

## Domaći zadatak 02 – Branislav Manojlović kp11 KI301-DZ02

### Tekst zadatka

Odabrati proizvoljnu aplikaciju i izvršiti sledeće:

1. Prikazati strukturni način programiranja,
2. Opisati na koji način je moguće prepoznati strukturni način programiranja,
3. Prikazati upotrebu return instrukcije u aplikaciji.

### Rešenje:

- 1) Strukturni način programiranja je prikazan kroz primer izgleda jednog struktuiranog programa:

```
package projektnizadatak;
import java.util.Scanner;
/**
 *
 * @author Branislav
 */
public class Zadatak {

    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        meni();
    }
    //metoda za poziv i izracunavanje matematickih funkcija: sqrt, ceil, exp
    public static void matematFunkcije(){
        String izbor2;
        System.out.println("Odaberite jednu ot tri matematicke fukcije:");
        System.out.println(" - sqrt");
        System.out.println(" - ceil");
        System.out.println(" - exp");
        izbor2 = sc.nextLine();

        if(izbor2.equals("sqrt")){
            double x;
            System.out.println("Kvadratni koren");
            System.out.println("Unesite broj X:");
            x = sc.nextDouble();
            System.out.println("Vas rezultat je: sqrt X = " + Math.sqrt(x));
        }else if( izbor2.equals("ceil")){
            double x;
            System.out.println("Zaokruzivanje broja");
            System.out.println("Unesite broj X:");
            x = sc.nextDouble();
            System.out.println("Vas rezultat je: ceil X = " + Math.ceil(x));
        }else if(izbor2.equals("exp")){
```

```

        double x;
        System.out.println("Eksponecijalna vrednost");
        System.out.println("Unesite broj X:");
        x = sc.nextDouble();
        System.out.println("Vas rezultat je: exp X = " + Math.exp(x));
    }else
        System.out.println("Izabrali se nepostojecu opciju pokusajte ponovo.");
    }
    //metoda koja nudi opciju i omogucava poziv drugih metoda za izracunavanje binomnog
    koeficijenta
    public static void binomniKoeficijent(){
        String opcija3;
        int n, k;
        System.out.println("Odaberite zeljenu opciju:");
        System.out.println("a) Računanje binomnog koeficijenta");
        opcija3 = sc.nextLine();
        if(opcija3.equals("a")){
            System.out.println("Unesite brojeve N i K za koje zelite da izracunate binomni
            koeficijent.");
            System.out.println("Unesite N:");
            n = sc.nextInt();
            System.out.println("Unesite K:");
            k = sc.nextInt();
            if(k<0 || k>n){
                System.out.println("Resenje = 0");
            }else {
                //izracunavanjeBinKoefic(n, k);
                System.out.println("Vas rezultat je: " + izracunavanjeBinKoefic(n, k));
            }
        }else {
            System.out.println("Odabrali ste nepostojecu opciju, jer jedina opcija je pod a");
        }
    }
    //metoda za izracunavanje binomnog koeficijenta
    public static double izracunavanjeBinKoefic(int n, int k){
        double rezBinKoef;
        rezBinKoef = faktorijal(n)/(faktorijal(k)*faktorijal(n-k));
        return rezBinKoef;
    }
    //metoda koja racuna faktorijal broja, treba mi za izracunavanje binomnog koeficijenta
    public static int faktorijal(int m){
        int rez=1;
        for(int i=1; i<=m; i++){
            rez = rez * i;
        }
        return rez;
    }
    //metoda koja vrsi obradu niza
    public static void obradaNiza(){
        int n;

```

```

char opcija4;
System.out.println("Unesite ceo broj N koji predstavlja veličinu niza:");
n = sc.nextInt();
int[] mojNiz = new int[n];
for(int i=0; i<n; i++){
    System.out.println("Unesite element " + i);
    mojNiz[i] = sc.nextInt();
}
System.out.println("Šta želite odrediti za vaš niz?");
System.out.println("a) Aritmetička sredina niza");
System.out.println("b) Da li je niz rastući?");
opcija4 = sc.next().charAt(0);
if(opcija4 == 'a'){
    double rez = sumaNiza(mojNiz)/mojNiz.length;
    System.out.println("Vas rezultat je: " + rez);
}else if(opcija4 == 'b'){
    //for petlja za proveru da li je niz rastuci
    boolean rastuci = false;
    for(int i=0; i<mojNiz.length-1; i++){
        if(mojNiz[i] < mojNiz[i+1]){
            rastuci = true;
        }else {
            rastuci = false;
        }
    }
    System.out.println("Vas rezultat je - Niz je rastuci: " + rastuci);
}else
    System.out.println("Odabrali ste nepostojecu opciju.");
}
//metoda koja izracunava sumu niza
public static int sumaNiza(int[] mojNiz){
    int suma=0;
    for(int i=0; i<mojNiz.length; i++){
        suma += mojNiz[i];
    }
    return suma;
}
//metoda za obradu matrica
public static void obradaMatrica(){
    int N, M;
    System.out.println("Unesite dimenziju N: ");
    N = sc.nextInt();
    System.out.println("Unesite dimenziju M: ");
    M = sc.nextInt();
    //kreiram matricu A sa random vrednostima
    int[][] A = new int[N][M];
    for(int i=0; i<N; i++){
        for(int j=0; j<M; j++){
            System.out.println("Unesite element na poziciji: [" + i + ", " + j + "]");
            A[i][j] = sc.nextInt();
        }
    }
}

```

```

    }
}
char opcija5;
System.out.println("Odaberite zeljenu opciju:");
System.out.println("a) Suma elemenata iznad glavne dijagonale");
opcija5 = sc.next().charAt(0);
if(opcija5 == 'a'){
    int sumaElemenata=0;
    //dupla for petlja koja izracunava sumu elemenata iznad glavne dijagonale
    for(int i=0; i<A.length; i++){
        for(int j=0; j<A[i].length; j++){
            if(i < j){
                sumaElemenata = sumaElemenata + A[i][j];
            }
        }
    }
    System.out.println("Vas rezultat je: SUMA = " +sumaElemenata);
}else
    System.out.println("Odabrali ste nepostojecu opciju. Jedina opcija je opcija a.");
}
//metoda za obradu stringova
public static void obradaStringova(){
    char odgovor6;
    String a, b;
    System.out.println("Unesite string a:");
    a = sc.nextLine();
    System.out.println("Unesite string b:");
    b = sc.nextLine();

    System.out.println("Odaberite zeljenu opciju:");
    System.out.println("a) Da li su zagrade u strinovima uparene?");

    odgovor6 = sc.next().charAt(0);
    if(odgovor6 == 'a'){
        boolean odg1 = a.contains("("), odg2 = a.contains(")");
        boolean odg3 = b.contains("("), odg4 = b.contains(")");
        for(int i=0; i<a.length(); i++){
            if(odg1 == true && odg2 == true && odg3 == true && odg4 == true){
                System.out.println("Zagrade su uparene");
            }else if(odg1 == true || odg2 == true || odg3 == true || odg4 == true){
                System.out.println("Zagrada ima ali nisu uparene");
            }else
                System.out.println("Nema zagrada u stringovima");
        }
    }else
        System.out.println("Odabrali ste nepostojecu opciju. Jedina opcija je opcija a.");
}

//metoda koja nudi meni-opcije u programu
public static void meni(){

```

```

int opcija;
System.out.println("Odaberite jednu od sledećih opcija");
System.out.println("1. MATEMATIČKE FUNKCIJE");
System.out.println("2. RAZNI ZADACI");
System.out.println("3. UNOS I OBRADA NIZOVA");
System.out.println("4. UNOS I OBRADA MATRICA");
System.out.println("5. UNOS I OBRADA STRINGOVA");
System.out.println("6. PREKID RADA");
opcija = sc.nextInt();

switch (opcija){
    case 1:
        System.out.println("1. MATEMATIČKE FUNKCIJE");
        matematFunkcije();
        break;
    case 2:
        System.out.println("2. RAZNI ZADACI");
        binomniKoeficijent();
        break;
    case 3:
        System.out.println("3. UNOS I OBRADA NIZOVA");
        obradaNiza();
        break;
    case 4:
        System.out.println("4. UNOS I OBRADA MATRICA");
        obradaMatrica();
        break;
    case 5:
        System.out.println("5. UNOS I OBRADA STRINGOVA");
        obradaStringova();
        break;
    case 6:
        System.out.println("Kraj programa!");
        System.out.println("Hvala vam sto ste koristili nas program.");
        break;
    default:
        System.out.println("Izabrali ste nepostojecu opciju. Pokusajte ponovo.");
}
}
}

```

- 2) Strukturni način programiranja je moguće prepoznati po tome što potprogrami i klase u ovakvim programima imaju komponente kao što su:
- sekvencijalni blokovi instrukcija,
  - selektivne instrukcije, tj. **if**, **if-else** i/ili **switch** instrukcije,
  - petljaste instrukcije, tj. **for** i **while** instrukcije.

Glavno pravilo strukturnog programiranja je da svaki potprogram mora da se sastoji od tri vrste instrukcija (sekvencijalne, selektivne i petljaste instrukcije). Pri tome druge kontrolne strukture poput: **break**, **continue**, **return**, **throw**, **catch** i slične treba izbegavati.

- 3) Primer upotrebe **return** instrukcije u aplikaciji:

```
@Override
public String toString() {
    return "Naziv: " + nazivProzora + ", cena: " + cena +
        " din., proizvođač: " + firmaProizvođač + ", materijal: " + materijal + '.';
}
```

Često se **return** instrukcija upotrebljava u metodama koje kao rezultat treba da daju/vrate neku vrednost. U ovoj metodi (**toString()**) kao vrednost vraća se **String**.

**MSc Branislav Manojlović**  
**Novi Sad 10.06.2018.**