

Prvi kolokvijum iz Objektno orijentisanog programiranja i Java-e

1. Implementirati klasu `A` koja pripada paketu `rs.ac.bg.mas` i koja ima privatni atribut `aA` tipa `int`, zaštićeni atribut `aB` tipa `float`, javni atribut `aC` tipa `char` i atribut `aD` tipa `boolean`, kao i javne metode `mA`, `mB` koje prihvataju po dva argumenta tipa `int` i vraćaju podatak tipa `char`. Omogućiti da objekti kreirani od klasa unutar paketa `rs.ac.bg.mas` imaju javni pristup i privatni u odnosu na objekte kreirane na osnovu klasa iz drugih paketa.

Rešenje. Sadržina filea `A.java`

```
package ra.ac.bg.mas;

public class A {

    private int aA;
    protected float aB;
    public char aC;
    boolean aD;

    public char mA(int a, int b) {
        char c = 'a';
        return c;
    }

    public char mB(int a, int b) {
        char c = 'a';
        return c;
    }
}
```

2. Interface

```
public interface Hodanje {
    public float kreni(float brzina, float trajanje);
    public float stani(float trajanje);
    public float trci(float brzina, float trajanje);
}
```

implementirati u klasi Zivotinja. U metodi `public static void main(String argv[])` kreirati jedan objekat `zivotinja` klase `Zivotinja` i 'hodati'. Ako metode interfeasa `Hodanje` vraćaju predjeni put, odrediti dužinu predjenog puta objekta `zivotinja`.

Rešenje. Sadržina filea `Hodanje.java`

```
package prviKolokvijum;

public interface Hodanje {
    public float kreni(float brzina, float trajanje);
    public float stani(float trajanje);
    public float trci(float brzina, float trajanje);
}
```

Sadržina filea `Zivotinja`

```
package prviKolokvijum;

public class Zivotinja implements Hodanje {
    @Override
    public float kreni(float brzina, float trajanje) {
        return brzina*trajanje;
    }
    @Override
    public float stani(float trajanje) {
        return 0;
    }
    @Override
    public float trci(float brzina, float trajanje) {
        return brzina*trajanje;
    }
    public static void main(String argv[]) {
        Zivotinja z = new Zivotinja();
        float duzinaPuti = 0;
        duzinaPuti += z.kreni(10, 2);
        duzinaPuti += z.trci(20,4);
        System.out.println("Predjeni put:=" + duzinaPuti);
    }
}
```

3. Pretpostavimo da klase su klase A i B implementirane na sledeći način

```
public class A {
    private int aA;
    protected int aB;
    public A(int aa, int bb) {
        aA=aa;
        aB=bb;
    }
}

public class B extends A {
    public B(int aa, int bb) {

    }
}
```

implementirati konstruktor klase B tako da se klase mogu prevesti.

Rešenje. Treba samo obezbediti poziv konstruktoru klase A, dakle, ubaciti liniju

```
super(aa,bb);
```

unutar tela konstruktora klase B.

4. U jednom zoološkom vrtu žive: lav Laza, žirafa Mica, roda Teodora, medved Meda, slonica Cica, žaba Živka, pingvin Pera i foka Sara. Prilikom jedne posete zoološkom vrtu uočili ste prodavca karata Živojina, čuvara vrta Žitomira, vodiča Vukadinku i posetioce Miru, Jelenu, Dijana, Aleksandra i Bojana. Pomenuti posetioći, zajedno sa vama, bili su predvodjeni vodičem Vukadinkom prilikom posete vrtu. Koristeći metode klasifikacije, generalizacije i specijalizacije, formirati hijerarhiju klasa i na osnovu nje simulirati vašu posetu zoološkom vrtu.

Pretpostaviti da vodič nudi grupi posetiocima razgledanje kroz dve rute. Prva ruta obilazi kaveze sledećim redosledom: lav, žirafa, roda, medved, slonica, žaba, pingvin i foka. Druga ruta obilazi kaveze redosledom: foka, pingvin, žaba, slonica, medved, roda, žirafa i lav. Posetioći se izjašnjavaju koju rutu žele slediti i vodič organizuje rutu poštujući volju većine. Poddrazumevati da je poseta grupe posetilaca kavezu sa lavom Lazom, slonicom

Cicom i medvedom Medom završena posle objašnjavanja vodiča Vukadinke, ishrane i oglašavanja životinja. Poseta kavezima sa žabom Živkom, pingvinom Perom i fokom Sarom je završena posle objašnjenja vodiča Vukadinke i ishrane, plivanja i oglašavanja pomenutih životinja. Poseta kaveza sa rodnom Teodorom je završena posle objašnjenja vodiča Vukadinke i ishrane, leta i oglašavanja pomenute životinje.

Rešenje. Podrazumevamo da su sve klase javne, dakle, nalaze se u fileovima koji imaju imena jednaka imenu klase sa ekstenzijom `java`, i da sve klase pripadaju paketu `zoo`.

```
package zoo;
public class Karta {
    private final float cena;
    private boolean prodata;
    public Karta(float cena) {
        this.cena = cena;
        prodata = false;
    }
    public float getCena(){
        return cena;
    }
    public void setProdata() {
        prodata = true;
    }
    public boolean getProdata() {
        return prodata;
    }
}

package zoo;
import java.util.ArrayList;
public class Ruta extends ArrayList<Zivotinja> {
    private String name;
    public Ruta(String name) {
        this.name = name;
    }
}

package zoo;
```

```

import java.util.Iterator;
public class Bice {
    protected String name;
    public Bice(String name) {
        this.name = name;
    }
    public static void main(String argv[]) {
        Lav laza = new Lav("Laza");
        Zirafa mica = new Zirafa("Mica");
        Medved meda = new Medved("Meda");
        Slon cica = new Slon("Cica");
        Zaba zivka = new Zaba("Zivka");
        Pingvin pera = new Pingvin("Pera");
        Foka sara = new Foka("Sara");
        Roda teodora = new Roda("Teodora");

        ProdavacKarata zivojin = new ProdavacKarata("Zivojin","0");
        Cuvar zitomir = new Cuvar("Zitomir","1");
        zitomir.cuvaj();
        Vodica vukadinka = new Vodica("Vukadinka","2");

        Ruta rutaJedan = new Ruta("Jedan");
        rutaJedan.add(laza);
        rutaJedan.add(mica);
        rutaJedan.add(teodora);
        rutaJedan.add(meda);
        rutaJedan.add(cica);
        rutaJedan.add(zivka);
        rutaJedan.add(pera);
        rutaJedan.add(sara);

        Ruta rutaDva = new Ruta("Dva");
        rutaDva.add(sara);
        rutaDva.add(pera);
        rutaDva.add(zivka);
        rutaDva.add(cica);
        rutaDva.add(meda);
        rutaDva.add(teodora);
    }
}

```

```

rutaDva.add(mica);
rutaDva.add(laza);

Posetilac mira = new Posetilac("Mira",1);
mira.uzmiKartu(zivojin.dajKartu());
Posetilac jelena = new Posetilac("Jelena",2);
jelena.uzmiKartu(zivojin.dajKartu());
Posetilac dijana = new Posetilac("Dijana",1);
dijana.uzmiKartu(zivojin.dajKartu());
Posetilac aleksandra = new Posetilac("Aleksandra",2);
aleksandra.uzmiKartu(zivojin.dajKartu());
Posetilac bojana = new Posetilac("Bojana",1);
bojana.uzmiKartu(zivojin.dajKartu());

vukadinka.dodajGlas(mira.biraj());
vukadinka.dodajGlas(jelena.biraj());
vukadinka.dodajGlas(dijana.biraj());
vukadinka.dodajGlas(aleksandra.biraj());
vukadinka.dodajGlas(bojana.biraj());

Ruta ruta;
if(vukadinka.objaviRutu() == 1)
    ruta = rutaJedan;
else
    ruta = rutaDva;

for (Zivotinja z : ruta) {
    vukadinka.objasni(z);
    z.jedi();
    z.oglasise();
    if(z instanceof VodenaZivotinja)
        ((VodenaZivotinja) z).plivaj();
    if(z instanceof LetecaZivotinja)
        ((LetecaZivotinja) z).leti();
}
}
}

```

```
package zoo;
public class Covek extends Bice {
    public Covek(String name) {
        super(name);
    }
}
```

```
package zoo;
public class Zaposleni extends Covek {
    String ID;
    public Zaposleni(String name, String ID) {
        super(name);
        this.ID = ID;
    }
}
```

```
package zoo;
public class Cuvar extends Zaposleni {
    public Cuvar(String name, String ID) {
        super(name, ID);
    }
    public void cuvaj() {

    }
}
```

```
package zoo;
public class ProdavacKarata extends Zaposleni {
    public ProdavacKarata(String name, String ID) {
        super(name, ID);
    }
    public Karta dajKartu() {
        Karta k = new Karta(100);
        k.setProdata();
        return k;
    }
}
```

```
package zoo;
```

```

public class Vodici extends Zaposleni {
    private int rutaJ;
    private int rutaD;
    public Vodici(String name, String ID) {
        super(name, ID);
        rutaJ = 0;
        rutaD = 0;
    }
    public String objasni(Zivotinja z) {
        return "Ovo je "+z.getName();
    }
    public void dodajGlas(int ruta) {
        if(ruta == 1)
            ++rutaJ;
        if(ruta == 2)
            ++rutaD;
    }
    public int objaviRutu() {
        if(rutaJ > rutaD)
            return rutaJ;
        else
            return rutaD;
    }
}

```

```

public class Zivotinja extends Bice {
    public Zivotinja(String name) {
        super(name);
    }
    public String getName() {
        return name;
    }
    public void jedi() {

    }
    public void oglasise() {

    }
}

```



```
}
```

```
package zoo;  
public class LetecaZivotinja extends Zivotinja {  
    public LetecaZivotinja(String name) {  
        super(name);  
    }  
    public void leti() {  
  
    }  
}
```

```
package zoo;  
public class VodenaZivotinja extends Zivotinja {  
    public VodenaZivotinja(String name) {  
        super(name);  
    }  
    public void plivaj() {  
  
    }  
}
```

```
package zoo;  
public class Lav extends Zivotinja {  
    public Lav(String name) {  
        super(name);  
    }  
}
```

```
package zoo;  
public class Zirafa extends Zivotinja {  
    public Zirafa(String name) {  
        super(name);  
    }  
}
```

```
package zoo;  
public class Roda extends LetecaZivotinja {  
    public Roda(String name) {
```

```

        super(name);
    }
}

package zoo;
public class Medved extends Zivotinja {
    public Medved(String name) {
        super(name);
    }
}

package zoo;
public class Slon extends Zivotinja {
    public Slon(String name) {
        super(name);
    }
}

package zoo;
public class Zaba extends VodenaZivotinja {
    public Zaba(String name){
        super(name);
    }
}

package zoo;
public class Pingvin extends VodenaZivotinja {
    public Pingvin(String name) {
        super(name);
    }
}

package zoo;
public class Foka extends VodenaZivotinja {
    public Foka(String name) {
        super(name);
    }
}

```