ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ

Београд 24.03.2017.

Први колоквијум из
Објектно оријентисаног програмирања II

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

- *(10 поена)* ***Планина*** има име и висину (у метрима), који могу да се дохвате. Може да се састави текстуални опис у облику *име* **(***висина***)**, као и текстуални опис низа планина у облику *планина* **,** *планина* **, ...,** *планина*.
- *(30 поена)* ***Планинар*** има име, јединствени целобројни идентификациони број и збирку планина задатог капацитета. Збирка се ствара празна, након чега планине могу да се додају једна по једна. Планинар може да се пење по задатој планини. Пењање треба да врати индикатор успешности (*true*/*false*). Уколико је пењање било успешно, задата планина се ставља у збирку планина и враћа се *true*. Уколико је пењање било неуспешно враћа се *false*. Може да се састави текстуални опис планинара у облику *име*-*идентификатор* **(***планина* **,** *планина* **, ...,** *планина***)**.
- ***Класични планинар*** је планинар који се пење по планинама нижим од 2000m, иначе је пењање неуспешно. Текстуални облик је **K_***име*-*идент* **(***планина* **,** *планина* **, ...,** *планина***)**.
- ***Алпиниста*** је планинар коме може да се додели други алпиниста као партнер и који се пење по планинама вишим од 3000m. Партнер не може да буде класични планинар. Уколико алпиниста нема додељеног партнера или је планина нижа од 3000m, пењање је неуспешно. Текстуални опис је **A_***име*-*идент* **(***планина* **,** *планина* **, ...,** *планина***)**.
- *(30 поена)* ***Планинарско друштво*** има назив, годину оснивања и листу планинара. Ствара се са непразном листом планинара (оснивача). Планинар може да се учлани, при чему се додаје на крај листе. Може да се дохвати планинар са задате позиције у листи (позиције иду од 1) и да се дохвати укупан број планинара у листи. Текстуални опис садржи опис сваког планинара у посебном реду.

(0 поена) Приложена је класа са главном функцијом која направи две планине, *Монт Блан* висине *4807m* и *Златибор* висине *1496m*, а затим планинарско друштво са задатим алпинистом и класичним планинаром. Затим се планинарском друштву додају још један алпиниста и још један класични планинар. Након тога се првом планинару у друштву зада да се пење на Монт Блан, а другом да се пење на Златибор. Затим се првом планинару у друштву додели трећи планинар као партнер и обрнуто и оба планинара се пошаљу на Монт Блан. После сваког слања на планину на стандардном излазу се испише планинар, планина и статус успеха пењања. На крају се прикаже планинарско друштво на стандардном излазу.

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

б) Рад се предаје на предвиђеном мрежном диску.

в) На располагању је приступ сајту Oracle. Није дозвољено имати поред себе друге материјале, нити уз себе имати електронске уређаје, без обзира да ли су укључени или искључени.

г) Резултати колоквијума биће објављени на *Web*-у на адреси: `http://rti.etf.rs/rti/ir2oo2/index.html/`

```
package planinarenje;
public class Planina {
  private String ime;
  private int visina;
  public Planina(String ime, int visina){
    this.ime = ime; this.visina = visina; }
  public String getIme() { return ime; }
  public int getVisina() { return visina; }
  public String toString(){
    return ime + "(" + visina + ")"; }
  public static String toString (
    Planina[] niz) {
    StringBuilder s = new StringBuilder();
    for(int i = 0; i < niz.length; i++){
      if(niz[i] == null) continue;
      if (i != 0)s.append(", ");
      s.append(niz[i]);
    } return s.toString();
  }
}
```

```
package planinarenje;
public abstract class Planinar {
  private String ime;
  private static int ukID = 0;
  private int id = ++ukID;
  private Planina[] zbirka;
  private int pop;
  public Planinar(String ime, int kap){
    this.ime = ime;
    zbirka = new Planina[kap]; }
  public void dodaj(Planina p){
    if (pop < zbirka.length){
      zbirka[pop] = p; pop++;
    }
  }
  protected abstract boolean penjiSe(Planina p);
  public boolean popniSe(Planina p){
    if (penjiSe(p)){
      dodaj(p); return true;
    } return false;
  }
  public String toString(){
    return ime + "-" + id + "(" +
      Planina.toString(zbirka) + ")";
   }
}
```

```
package planinarenje;
public class KlasicniPlaninar extends Planinar {
  private static int V = 2000;
  public KlasicniPlaninar(String ime, int kap){
     super(ime, kap); }
  protected boolean penjiSe(Planina p) {
    if (p.getVisina() < V) return true;
    else return false;
  }
  public String toString(){
    return "K_" + super.toString(); } }
class Alpinista extends Planinar {
  private static int V = 3000;
  private Planinar partner;
  public Alpinista(String ime, int kap) {
    super(ime, kap);
  }
  public void dodeli(Planinar p){
    if (!(p instanceof KlasicniPlaninar))
      partner = p;
  }
  protected boolean penjiSe(Planina p) {
    if (partner != null && p.getVisina() >= V)
        return true;
     else return false;
  }
  public String toString(){
    return "A_" + super.toString();
  }
}
```

```
package planinarenje;
public class PlaninskoDrustvo {
  private String naziv;
  private int godina;
  private Elem prvi, posl;
  private static class Elem{
    Planinar p; Elem sled;
    public Elem(Planinar p){
      this.p = p; }
  }
  public PlaninskoDrustvo(String naziv,
    int godina, Planinar[] osnivaci) {
    this.naziv = naziv; this.godina = godina;
    for (int i = 0; i < osnivaci.length; i++)
      uclani(osnivaci[i]);
  }
  public PlaninskoDrustvo uclani(Planinar p){
    Elem novi = new Elem(p);
    if (prvi == null) prvi = novi;
    else posl.sled = novi;
    posl = novi; return this;
  }
  public Planinar dohvatiPlaninara(int i){
    int k = 1; Elem tek = prvi;
    while(tek != null && k < i){
      tek = tek.sled;
      k++; }
    if (tek != null) return tek.p;
    return null;
  }
  public int brojPlaninara(){
    int k = 0; Elem tek = prvi;
    while(tek != null){
      tek = tek.sled; k++;
    } return k;
  }
  public String toString(){
    StringBuilder s = new StringBuilder();
    for(Elem tek = prvi; tek != null;
        tek = tek.sled){
        s.append(tek.p + "\n");
    }
    return s.toString();
  }
}
```

```
import planinarenje.*;
public class Test {
  private static void popniSe(Planinar planinar,
    Planina planina){
    if(planinar.popniSe(planina) == true)
      System.out.println(planinar + " - USPEH");
    else
      System.out.println(planinar +
        " " + planina +" - NEUSPEH");
  }
  public static void main(String[] args){
    Planina montBlan =
      new Planina("Mont Blan", 4807);
    Planina zlatibor =
      new Planina("Zlatibor", 1496);
    PlaninskoDrustvo drustvo =
      new PlaninskoDrustvo("Drustvo", 2000,
      new Planinar[]{
        new Alpinista("Ognjen", 5),
        new KlasicniPlaninar("Marko", 10)});
    drustvo.uclani(new Alpinista("Zoran", 15)).
    uclani(new KlasicniPlaninar("Milos", 2));
popniSe(drustvo.dohvatiPlaninara(1), montBlan);
popniSe(drustvo.dohvatiPlaninara(2), zlatibor);
    if(drustvo.dohvatiPlaninara(1)
      instanceof Alpinista)
   ((Alpinista)drustvo.dohvatiPlaninara(1)).
     dodeli(drustvo.dohvatiPlaninara(3));
    if(drustvo.dohvatiPlaninara(3)
       instanceof Alpinista)
   ((Alpinista)drustvo.dohvatiPlaninara(3)).dod
eli(drustvo.dohvatiPlaninara(1));
popniSe(drustvo.dohvatiPlaninara(1), montBlan);
popniSe(drustvo.dohvatiPlaninara(3), montBlan);
  }
}
```

```
A_Ognjen-1() Mont Blan(4807) - NEUSPEH
K_Marko-2(Zlatibor(1496)) - USPEH
A_Ognjen-1(Mont Blan(4807)) - USPEH
A_Zoran-3(Mont Blan(4807)) - USPEH
```