

Vježbe 7

Kreirati klasu `Takmicar` koja ima podatke članove ime, prezime, broj osvojenih medalja i godište. Sve takmičare smjestiti u listu i odštampati je. Implementirati mehanizam za dvije vrste sortiranja takmičara. U prvom slučaju, takmičare treba sortirati po broju osvojenih medalja, a u drugom slučaju po godištu. Demonstrirati sortiranje upotrebom `Set` kolekcije.

```
public class Takmicar {  
  
    String ime, prezime;  
    int brojMedalja, godiste;  
  
    public Takmicar(String a, String b, int c, int d) {  
        setIme(a);  
        setPrezime(b);  
        setBrojMedalja(c);  
        setGodiste(d);  
    }  
  
    public String getIme() {  
        return ime;  
    }  
    public void setIme(String a) {  
        ime = a;  
    }  
    public String getPrezime() {  
        return prezime;  
    }  
    public void setPrezime(String a) {  
        prezime = a;  
    }  
    public int getBrojMedalja() {  
        return brojMedalja;  
    }  
    public void setBrojMedalja(int a) {  
        brojMedalja = a;  
    }  
    public int getGodiste() {  
        return godiste;  
    }  
    public void setGodiste(int a) {  
        godiste = a;  
    }  
  
    @Override  
    public String toString() {  
        return "Takmicar [ime=" + getIme() +  
            ", prezime=" + getPrezime() +  
            ", brojMedalja=" + getBrojMedalja() +  
            ", godiste=" + getGodiste() + "];"  
    }  
}
```

```

import java.util.Comparator;
public class TakmicarUspjesniji implements Comparator< Takmicar > {
    public int compare(Takmicar prvi, Takmicar drugi) {
        if(prvi.getBrojMedalja() == drugi.getBrojMedalja())
            return 0;
        else if(prvi.getBrojMedalja() > drugi.getBrojMedalja())
            return 1;
        else
            return -1;
    }
}

```

```

import java.util.Comparator;
public class TakmicarStariji implements Comparator< Takmicar > {
    public int compare(Takmicar prvi, Takmicar drugi) {
        if(prvi.getGodiste() == drugi.getGodiste())
            return 0;
        else if(prvi.getGodiste() > drugi.getGodiste())
            return -1;
        else
            return 1;
    }
}

```

```

import java.util.*;
public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Takmicar t1 = new Takmicar("Marko", "Peric", 16, 1998);
        Takmicar t2 = new Takmicar("Ivan", "Ilic", 17, 1996);
        Takmicar t3 = new Takmicar("Denis", "Pepic", 21, 1999);
        Takmicar t4 = new Takmicar("Jagos", "Maric", 9, 2001);

        LinkedList<Takmicar> takmicari = new LinkedList<Takmicar>();
        takmicari.add(t1);takmicari.add(t2);
        takmicari.add(t3);takmicari.add(t4);
        takmicari.add(t1);takmicari.add(t2);

        Iterator<Takmicar> iter = takmicari.iterator();
        System.out.println("Štampanje liste pomoću iteratora:");
        while(iter.hasNext())
            System.out.println(iter.next() + " ");

        Collections.sort(takmicari, new TakmicarUspjesniji());

        System.out.println("\nPoredjenje po uspjehnosti:");
        for(Takmicar takmicar: takmicari)
            System.out.println(takmicar);

        Collections.sort(takmicari, new TakmicarStariji());

        System.out.println("\nPoredjenje po starosti:");
        for(Takmicar takmicar: takmicari)
            System.out.println(takmicar);
    }
}

```

```
Set<Takmicar> hSet = new HashSet<Takmicar>(takmicari);
Set<Takmicar> tSet1 = new TreeSet<Takmicar>(new TakmicarUspjesniji());
tSet1.addAll(hSet);

System.out.println("\nPo uspjehnosti(TreeSet):");
    for(Takmicar takmicar: tSet1)
        System.out.println(takmicar);
    }
}
```