

Универзитет у Крагујевцу
Факултет инжењерских наука



Пројектна документација:
Rent a Car програм

Студент:
Јован Петровић
556/2015

Ментор:
Проф. др Александар
Пеулић

Садржај

1. Увод.....	2
2. Имплементација.....	3
3. UML дијаграми.....	8
3.1. Дијаграм класа.....	9
3.2. Use case дијаграм.....	10

1. Увод

У овој пројектној документацији је објашњено како функционише програм за одабир возила у једној фирми за изнајмљивање возила. Програм представља интерфејс који комуницира са потенцијалном муштеријом која жели да изнајми возило, или пак само да изврши увид у избор возила. Терминал је реализован да му се може приступити преко интернета или на самом ауто плацу.

2. Имплементација

Свако возило поседује сопствене карактеристике, а у овом програму то су име модела, просечна брзина, потрошња горива, врста горива и снага.

```
package vozilo;

public class Vozilo {

    protected String model;
    protected double prosečnaBrzina; //izrazena u km/h
    protected double potrošnja; //prosečna,izrazena u l/100km
    protected String gorivo; //tip goriva:benzin ili dizel
    protected int snaga; //izrazena u kW
```

Постоји више конструктора али је препоручено инстанцирати објекат уношењем свих аргумента одмах.

```
    protected Vozilo(String model, double prosečnaBrzina, double potrošnja,
int snaga) {
        this.model=model;
        this.prosečnaBrzina=prosečnaBrzina;
        this.potrošnja=potrošnja;
        this.snaga=snaga;
        this.gorivo="nedefinisano";
    }

    protected Vozilo(String model, double prosečnaBrzina, double potrošnja,int
snaga,String gorivo) {
        this.model=model;
        this.prosečnaBrzina=prosečnaBrzina;
        this.potrošnja=potrošnja;
        this.snaga=snaga;
        this.gorivo=gorivo;
```

```
}  
protected Vozilo() {  
    this.model="nepoznat";  
    this.prosecnaBrzina=0;  
    this.potrosnja=0;  
    this.gorivo="nedefinisano";  
    this.snaga=0;  
}
```

У случају да уносилац података не одреди карактеристике возила, постоје методе за накнадни унос и испис истих.

```
public String getModel() {  
    return model;  
}  
public void setModel(String model) {  
    this.model = model;  
}  
public int getSnaga() {  
    return snaga;  
}  
public void setSnaga(int snaga) {  
    this.snaga = snaga;  
}  
public double getProsecnaBrzina() {  
    return prosecnaBrzina;  
}  
public void setProsecnaBrzina(int prosecnaBrzina) {  
    this.prosecnaBrzina = prosecnaBrzina;  
}  
public double getPotrosnja() {  
    return potrosnja;  
}  
public void setPotrosnja(int potrosnja) {  
    this.potrosnja = potrosnja;  
}  
public String getGorivo() {  
    return gorivo;  
}  
public void setGorivo(String gorivo) {  
    this.gorivo = gorivo;  
}
```

За информације које се разликују од корисника до корисника, дефинишу се методе за израчунавање потребног горива и потребног времена у зависности од пута који корисник планира да пређе. Такође постоји опција за упоређивање снаге два возила, у случају да је та карактеристика од важности.

```
public double potrebnoGoriva(double g){
    return g/100*potrosnja;
}

public double potrebnoVremena(double k){
    return k*60/this.prosecnaBrzina;
}

public void uporediSnagu(Vozilo vozilo2){
    if (this.snaga > vozilo2.snaga){
        System.out.println("Возило " + this.model + " је снажније од
возила " + vozilo2.model + ".\n");
    }
    else if (this.snaga < vozilo2.snaga){
        System.out.println("Возило " + this.model + " је слабије од
возила " + vozilo2.model + ".\n");
    }
    else
        System.out.println("Возила су исте снаге.\n");
}
```

Последња од укључених функција програма је текстуални испис свих карактеристика изабраног возила.

```
public void infoVozilo(){
    System.out.println("Назив модела: " + this.model);
    System.out.println("Просечна брзина возила износи " +
this.prosecnaBrzina + "km/h");
    System.out.println("Потрошња возила износи " + this.potrosnja +
"l/100km");
    System.out.println("Снага возила износи " + this.snaga + "kW");
    System.out.println("Ово возило иде на " + this.gorivo + "\n");
}
```

Постоје две подкласе, Dizelas и Benzinac, које немају велику важност за пројекат, већ служе као демонстрација наслеђивања.

Прво што особа која приступи систему види је листа возила из које може да изабере једно. Листа возила је креирана из једнодимензионалног низа објеката `vozila`. Власник фирме сам врши унос и евиденцију возила у програму. Помоћу **for** петље се аутоматски испишује свако возило из листе уз продружен број, почевши од 1. Избор возила се врши преко уноса са тастатуре.

```

package vozilo;
import java.util.Scanner;

public class Driver2 {

    public static void main(String[] args) {

        Vozilo[] vozila={
new Vozilo("VW_Golf",90,5,50,"dizel"),
new Vozilo("Opel_Astra",100,6,70,"dizel"),
new Vozilo("BMW_320d",120,9,105,"dizel"),
new Vozilo("Citroen_C4",85,8,50,"benzin"),
new Vozilo("Renault_Megane",95,8.5,65,"benzin")};

        while(true){
            int povratak=1;
            System.out.println("Преглед доступних модела:");
            for(int v=0;v<vozila.length;v++){
                System.out.println(v+1 + "." + vozila[v].model);
            }
            System.out.println("Унесите број за листу опција.");
        }
    }
}

```

Програм ради у бесконачној петљи, замишљен да одмах буде доступан за коришћење следећој муштерији, или да послужи последњу муштерију у смислу информација о неком другом возилу. Променљива `povratak` се користи за контролу улажења у петљу, програм преко ње зна када да се врати на почетну листу возила а када да штампа опције за изабрано возило. Контрола изузетака (exceptions) се врши помоћу **if** конструкције, где се избацује грешка о уносу ако је унети број већи од редног броја последњег излистаног возила или мањи од редног броја првог.

```

Scanner in = new Scanner(System.in);
int indeks = in.nextInt();
if(indeks>vozila.length || indeks<1){
    System.out.println("Грешка у уносу.\n");
    continue;
}

```

У случају пролаза у следећи корак, програм штампа доступне опције за изабрано возило. Реализација изабране опције се врши преко **switch** команде, где се сваки број дефинише посебном функцијом, а реализује само онај који је унет.

```

System.out.println("Изабрали сте " + vozila[indeks-1].model + "\n");

while (povratak!=0) {
    System.out.println("Списак опција за " + vozila[indeks-1].model + ":");
    System.out.println("1. Информације о возилу");
    System.out.println("2. Упоредите са другим возилом");
}

```

```
System.out.println("3.Унесите жељену километражу");
System.out.println("4.Изнајмите возило\n");
System.out.println("0.Повратак на почетак");
int unos1=in.nextInt();

switch(unos1){
```

Стандардни (default) случај је грешка у уносу. Прва опција је прост текстуални испис карактеристика изабраног возила. Други представља упоређивање изабраног возила са другим возилом у понуди. Листа возила за упоређивање је реализована тако да се већ изабрано возило не исписује, а да случајни његов избор прикаже грешку у уносу. Трећа опција је генерисање информација од потенцијалног значаја муштерији, у случају да зна колики пут треба да пређе, може сазнати колико времена је за то потребно као и количину горива које ће утрошити у ту сврху. Опција 4 потврђује изнајмљивање изабраног возила и излази из палете опција приказиване за дато возило.

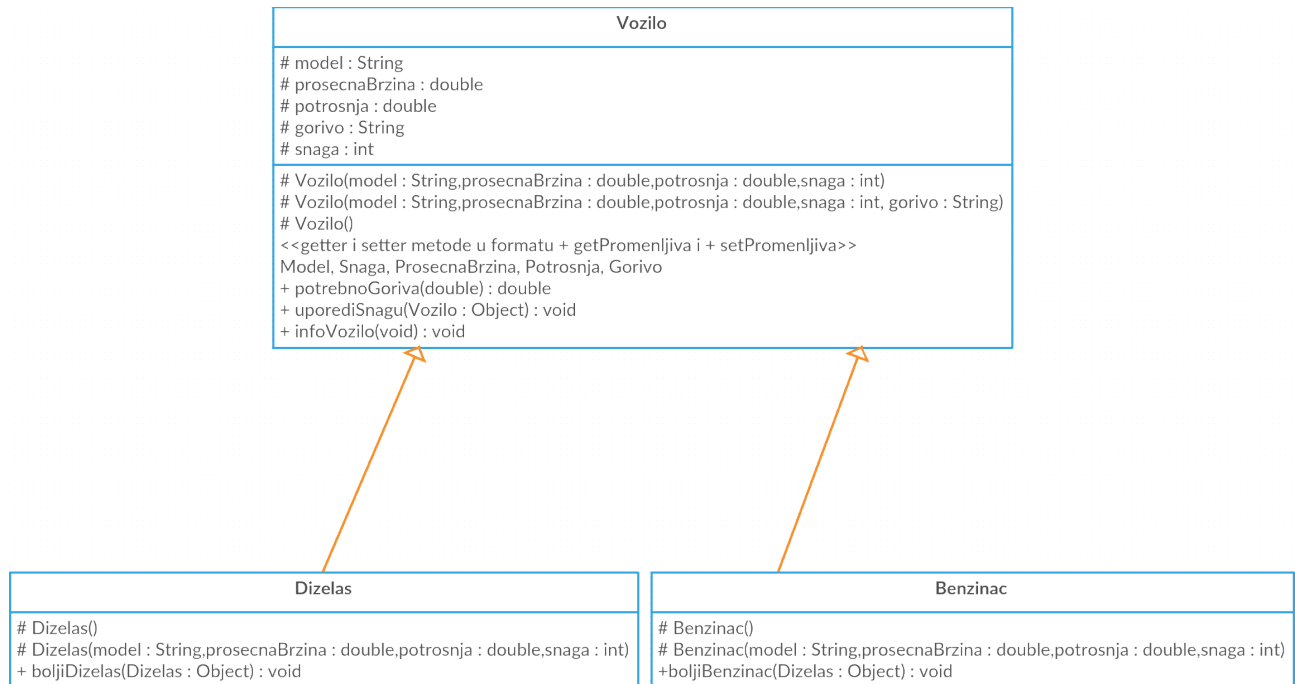
```
case 0: {
    System.out.println("Враћени сте на почетак.\n");
    povratak=0;
    continue;
}
case 1:{
    vozila[indeks-1].infoVozilo();
    break;
}
case 2:{
    System.out.println("Изаберите возило за упоређивање:");
    for (int i=0;i<vozila.length;i++){
        if(i==indeks-1){
            i++;
        }
        System.out.println(i+1 + "." + vozila[i].model);
    }
    int unos2=in.nextInt();
    if(unos2>0 && unos2-1<vozila.length){
        if (indeks-1!=unos2-1){
            vozila[indeks-1].uporediSnagu(vozila[unos2-1]);
            continue;
        }
        else
            System.out.println("Покушали сте да упоредите са истим возилом.\n");
        continue;
    }
    else
        System.out.println("Грешка у уносу!\n");
        continue;
}
case 3:{
    System.out.println("Унесите број km које планирате да пређете:");
```

```
double kilometraza=in.nextDouble();
if (vozila[indeks-1].gorivo=="benzin"){
    System.out.println("За " + kilometraza + "km потребно је " +
vozila[indeks-1].potrebnoGoriva(kilometraza) + "l бензина.");
    System.out.println("Жељену километражу можете прећи за " +
vozila[indeks-1].potrebnoVremena(kilometraza) + "min.\n");
}
else{
    System.out.println("За " + kilometraza + "km потребно је " +
vozila[indeks-1].potrebnoGoriva(kilometraza) + "l дизела.");
    System.out.println("Жељену километражу можете прећи за " +
vozila[indeks-1].potrebnoVremena(kilometraza) + "min.\n");}
    break;

}
case 4:{
    System.out.println("Успешно сте изнајмили возило!\n");
    povratak=0;
    break;
}
default:{
    System.out.println("Грешка у уносу!\n");
    break;
}
}
}
}
```


3.UML дијаграми

3.1.Дијаграм класа



3.2. Use case дијаграм

