

### LABORATORIJSKE VJEŽBE 3

1. Napraviti klasu **Radnik** koja ima privatne atribute: **ime** (String), **grs** (br godina radnog staža - int), **koef** (koeficijent - double) i **p\_rad** (broj sati prekovremenog rada - int). Pored ovih podataka, klasa posjeduje i javnu statičku promjenljivu **min** (minimalna cijena rada -double), kao i metode **UnesiPodatke** (služi za inicijalizaciju objekta) i **plata** koja računa platu radnika po sljedećoj formuli:  $plata = koef * min + 0.01 * grs * (koef * min) + p\_rad * 10$ . U funkciji **main**, sa tastature unijeti minimalnu cijenu rada. (Kako je to moguće, kada nismo kreirali objekat?) Nakon toga, deklarirati i inicijalizovati dva objekta tipa Radnik. Podatke o radnicima je potrebno unijeti sa tastature i proslijediti ih metodi UnesiPodatke. (Da li unosimo i podatak **min**?). Plate radnika prikazati na standardnom izlazu.

```
import java.util.Scanner;

public class Radnik {

    public static double min;
    private String ime;
    private int grs, p_rad;
    private double koef;

    void UnesiPodatke(String ime, int grs, int p_rad, double koef) {
        this.ime = ime;
        this.grs = grs;
        this.koef = koef;
        this.p_rad = p_rad;
    }

    public double plata() {
        return koef * min + 0.01 * grs * (koef * min) + p_rad * 10;
    }

    public static void main(String[] args) {
        Scanner unos = new Scanner(System.in);
        System.out.println("Unesi minimalnu cijenu rada:");
        min = unos.nextDouble();
        Radnik f1 = new Radnik(), f2 = new Radnik();
        System.out.print("Unesi ime 1. radnika:");
        f1.ime = unos.next();
        System.out.print("Unesi redom podatke za 1. radnika(grs, p_rad i koef):");
        f1.grs = unos.nextInt();
        f1.p_rad = unos.nextInt();
        f1.koef = unos.nextDouble();
        f1.UnesiPodatke(f1.ime, f1.grs, f1.p_rad, f1.koef);
        System.out.print("Unesi ime 2. radnika:");
        f2.ime = unos.next();
        System.out.print("Unesi redom podatke za 2. radnika(grs, p_rad i koef): ");
        f2.grs = unos.nextInt();
        f2.p_rad = unos.nextInt();
        f2.koef = unos.nextDouble();
        f2.UnesiPodatke(f2.ime, f2.grs, f2.p_rad, f2.koef);
        System.out.print("Plata I radnika iznosi: " + f1.plata() + "eura.");
        System.out.print("Plata II radnika iznosi: " + f2.plata() + "eura.");
    }
}
```

2. Kreirati klasu **Racunar** koja ima javni podatak **clan godina** (za godinu proizvodnje racunara). Zatim kreirati klasu nasljednicu klase **Racunar** pod nazivom **Laptop**. Klasa **Laptop** treba da ima privatne podatke **clanove tezina** i **model**, kao i metode za vraćanje i podešavanje modela. U glavnom programu kreirati

objekat klase Laptop i postaviti model na "HP" a godinu na 2019. Prikazati model i godinu proizvodnje racunara.

```
public class Racunar {
    int godina;

    public static void main(String[] args) {
        Laptop l = new Laptop();
        l.setModel("HP");
        l.godina = 2019;
        System.out.println("Model " + l.getModel() + ", " + "Godina proizvodnje" +
l.godina);
    }
}

class Laptop extends Racunar {
    private String model;
    private double tezina;

    public String getModel() {
        return this.model;
    }

    public void setModel(String a) {
        this.model = a;
    }
}
```

3. Kreirati klasu ArrayMin koja od korisnika traži unos duzine niza, a nakon toga unos i elemenata niza. Program određuje na kojoj poziciji se nalazi najmanji element niza koji smo unijeli, kao i njegovu vrijednost. Zatim program štampa obavještenje sa pozicijom minimalnog elementa i njegovom vrijednošću.

```
import java.util.Scanner;

public class ArrayMin {

    public static int najmanji(int[] array, int size) {
        //pretstavimo da je najmanji clan niza prvi clan
        //a kasnije uporedjujemo svaki sledeci sa minimumom.
        int pozicija_min = 0;
        int min = array[0];
        for (int i = 1; i < size; i++) {
            //Kada pronadjemo clan niza koji je manji nego dotadasnji minimum
            //onda taj clan niza proglašavamo za novi minimum
            if (array[i] < min) {
                min = array[i];
                pozicija_min = i; //pamtimo poziciju najmanjeg clana niza.
            }
        }
        return pozicija_min;
    }

    public static void main(String[] args) {
        System.out.print("Unesite duzinu niza: ");
        Scanner duzina = new Scanner(System.in);
        int size = duzina.nextInt();
        System.out.print("Unesite " + " " + size + " " + "elemenata niza");
        Scanner unos = new Scanner(System.in);
        //formiramo niz i smijestamo njegove clanove u niz array
        int array[] = new int[size];
    }
}
```

```

        for (int i = 0; i < array.length; i++) {
            array[i] = unos.nextInt();
        }
        // Poziva se metoda najmanji i njoj se kao
        //argumenti prosledjuju niz i duzina niza.
        System.out.println(najmanji(array, size));
    }
}

```

4. Kreirati klasu *Stanje* koja ima podatke članove *stanje\_racunar* i *stanje\_laptop*, konstruktore, metodu za povećanje stanja računara i laptopova (broj računara i laptopova se ovom metodom istovremeno uvećava), metodu za umanjnje stanja računara i metodu za umanjnje stanja laptopova. Klasa treba da posjeduje metode za vraćanje trenutnog stanja računara i laptopova. Ukoliko je vrijednost argumenta metoda kojima se umanjuje stanje veća od raspoloživog stanja, metoda izbaca izuzetak *PrevelikaPotraznja* sa odgovarajućom porukom korisniku i brojem koji označava koliko komada računara/laptopova nedostaje do traženog iznosa. Klasa *PrevelikaPotraznja* je nasljednica klase *Exception*. U glavnom programu inicijalizovati objekat klase *Stanje*, demonstrirati povećanje stanja objekta i demonstrirati pozive funkcija za umanjnje stanja.

```

public class Magacin {

    public static void main(String[] args) {
        Stanje c = new Stanje();
        System.out.println("\nPovecaj racunar");
        c.povecaj(500, 1500);
        try {
            System.out.println("\nUmanji racunare");
            c.umanjiRacunar(100);
            System.out.println("\nUmanji laptopove");
            c.umanjiLaptop(20);
        } catch (PrevelikaPotraznja e) {
            System.out.println("Nemate dovoljno sredstava, fali " + e.getKolicina());
        }
    }
}

class PrevelikaPotraznja extends Exception {

    private int kolicina;

    public PrevelikaPotraznja(int kolicina) {
        this.kolicina = kolicina;
    }

    public int getKolicina() {
        return kolicina;
    }
}

class Stanje {

    private int stanje_racunar;
    private int stanje_laptop;

    public Stanje() {
        this.stanje_racunar = 0;
        this.stanje_laptop = 0;
    }

    public Stanje(int stanje_racunar, int stanje_laptop) {

```

```
        this.stanje_racunar = stanje_racunar;
        this.stanje_laptop = stanje_laptop;
    }

    public void povecaj(int kol_rac, int kol_lap) {
        stanje_racunar += kol_rac;
        stanje_laptop += kol_lap;
    }

    public void umanjiRacunar(int kol_rac) throws PrevelikaPotraznja {
        if (kol_rac <= stanje_racunar) {
            stanje_racunar -= kol_rac;
        } else {
            int razlika = kol_rac - stanje_racunar;
            throw new PrevelikaPotraznja(razlika);
        }
    }

    public void umanjiLaptop(int kol_lap) throws PrevelikaPotraznja {
        if (kol_lap <= stanje_laptop) {
            stanje_laptop -= kol_lap;
        } else {
            int razlika = kol_lap - stanje_laptop;
            throw new PrevelikaPotraznja(razlika);
        }
    }

    public int getStanjeRacunar() {
        return stanje_racunar;
    }

    public int getStanjeLaptop() {
        return stanje_laptop;
    }
}
```