. Текстуални облик је *парцела: (површSt, приносSt, период)*.
- **Њива** је парцела која има задат годишњи принос по квадратном метру. Принос по квадратном метру може да се промени. Ознака врсте је **N**. Текстуални облик је *парцела: приносПоМ*.
- (30 поена) **Атар** је парцела која садржи произвољан број других парцела. Ствара се празан или са задатим низом парцела, након чега му се парцеле могу додавати појединачно. Ознака врсте је **A**. Текстуални опис садржи опис самог атара као парцеле, а затим описе садржаних парцела у одговарајућем броју редова.

(10 поена) Написати на језику *Java* **програм** који направи два атара, пар шума и њива које припадају овим атарима, а затим трећи атар који садржи претходна два и једну шуму. Затим испише трећи атар и његов принос за неки период. Користити константне параметре (не треба ништа учитавати).

---

**НАПОМЕНЕ:** а) Колоквијум траје **100** минута.

б) Рад се предаје искључиво у вежбанци за испите (-5 поена за неадекватну вежбанку). Није дозвољено имати поред себе друге листове папира, нити уз себе имати мобилни телефон, без обзира да ли је укључен или искључен.

в) Водити рачуна о уредности. Нечитки делови текста ће бити третирани као непостојећи. Решења задатака навести по горњем редоследу (-1 поен за лош редослед). Препоручује се рад обичном графитном оловком.

г) Резултати колоквијума биће објављени на *Web*-у на адреси:

<http://rti.etf.bg.ac.rs/rti/ir2oo2/index.html/>

```
// Parcela.java
package parcele;
public abstract class Parcela {
    private static int ukID = 0;
    private int id = ++ukID;
    private double povrs;
    public Parcela(double p) {
        povrs = p;
    }
    public int dohID() { return id; }
    public double dohPovrs() {
        return povrs;
    }
    public abstract char vrsta();
    public abstract double prinos(int godine);
    public String toString() {
        return vrsta()+"-"+id+"["+povrs+"]";
    }
}
```

```
// Suma.java
package parcele;
public class Suma extends Parcela {
    private double povrsSt;
    private double prinosSt;
    private int sazrevanje;
    public Suma(double p, double povrs,
                double pr, int s) {
        super(p);
        povrsSt = povrs; prinosSt = pr;
        sazrevanje = s;
    }
    public char vrsta() {
        return 'S';
    }
    public double prinos(int godine) {
        return dohPovrs()/povrsSt*prinosSt*
            (godine/sazrevanje);
    }
    public String toString() {
        return super.toString()+":("+povrsSt+",
            "+prinosSt+", "+sazrevanje+")";
    }
}
```

```
// Njiva.java
package parcele;
public class Njiva extends Parcela {
    private double prinosKvm;
    public Njiva(double p, double prinos) {
        super(p);
        prinosKvm = prinos;
    }
    public void promeniPrinos(double prinos) {
        prinosKvm = prinos;
    }
    public char vrsta() { return 'N'; }
    public double prinos(int godine) {
        return dohPovrs()*prinosKvm*godine;
    }
    public String toString() {
        return super.toString()+":"+prinosKvm;
    }
}
```

```
// Atar.java
package parcele;
public class Atar extends Parcela {
    private static class Elem {
        Parcela parc;
        Elem sled;
        Elem(Parcela p) { parc = p; }
    }
    private Elem prvi, posl;
    public Atar(double p) {
        super(p);
    }
}
```

```
public Atar(double p, Parcela[] parc) {
    super(p);
    for(int i=0;i<parc.length;i++)
        dodaj(parc[i]);
}
public Atar dodaj(Parcela parc) {
    Elem novi = new Elem(parc);
    if (prvi == null) prvi = novi;
    else posl.sled = novi;
    posl = novi;
    return this;
}
public char vrsta() {
    return 'A';
}
public double prinos(int godine) {
    double prinos = 0;
    for(Elem tek=prvi;tek!=null;tek=tek.sled)
        prinos += tek.parc.prios(godine);
    return prinos;
}
public String toString() {
    String str = super.toString()+"\n";
    for(Elem tek=prvi;tek!=null;tek=tek.sled)
        str += tek.parc + "\n";
    return str;
}
}
```

```
// Main.java
package parcele;
public class Main {
    public static void main(String[] varg) {
        Suma s1 = new Suma(100, 2, 20, 3),
            s2 = new Suma(500, 2, 30, 4),
            s3 = new Suma(200, 4, 15, 2);
        Njiva n1 = new Njiva(200, 5),
            n2 = new Njiva(1000, 3);
        Atar a1 = new Atar(2000),
            a2 = new Atar(5000),
            a3 = new Atar(10000);
        a1.dodaj(s1); a1.dodaj(n1);
        a2.dodaj(s2); a2.dodaj(n2);
        a3.dodaj(a1); a3.dodaj(a2);
        a3.dodaj(s3);
        System.out.println(a3);
        System.out.println(a3.prios(5));
    }
}
```

```
A-8[10000.0]
A-6[2000.0]
S-1[100.0]:(2.0, 20.0, 3)
N-4[200.0]:5.0

A-7[5000.0]
S-2[500.0]:(2.0, 30.0, 4)
N-5[1000.0]:3.0

S-3[200.0]:(4.0, 15.0, 2)

30000.0
```