ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ Београд 23.03.2016.

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 Јава могуће створити: (1) низ који садржи по вредности податке простог (примитивног) типа на стеку, (2) низ који садржи по вредности податке класног типа на стеку, (3) низ који садржи по вредности податке простог типа у меморији за динамичку алокацију (*heap*), (4) низ који садржи по вредности податке класног типа у меморији за динамичку алокацију (*heap*)?

б) Која права приступа су могућа, који модификатор их означава и које је право подразумевано за (1) чланове класе (2) чланове пакета?

в) Шта означава: (1) `this.a`, (2) `this(a)`, (3) `super.a` и (4) `super(a)`?

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

- *(30 поена)* ***Авион*** има једнозначан аутоматски генерисан целобројан серијски број и задату сопствену тежину (у килограмима). Могу да му се дохвате серијски број, једнословна ознака врсте и да му се одреди укупна тежина. Може да се састави текстуални опис авиона у облику *врста*`_`*серијски број*`[`*тежина*`]`.
- ***Путнички*** авион има задат капацитет (број путника), подразумевано 400. Ознака врсте је **`P`**. Приликом одређивања укупне тежине путничког авиона, сматра се да је авион потпуно попуњен (до пуног капацитета), при чему сваки путник има пртљаг. Просечна тежина путника у авиону износи 80 килограма, а просечна тежина пртљага 30 килограма.
- ***Теретни*** авион има задату носивост (у килограмима), подразумевано 40000. Ознака врсте је **`T`**. Приликом одређивања укупне тежине теретног авиона, сматра се да је авион оптерећен теретом до максимума носивости.
- *(30 поена)* ***Аеродром*** садржи задати број места за авионе, максималну тежину авиона који може да слети на аеродром и цену аеродромске таксе по јединици укупне тежине (килограму) авиона. Ствара се празан, након чега му се авиони могу додавати (слетати) и узимати (полетати), појединачно. Приликом слетања, задаје се авион и редни број места на које слеће, а резултат извршавања је успешност операције. Приликом полетања, задаје се редни број места са којег авион полеће, а резултат извршавања је успешност операције. Може да се одреди приход аеродрома као сума свих такси које се наплаћују свим авионима који се у том тренутку налазе на њему. Текстуални опис садржи описе садржаних места (опис авиона ако је место пуно, иначе текст **`<<prazno>>`**) у одговарајућем броју редова.

(10 поена) Написати на језику *Java* **програм** који направи аеродром, и направи неколико путничких и теретних авиона који слете и полете са њега. Затим испише аеродром и његов приход. 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/`

```java
//Avion.java
package aerodrom;

public abstract class Avion {
   private static int ukID = 0;
   private final int id = ++ukID;
   private double tezina;

   public Avion(double t)
   {  tezina = t;   }

   public int dohID()
   {  return id; }

   public double dohTez()
   {  return tezina;  }

   public abstract char vrsta();

   public String toString()
   {  return vrsta() + "_" + dohID() + "[" +
                           dohTez() + "]"; }
}
```

```java
//Putnicki.java
package aerodrom;

public class Putnicki extends Avion {
   private static final double
                              TEZ_PUTNIKA = 80;
   private static final double
                           TEZ_PRTLJAGA = 30;
   private int kapacitet;

   public Putnicki(double t, int n)
   {  super(t);
      kapacitet = n;  }

   public Putnicki(double t)
   {  this(t, 400); }

   @Override
   public char vrsta()
   {  return 'P';   }

   @Override
   public double dohTez()
   {  return super.dohTez() + kapacitet *
                 (TEZ_PUTNIKA + TEZ_PRTLJAGA); }
}
```

```java
//Teretni.java
package aerodrom;

public class Teretni extends Avion {
   private double nosivost;

   public Teretni(double t, double n)
   {  super(t);
      nosivost = n; }

   public Teretni(double t)
   {  this(t, 40000); }

   @Override
   public char vrsta()
   {  return 'T';   }

   @Override
   public double dohTez()
   {  return super.dohTez() + nosivost;  }
}
```

```java
//Aerodrom.java
package aerodrom;

public class Aerodrom {
   private Avion[] mesta;
   private double maxTez;
   private double jedCena;

   public Aerodrom(int n, double t, double c)
   {  mesta = new Avion[n];
      maxTez = t; jedCena = c;   }

   public boolean sleti(Avion a, int i) {
      if (a.dohTez() > maxTez) return false;
      if (i < 0 || i >= mesta.length)
                                  return false;
      if (mesta[i] != null) return false;
      mesta[i] = a;
      return true;
   }

   public boolean poleti(int i) {
      if (i < 0 || i >= mesta.length)
                                  return false;
      if (mesta[i] == null) return false;
      mesta[i] = null;
      return true;
   }

   public double prihod() {
      double prihod = 0;
      for(int i = 0; i < mesta.length; i++)
         if (mesta[i] != null)
            prihod += mesta[i].dohTez()*jedCena;
      return prihod;
   }

   public String toString() {
      StringBuilder str = new StringBuilder();
      for(int i = 0; i < mesta.length; i++)
         str.append((mesta[i] != null) ?
                        mesta[i] : "<<prazno>>")
            .append('\n');
      return str.toString();
   }
}
```

```java
//Main.java
package aerodrom;

public class Main {
   public static void main(String[] args) {
      Putnicki p1 = new Putnicki(300000),
           p2 = new Putnicki(250000,425);
      Teretni t1 = new Teretni(200000),
           t2 = new Teretni(230000,120000);
      Aerodrom a = new Aerodrom(10,430000,0.1);
      a.sleti(p1, 3); a.sleti(p2, 7);
      a.sleti(t1, 9); a.sleti(t2, 0);
      a.poleti(7); a.poleti(1);
      System.out.println(a);
      System.out.println(a.prihod());
   }
}
```

```
T_4[350000.0]
<<prazno>>
<<prazno>>
P_1[344000.0]
<<prazno>>
<<prazno>>
<<prazno>>
<<prazno>>
<<prazno>>
T_3[240000.0]

93400.0
```