

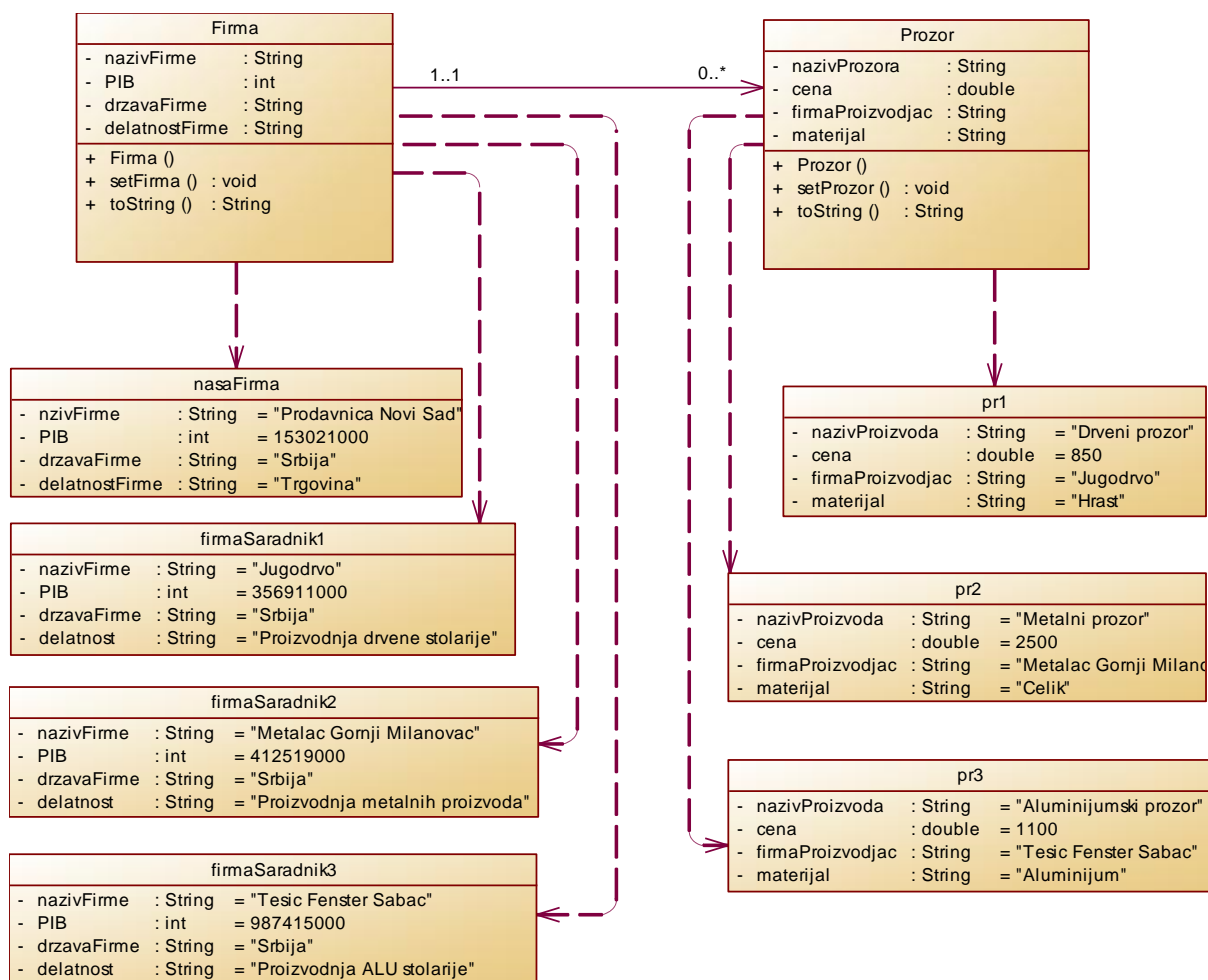
KI104-DZ01-Branislav_Manojlovic_kp11

Zadatak 1:

Napraviti klasni model u **PowerDesigner**-u i implementaciju modela u **Javi** za sledeći primer: Naša firma se bavi proizvodnjom prozora . Prozore delimo na drvene, metalne i aluminijumske. Svaki prozor ima svoju cenu, firmu kao i naziv. O firmama sa kojim sarađujemo obično čuvamo informacije kao što su naziv firme, pib i država firme. Dobijeni model implementirati u Javi.

Rešenje:

Pomoću Power Designer alata, kreiramo UML dijagram, prikazan na slici 1.

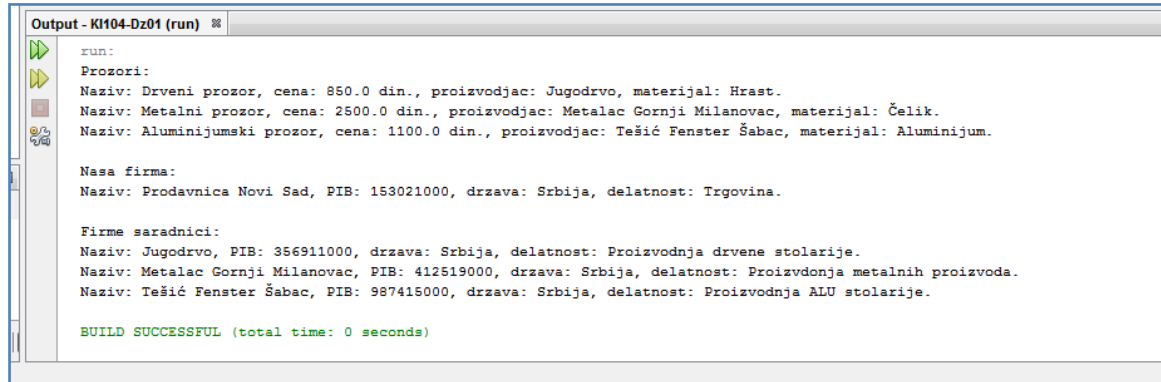


Slika 1. UML dijagram u zadatku 1

PWD projekat sa modelom prikazanim na slici, nalazi se u folderu: **KI104-DZ01-Branislav_Manojlovic_kp11.**, koji prilažem u domaćem radu.

Implementacija u Java programskom jeziku je prikazana u paketu **domacizadatak1**, koji se nalazi u **projektu KI104-DZ01**.

Prilikom pokretanja programa, rešenje koje se dobija ispisano na konzoli je prikazano na slici 2.

The image shows a screenshot of a Java IDE's output window. The title bar reads "Output - KI104-Dz01 (run)". The output text is as follows:

```
run:
Prozori:
Naziv: Drveni prozor, cena: 850.0 din., proizvođjac: Jugodrvo, materijal: Hrast.
Naziv: Metalni prozor, cena: 2500.0 din., proizvođjac: Metalac Gornji Milanovac, materijal: Čelik.
Naziv: Aluminijski prozor, cena: 1100.0 din., proizvođjac: Tešić Fenster Šabac, materijal: Aluminijs.

Nasa firma:
Naziv: Prodavnica Novi Sad, PIB: 153021000, drzava: Srbija, delatnost: Trgovina.

Firme saradnici:
Naziv: Jugodrvo, PIB: 356911000, drzava: Srbija, delatnost: Proizvođnja drvene stolarije.
Naziv: Metalac Gornji Milanovac, PIB: 412519000, drzava: Srbija, delatnost: Proizvođnja metalnih proizvoda.
Naziv: Tešić Fenster Šabac, PIB: 987415000, drzava: Srbija, delatnost: Proizvođnja ALU stolarije.

BUILD SUCCESSFUL (total time: 0 seconds)
```

Slika 2. Ispis rešenja u zadatku 1

Zadatak 2:

Napraviti simulaciju igrice u kojoj se trkaju dva motocikla. Treba napraviti klasu Motocikl i klasu Trka. Svaki Motocikl treba da ima naziv, maksimalnu brzinu kao i ubrzanje. Klasa Trka treba da predstavlja klasu u kojoj se odvija sama trka. Od atributa klasa treba da ima dva motocikla, dužinu staze i, vreme trke, kao i koji je motocikl pobednik na kraju trke. Koristeći while petlju napraviti simulaciju trkanja tako da se vreme povećava za jedan a brzina motocikla ja jednaka njegovom ubrzanju pomnoženom sa vremenom. Paziti da brzina motocikla ne može preći maksimalnu brzinu.

Rešenje:

```
package domacizadatak2;
/** @author Branislav */
public class Motocikl {
/*----- ATRIBUTI -----*/
    String NazivMotora;
    int MaxBrzina;
    double Ubrzanje;
/*----- KONSTRUKTORI -----*/
    public Motocikl() {
    }
/*----- METODE -----*/
    public void setMotocikl(String NazivMotora, int MaxBrzina, double Ubrzanje) {
        this.NazivMotora = NazivMotora;
        this.MaxBrzina = MaxBrzina;
        this.Ubrzanje = Ubrzanje;
    }

    public String getNazivMotora() {
        return NazivMotora;
    }

    public int getMaxBrzina() {
        return MaxBrzina;
    }
    public double getUbrzanje() {
        return Ubrzanje;
    }
    @Override
    public String toString() {
        return " model: " + NazivMotora + ", max. brzina: " + MaxBrzina +
            " km/h, ubrzanje: " + Ubrzanje + '.';
    }
}
```

```

package domacizadatak2;
/** @author Branislav */
public class Trka {
/*----- ATRIBUTI -----*/
    Motocikl Ucesnik1;
    Motocikl Ucesnik2;
    double predjeniPut1;
    double predjeniPut2;
    int vremeTrke = 1;
    Motocikl pobednikTrke;
/*----- KONSTRUKTORI -----*/
    public Trka() {
    }
/*----- METODE -----*/
    public Trka(Motocikl motor1, Motocikl motor2, int vreme) {
        this.Ucesnik1 = motor1;
        this.Ucesnik2 = motor2;

        while(predjeniPut2 < vreme && predjeniPut1 < vreme){
            double brzinaPrvog = vremeTrke * motor1.getUbrzanje();
            if(brzinaPrvog > motor1.getMaxBrzina()){
                brzinaPrvog = motor1.getMaxBrzina();
            }
            double brzinaDrugog = vremeTrke * motor2.getUbrzanje();
            if(brzinaDrugog > motor2.getMaxBrzina()){
                brzinaDrugog = motor2.getMaxBrzina();
            }
            predjeniPut1 += brzinaPrvog;
            predjeniPut2 += brzinaDrugog;
        }
        // Petlja if-else kojom na osnovu ispunjenosti zadatog uslova
        // odredjujem pobednika
        if(predjeniPut1>predjeniPut2){
            pobednikTrke = motor1;
        }else if(predjeniPut1<predjeniPut2){
            pobednikTrke = motor2;
        }else {
            pobednikTrke = null;
        }
    }

    public Motocikl getPobednik() {
        return pobednikTrke;
    }
}

```

```

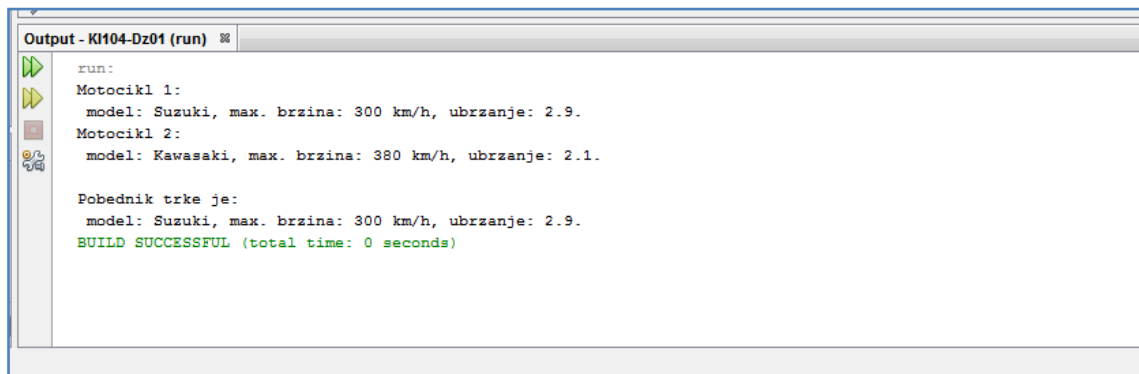
package domacizadatak2;
/** @author Branislav */
public class Main {
    public static void main(String[] args) {
        // Kreiram objekte tipa Motocikl i zadajem vrednosti njihovih atributa
        System.out.println("Motocikl 1:");
        Motocikl motor1 = new Motocikl();
        motor1.setMotocikl("Suzuki", 300, 2.9);
        System.out.println(motor1.toString());

        System.out.println("Motocikl 2:");
        Motocikl motor2 = new Motocikl();
        motor2.setMotocikl("Kawasaki", 320, 2.1);
        System.out.println(motor2.toString());

        // Kreiram objekat Trka i kao argumente zadajem parametre koji se
        // odnose na motor 1 i motor 2 koji su ucesnici u trci.
        Trka trka = new Trka(motor1, motor2, 20000);
        System.out.println("\nPobednik trke je: ");
        System.out.println(trka.getPobednik());
    }
}

```

Nakon pokretanja programa, dobijam rešenje prikazano na slici 3.



Slika 3. Ispis rešenja u zadatku 2

Napomena: Pored ovog tekstualnog fajla, prilažem i kompletan projekat u zipovanom fajlu: **KI104-Dz01.zip**.

MSc Branislav Manojlović