Laboratorijske vežbe – dvočas br. 8

Zadatak 1

(radi ga laborant u saradnji sa studentima, a jedan deo studenti rade sami)

Napraviti javnu klasu ParkingMesto koja ima:

- Privatni atribut slobodno koji predstavlja indikator zauzetosti parking mesta. Ovaj indikator ima vrednost TRUE ako je mesto slobodno a FALSE ako nije.
- Privatni atribut registarskiBroj koji predstavlja registarski broj vozila koje se nalazi na tom parking mestu (String).
- Odgovarajuće javne get i set metode za ova dva atributa. ***(ponoviti automatsko generisanje get i set metoda u Eclipse okruženju)***

Napraviti javnu klasu Parking koja ima:

- Privatni atribut mesta koji predstavlja niz objekata klase ParkingMesto. ***(niz čiji su elementi objekti, a ne prosti tipovi podataka)***
- Javni konstruktor koji kao ulazni argument prima broj koji predstavlja kapacitet parkinga tj. ukupan broj parking mesta. Ako je uneti broj veći od nule, potrebno je inicijalizovati atribut mesta na taj broj. Ako je uneti broj nula ili manji od nule, kapacitet parking mesta se postavlja na 40 i ispisuje se poruka o grešci na ekranu. U svakom slučaju je potrebno inicijalizovati svako parking mesto i postaviti ga da bude slobodno. ***(objasniti da su elementi niza objekti pa se moraju i oni pojedinačno inicijalizovati)***
- Javnu metodu koja proverava da li na parkingu ima slobodnih mesta i vraća TRUE ako ima, a FALSE ako nema.
- Javnu metodu za 'uvođenje' vozila na parking. Ova metoda kao ulazni argument dobija registarski broj vozila. Prvo je potrebno proveriti da li na parkingu ima slobodnih mesta. Ako ima, potrebno je uvesti vozilo na prvo slobodno mesto. Ako slobodnih mesta nema, ispisati poruku o tome na ekranu.
  (naredne dve metode studenti rade sami)
- Javnu metodu koja kao ulazni argument prima registarski broj vozila i proverava da li se to vozilo nalazi na parkingu. Ako se nalazi, metoda vraća TRUE a u suprotnom FALSE.
- Javnu metodu za "izvođenje" vozila sa parkinga. Ova metoda kao ulazni argument dobija registarski broj vozila. Prvo je potrebno proveriti da li se vozilo sa tim registarskim brojem nalazi na parkingu. Ako se nalazi, potrebno ga je izvesti, tako da parking mesto ponovo postane slobodno. Izvodjenje vozila podrazumeva da se mesto na kome je bilo označi kao slobodno i da se podatak o njegovom registarskom broju ukloni.

Napraviti klasu **TestParking** koja kreira jedan objekat klase Parking kapaciteta 20 mesta i u njega unosi kola sa tablicama "BG 123-456" i "NS 234-56".

```
public class ParkingMesto {

	private boolean slobodno;

	private String registarskiBroj;

	public String getRegistarskiBroj() {
		return registarskiBroj;
	}

	public void setRegistarskiBroj(String registarskiBroj) {
		this.registarskiBroj = registarskiBroj;
	}

	public boolean isSlobodno() {
		return slobodno;
	}

	public void setSlobodno(boolean slobodno) {
		this.slobodno = slobodno;
	}
```

```java
}

public class Parking {

        private ParkingMesto[] mesta;

        public Parking (int brojMesta){
                if (brojMesta > 0){
                        mesta = new ParkingMesto[brojMesta];
                }
                else{
                        System.out.println("Greska");
                        mesta = new ParkingMesto[40];
                }

                for (int i=0; i<mesta.length;i++){
                        mesta[i] = new ParkingMesto();
                        mesta[i].setSlobodno(true);
                }
        }

        public boolean imaSlobodnih(){
                for (int i=0; i<mesta.length;i++)
                        if (mesta[i].isSlobodno()) return true;

                return false;
        }

        public boolean daLiJeNaParkingu (String regBr){
                for (int i=0; i<mesta.length;i++)
                        if (!(mesta[i].isSlobodno())){
                                String regBr1 = mesta[i].getRegistarskiBroj();
                                if (regBr1.equals(regBr)) return true;
                        }

                return false;
        }

        public void uvediNaParking (String regBr){
                if (!imaSlobodnih())
                        System.out.println("Nema slobodnih mesta");
                else{
                        for (int i=0; i<mesta.length;i++)
                                if (mesta[i].isSlobodno()){
                                        mesta[i].setSlobodno(false);
                                        mesta[i].setRegistarskiBroj(regBr);
                                        break;
                                }
                }
        }

        public void izvediSaParkinga (String regBr){
                if (!daLiJeNaParkingu(regBr))
                        System.out.println("To vozilo se ne nalazi na parkingu");
                else{
                        for (int i=0; i<mesta.length;i++)
                                if (!mesta[i].isSlobodno() &&
                                     mesta[i].getRegistarskiBroj().equals(regBr)){
                                        mesta[i].setSlobodno(true);
                                        mesta[i].setRegistarskiBroj(null);
                                        break;
                                }
                }
        }

}

public class TestParking {

        public static void main(String[] args) {

                Parking p = new Parking(20);

                p.uvediNaParking("BG 123-456");
                p.uvediNaParking("NS 234-56");

        }

}
```

Zadatak 2
(radi ga laborant u saradnji sa studentima, jedan deo studenti rade sami)

Napraviti javnu klasu Motocikl koja ima:

- Privatni atribut marka.
- Privatni atribut model.
- Privatni atribut kubikaza koji predstavlja broj kubika motocikla (ceo broj).
- Odgovarajuće get i set metode za ove atribute.
- Redefinisanu toString metodu klase Object koja vraća String sa svim podacima o motociklu uz odgovarajuću poruku.
- Redefinisanu equals metodu klase Object koja kao ulazni argument prima objekat klase Object, ali se smatra da će se zaista unositi objekti klase Motocikl. Ova metoda vraća true ako je vrednost atributa marka, model i kubikaza jednaka marki, modelu i kubikazi motocikla koji je unet kao ulazni argument. U suprotnom, metoda vraća false.

Napraviti javnu klasu BazaMotocikala koja ima:

- Privatni atribut motocikli koji predstavlja listu objekata klase Motocikl. ***(liste, LinkedList klasa, i napomena da se ona nalazi u "java.util" biblioteci)***
- Javni konstruktor koji inicijalizuje atribut motocikli.***(inicijalizacija liste)***
- Javnu metodu daLiJeUBazi koja kao ulazni argument prima objekat klase Motocikl i proverava da li se isti motocikl već nalazi u bazi.***(contains metoda i napomena da ona koristi equals metodu da proveri da li je to taj objekat)***

  (naredne dve metode studenti rade sami - napomenuti da pronadju odgovarajuće metode iz padajuće liste koju Eclipse izbacuje)
- Javnu metodu unesiUBazu koja kao ulazni argument prima objekat klase Motocikl i unosi ga u listu. Unošenje se vrši samo ako uneta vrednost nije null i ako u bazi već ne postoji isti motocikl. ***(add metoda)***
- Javnu metodu izbaciIzBaze koja kao ulazni argument prima objekat klase Motocikl i briše ga iz baze. Brisanje se vrši samo ako u bazi postoji isti motocikl. ***(remove metoda)***
- Javni metodu ispisi koja na ekranu ispisuje podatke o svim motociklima. ***(get metoda)***

Napraviti klasu TestBazeMotocikala koja kreira objekat klase BazaMotockala i tri objekta klase Motocikl :"Honda" - "CB 750 F" - 748, "Kawasaki" - "ER 5" - 498 i "Honda" - "CB 750 F" - 748. Potrebno je prva dva objekta ubaciti u bazu i ispisati sadržaj baze. Nakon toga, potrebno je proveriti da li se treći motocikl već nalazi u bazi i izbaciti ga iz baze. Ponovo na ekranu ispisati sadržaj baze.

```
public class Motocikl {

        private String marka;
        private String model;
        private int kubikaza;

        public int getKubikaza() {
                return kubikaza;
        }
        public void setKubikaza(int kubikaza) {
                this.kubikaza = kubikaza;
        }
        public String getMarka() {
                return marka;
        }
        public void setMarka(String marka) {
                this.marka = marka;
        }
        public String getModel() {
                return model;
        }
        public void setModel(String model) {
                this.model = model;
        }

        public String toString(){
```

```
                return "Marka: "+marka+" Model: "+model+" Kubikaza: "+kubikaza;
        }

        public boolean equals(Object o){
                Motocikl m = (Motocikl)(o);

                if (marka.equals(m.getMarka())&&
                                (model.equals(m.getModel()))&&
                                kubikaza == m.getKubikaza()) return true;
                else return false;
        }

}

import java.util.*;

public class BazaMotocikala {

        private LinkedList <Motocikl> motocikli;

        public BazaMotocikala(){
                motocikli = new LinkedList <Motocikl> ();
        }

        public boolean daLiJeUBazi(Motocikl m){
                return motocikli.contains(m);
        }

        public void unesiUBazu(Motocikl m){
                if (daLiJeUBazi(m))
                        System.out.println("Taj motocikl se vec nalazi u bazi");
                else motocikli.add(m);
        }

        public void izbaciIzBaze(Motocikl m){
                if (!daLiJeUBazi(m))
                        System.out.println("Taj motocikl se ne nalazi u bazi");
                else {
                        motocikli.remove(m);
                }
        }

        public void ispisi(){
                for (int i=0; i<motocikli.size();i++)
                        System.out.println(motocikli.get(i));
        }

}

public class TestBazaMotocikala {

        public static void main(String[] args){

                BazaMotocikala b = new BazaMotocikala();

                Motocikl m1 = new Motocikl();
                m1.setMarka("Honda");
                m1.setModel("CB 750 F");
                m1.setKubikaza(748);

                Motocikl m2 = new Motocikl();
                m2.setMarka("Kawasaki");
                m2.setModel("ER 5");
                m2.setKubikaza(498);

                b.unesiUBazu(m1);
                b.unesiUBazu(m2);

                Motocikl m3 = new Motocikl();
                m3.setMarka("Honda");
                m3.setModel("CB 750 F");
                m3.setKubikaza(748);
                b.ispisi();

                System.out.println("Da li postoji "+b.daLiJeUBazi(m3));
                System.out.println("Izbacen ");
                b.izbaciIzBaze(m3);
                b.ispisi();
        }
}
```