

OBJEKTNO-ORIJENTISANI DIZAJN SOFTVERA

-IV čas računskih vježbi-

1. Napisati JAVA program kojim se vodi statistika troškova jednoga preduzeća. Preduzeće ima dva tipa izdataka, izdatke na plate radnika i izdatke na materijalne troškove. Svaki pojedinačni izdatak potrebno je bilježiti u tabeli koja posjeduje tri kolone: **Tip troška**, **Svrha troška** i **Iznos**. Za potrebe realizacije programa konstruisati abstraktnu klasu Radnik i javnu klasu MaterijalniTrosak. Iz prve izvesti klase Rukovodilac i Fizikalac. Klasa Rukovodilac ima sledeće podatke članove: ime i prezime, koeficijent, minimalnu cijenu rada, funkcijski dodatak i premiju, dok su podaci članovi klase Fizikalac: ime i prezime, koeficijent, minimalna cijenu rada, norma i ostvareni broj sati rada. Materijalni troškovi su opisani sa sledećim podacima: opis troška, količina i jedinična cijena. Sve klase treba da posjeduju metodu iznos() koja će vraćati iznos troška. Plata Rukovodioca se obračunava po formuli: $plata = koeficijent * minimalna_cijena_rada * (1 + premija + funkcijski_dodatak)$, dok se plata Fizikalca obračunava kao: $plata = koeficijent * minimalna_cijena_rada * (ostvareni_broj_sati < norma) ? (0.8) : (1 + (ostvareni_broj_sati - norma) / 100)$.

```
// Interfejs Troskovi
public interface Troskovi
{
    public float iznos();
    public String tipTroska();
    public String svrhaTroska();
}

// Realizacija abstraktne klase Radnik
public abstract class Radnik implements Troskovi
{
    private String imePrezime;
    private static float minCijenaRada;
    private float koeficijent;
    Radnik()
    {
        this("",0);
    }
    Radnik(String i, float k)
    {
        imePrezime=i;
        koeficijent=k;
    }
    public String getImePrezime()
    {
        return imePrezime;
    }
    public void setImePrezime(String imePrezime)
    {
        this.imePrezime = imePrezime;
    }
    public static float getMinCijenaRada()
    {
        return minCijenaRada;
    }
    public static void setMinCijenaRada(float minCijenaRada)
    {
        Radnik.minCijenaRada = minCijenaRada;
    }
    public float getKoeficijent()
    {
        return koeficijent;
    }
    public void setKoeficijent(float koeficijent)
    {
        this.koeficijent = koeficijent;
    }
    @Override
    public String tipTroska()
    {

```

```

    return "Plata";
}
}

// Realizacija klase MaterijalniTrosak
public class MaterijalniTrosak implements Troskovi
{
    private String opisTroska;
    private float kolicina;
    private float jedinicnaCijena;
    MaterijalniTrosak()
    {
        this("",0,0);
    }
    MaterijalniTrosak(String Opis, float kol, float cijena)
    {
        this.opisTroska=Opis;
        this.kolicina=kol;
        this.jedinicnaCijena=cijena;
    }
    public String getOpisTroska()
    {
        return opisTroska;
    }
    public void setOpisTroska(String opisTroska)
    {
        this.opisTroska = opisTroska;
    }
    public float getKolicina()
    {
        return kolicina;
    }
    public void setKolicina(float kolicina)
    {
        this.kolicina = kolicina;
    }
    public float getJedinicnaCijena()
    {
        return jedinicnaCijena;
    }
    public void setJedinicnaCijena(float jedinicnaCijena)
    {
        this.jedinicnaCijena = jedinicnaCijena;
    }
    @Override
    public float iznos()
    {
        return kolicina*jedinicnaCijena;
    }
    @Override
    public String tipTroska()
    {
        return "Materijalni troškovi";
    }
    @Override
    public String svrhaTroska()
    {
        return opisTroska;
    }
}

// Realizacija klase Rukovodilac
public class Rukovodilac extends Radnik
{
    private float funkcijskiDodatak;
    private float premija;
    Rukovodilac()
    {

```

```

    this("",0,0,0);
}
Rukovodilac(String imePrezime,float koef,float dodatak, float prem)
{
    super(imePrezime,koef);
    this.funkcijskiDodatak=dodatak;
    this.premija=prem;
}
public float getFunkcijskiDodatak()
{
    return funkcijskiDodatak;
}
public void setFunkcijskiDodatak(float funkcijskiDodatak)
{
    this.funkcijskiDodatak = funkcijskiDodatak;
}
public float getPremija()
{
    return premija;
}
public void setPremija(float premija)
{
    this.premija = premija;
}
@Override
public float iznos()
{
    float plata=super.getKoeficijent()*super.getMinCijenaRada();
    return plata*(1+funkcijskiDodatak+premija);
}
@Override
public String svrhaTroska()
{
    return "Plata za rukovodioca: " + super.getImePrezime();
}
}

// Realizacija klase Radnik
public class Fizikalac extends Radnik
{
    private int norma;
    private int ostvareniBrojSati;
    Fizikalac()
    {
        this("",0,0,0);
    }
    Fizikalac(String ime, float koef, int norm, int sati)
    {
        super(ime,koef);
        norma=norm;
        ostvareniBrojSati=sati;
    }
    @Override
    public float iznos()
    {
        float plata;
        plata=super.getKoeficijent()*super.getMinCijenaRada();
        if (ostvareniBrojSati<norma)
            plata=plata*(float)0.8;
        else
            plata=plata*(1+(ostvareniBrojSati-norma)/100);
        return plata;
    }
    @Override
    public String svrhaTroska()
    {
        return "Plata za fizikalca: " + super.getImePrezime();
    }
}

```

```

}

// Realizacija klase StatistikaTroskova
import java.util.Scanner;
import java.io.*;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFCellStyle;

public class StatistikaTroskova
{
    public static void main(String[] args)
    {
        int i,br,iT=0,norma,sati;
        String ime,prezime;
        float koef, dodatak, premija;
        Scanner sc=new Scanner (System.in);
        System.out.println("Unesi minimalnu cijenu rada");
        Radnik.setMinCijenaRada(sc.nextFloat());
        System.out.println("Unesi broj ukupnih unosa troškova: ");
        br=sc.nextInt();
        Troskovi T[] = new Troskovi [br];
        System.out.println("Unesi broj rukovodilaca");
        br=sc.nextInt();
        System.out.println("Unesi rukovodioce u sledećem formatu:");
        System.out.println("Ime Prezime Koeficijent Dodatak Premija");
        for (i=0;i<br;i++)
        {
            System.out.printf("Radnik broj %d\n",i+1);
            ime=sc.next();
            prezime=sc.next();
            ime=ime+" "+prezime;
            koef=sc.nextFloat();
            dodatak=sc.nextFloat();
            premija=sc.nextFloat();
            T[iT++]=new Rukovodilac(ime,koef,dodatak,premija);
        }
        System.out.println("Unesi broj fizikalaca");
        br=sc.nextInt();
        System.out.println("Unesi fizikalce u sledećem formatu:");
        System.out.println("Ime Prezime Koeficijent Norma Broj ostvarenih sati");
        for (i=0;i<br;i++)
        {
            System.out.printf("Fizikalac broj %d\n",i+1);
            ime=sc.next();
            prezime=sc.next();
            ime=ime+" "+prezime;
            koef=sc.nextFloat();
            norma=sc.nextInt();
            sati=sc.nextInt();
            T[iT++]=new Fizikalac(ime,koef,norma,sati);
        }
        System.out.println("Unesi broj materijalnih troškova");
        br=sc.nextInt();
        System.out.println("Unesi materijalne troškove u sledećem formatu:");
        System.out.println("Opis troška Količina Cijena");
        String opis;
        float kolicina,cijena;
        for (i=0;i<br;i++)
        {
            System.out.printf("Trošak broj %d\n",i+1);
            opis=sc.next();
            kolicina=sc.nextFloat();
            cijena=sc.nextFloat();
            T[iT++]=new MaterijalniTrosak(opis,kolicina,cijena);
        }
    }
}

```

```
System.out.println("Tip troška \t Svrha troška \t Iznos");
for (i=0;i<iT;i++)
{
    // Kreiranje reda
    System.out.println(T[i].tipTroska() + T[i].svrhaTroska() + T[i].iznos());
}
}
}
```