

## Projektni\_Zadatak\_Branislav\_Manojlovic\_kp11

### Zadatak:

Napisati algoritam koji pomoću funkcije (potprograma) koja računa sumu brojeva od 1 - n izračunava formulu.

$$F = (1+2+ \dots + n)(1+2+ \dots + m)/(1+2+\dots+n*m).$$

### Rešenje:

#### 1) Analiza

##### Opis problema

Problem koji treba da rešim u okviru datog projektnog zadatka je izračunavanje vrednosti promenljive **F**. Vrednost promenljive **F** se izračunava pomoću navedene formule.

Ulazni podaci koji su mi potrebni da bih izračunao vrednost promenljive **F** i time rešio projektni zadatak su:

- Broj **n**,
- Broj **m**.

Ovo su podaci čiji se unos zahteva od strane korisnika, tj. koji se unose u program. Ovi podaci mogu da budu tipa **int**.

Na osnovu unetih vrednosti za **n** i **m** izračunavam dve sume i to:

- $\text{SumN} = 1 + 2 + \dots + n$
- $\text{SumM} = 1 + 2 + \dots + m$

Izračunavanje ove dve sume smatram kao da su mi prva dva potprograma unutar velikog zajedničkog programa, tj. oni će mi dati prva dva podalgoritma u okviru glavnog podalgoritma.

Sledeći potreban podatak mi je proizvod unetih brojeva **n** i **m**, koji obeležavam sa **k**. Izračunavam ga kao  $k = n * m$ . Pomoću broja **k** izračunaću treću sumu:

- $\text{SumK} = 1 + 2 + \dots + k$

Ova kalkulacija **SumK** mi predstavlja treći podalgoritam.

Poslednje što izračunavam je zapravo glavni problem u programu a to je **F**. Dakle, ovde konkretno primenjujem zadatu formulu za **F** i izračunavam krajnje rešenje problema. Takođe, treba napomenuti da ovde mora da se izvrši kastovanje promenljivih **SumN**, **SumM** i **SumK** u tip float ili double, da bih dobio decimalan broj i osigurao tačnost rezultata.

### Provere

Osnovna provera koju moram da izvršim je ona koja se odnosi na vrednosti proizvoda promenljivih  $n$  i  $m$ . Odnosno u mom zadatku ja sam ovaj proizvod obeležio sa  $k$ .

Vrši se provera da li je  $k = -1$ ? Ako jeste, mora se tražiti ponovni unos za  $n$  i  $m$ , jer ako bi ostalo kao što je uneto, onda bi imenilac u jednačini za  $F$  bio 0 što nije moguće, jer deljenje sa nulom nema smisla i ne bi se dobio racionalan broj.

### 2) Glavni algoritam

#### Pseudokod:

**korak 1:** START

**korak 2:** Unos broja  $n$  (input  $n$ ),

**korak 3:** Unos broja  $m$  (input  $m$ ),

**korak 4:** Poziva se potprogram *IzracunavanjeSumN*, za promenljivu  $n$ , što znači da se izračunava promenljiva  $SumN$ , primenom formule:  $SumN = 1 + 2 + \dots + n$ ,

**korak 5:** Poziva se potprogram *IzracunavanjeSumM*, za promenljivu  $m$ , što znači da se izračunava promenljiva  $SumM$ , primenom formule:  $SumM = 1 + 2 + \dots + m$ ,

**korak 6:** Izračunava se proizvod  $k = n * m$ ,

**korak 7:** Proverava se da li je  $k = -1$ ?

Ako DA, sledi ispis poruke da je „imenilac jednak 0“ i povratak na korak 2,

Ako NE, ide se dalje na korak 8,

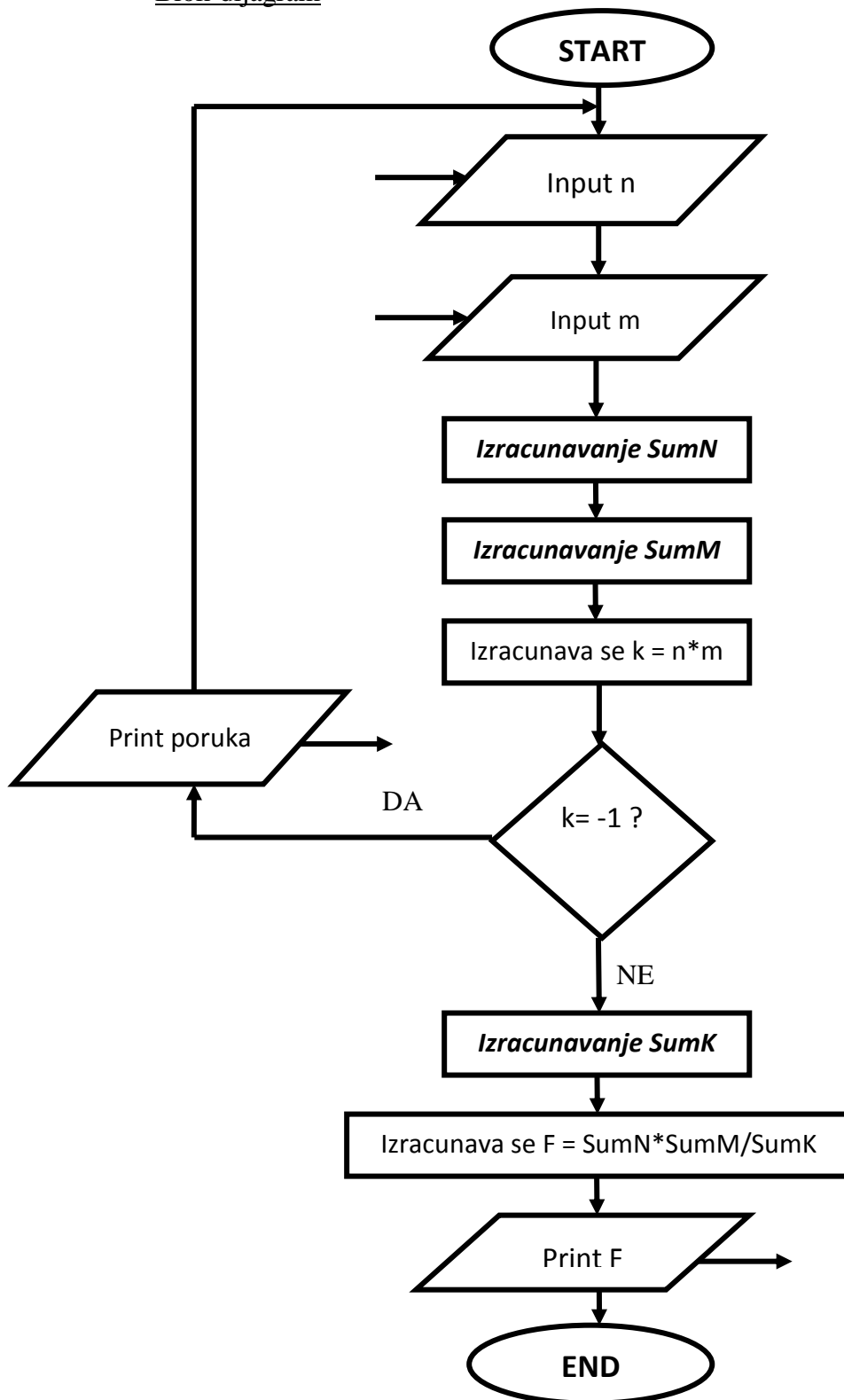
**korak 8:** Poziva se potprogram *IzračunavanjeSumK*, za promenljivu  $k$ , što znači da se izračunava promenljiva  $SumK$ , primenom formule:  $SumK = 1 + 2 + \dots + k$ ,

**korak 9:** Izračunava se  $F$  primenom formule:  $F = SumN * SumM / SumK$ ,

**korak 10:** Ispisuje se rezultat  $F$ ,

**korak 11:** END

Blok-dijagram



### 3) Podalgoritam *IzracunavanjeSumN*

#### Pseudokod

korak 1: START

korak 2: Inicijalizujem promenljivu n tipa int, kao korisnikov unos (npr. 5),

korak 3: Inicijalizujem promenljivu SumN, tipa int i setujem joj vrednost na 0.

korak 4: Kreiram for petlju za prolazak kroz niz brojeva od 1 do n,  
for (int i = 1; i <= n; i++)

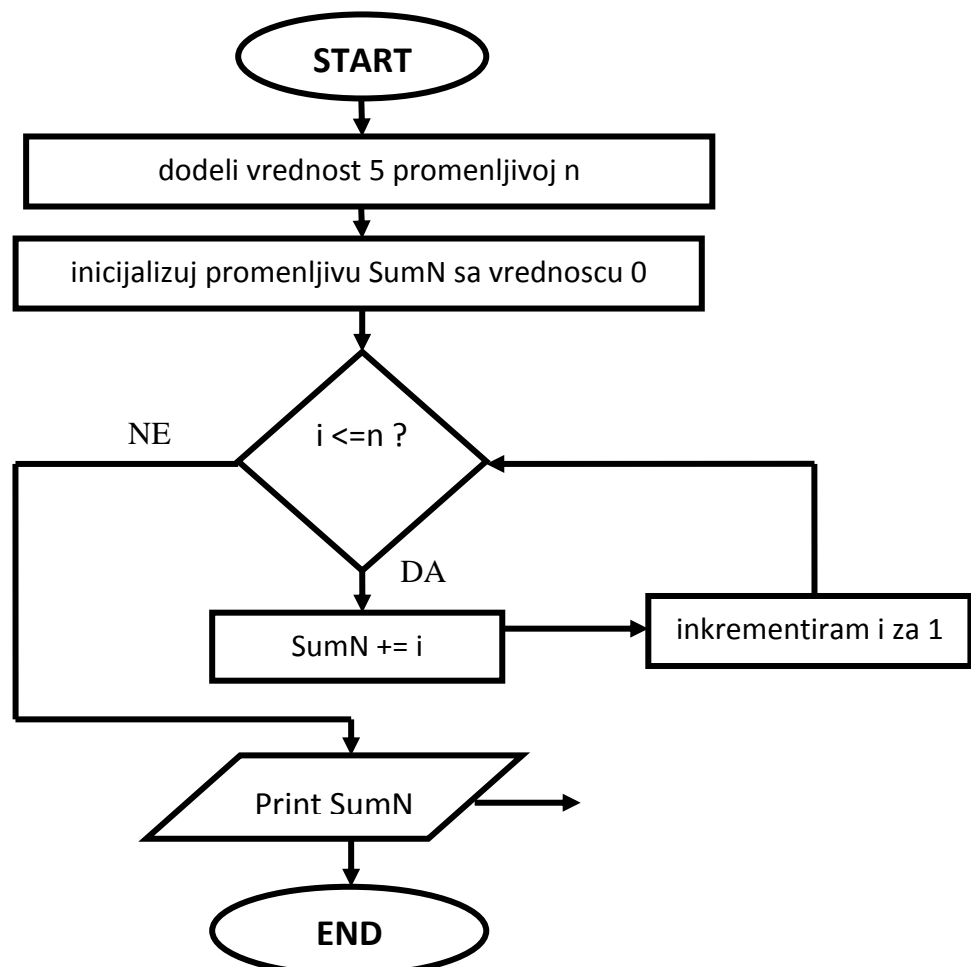
korak 5: Unutar for petlje inkrementiram SumN za **i**, tj. primenjujem formulu:  
 $SumN = 1 + 2 + \dots + n$

korak 6: Zatvaram for petlju,

korak 7: Ispis vrednosti promenljive SumN, pomoću Print SumN,

korak 8: END

#### Blok-dijagram za *IzracunavanjeSumN*



## Projektni zadatak

### 4) Testiranje

Testiranje algoritma vršim za vrednosti  $n=2$  i  $m=3$ . Prikaz test koraka zajedno sa njihovim opisom i očekivanim i stvarno dobijenim rezultatima prikazan je u sledećoj tabeli.

Tabela 1: Prikaz test koraka

Rb.	Test Step Name	Test Step Description	Expected Result	Actual Result	Status [Pass/Failed]
1	Start	Pokrećem program	Program pokrenut	Program pokrenut	Pass
2	Input n	Unosim $n = 2$	Promenljiva $n$ dobila vrednost 2	Promenljiva $n$ dobila vrednost 2	Pass
3	Input m	Unosim $m = 3$	Promenljiva $m$ dobila vrednost 3	Promenljiva $m$ dobila vrednost 3	Pass
4	Call Method	Pozivam IzracunavanjeSumN za $n = 2$ gde se vrši izracunavanje $SumN = 1+2$	$SumN = 3$	Dobija se $SumN = 3$	Pass
5	Call Method	Pozivam IzracunavanjeSumN za $m = 3$ gde se vrši izracunavanje $SumM = 1+2+3$	$SumM = 6$	Dobija se $SumM = 6$	Pass
6	Calculate k	Izracunavam $k$ , pomocu $k = m * n = 2 * 3$	Dobija se $k = 6$	Dobija se $k = 6$	Pass
7	Check Condition	Provera da li je $k$ razlicito od -1?	TRUE	$K = 6$ sledi TRUE	Pass
8	Call Method	Pozivam IzracunavanjeSumN za $k = 6$ gde se vrši izracunavanje $SumK = 1+2+3+4+5+6$	$SumK = 21$	$SumK = 21$	Pass
9	Calculate F	Izracunava se $F$ pomocu: $F = SumN * SumM / SumK$	Rezultat realan broj	$F = 0.85714$ sledi rezultat je realan broj	Pass
10	Output F	Ispis rezultata $F$	Rezultat ispisan na konzoli	Rezultat ispisan na konzoli	Pass
11	END	Kraj programa	Program završen	Program završen	Pass
12	Ukupno test:				Pass

Kao što se vidi iz prethodne tabele, test algoritma je završen uspešno. Za predefinisane vrednosti promenljivih  $n=2$  i  $m=3$ , svaki test korak (Test Step) je imao svoj očekivani rezultat koji se u potpunosti poklopio sa stvarnim rezultatom dobijenim izvršavanjem programa. Očekivani rezultat (Expected result) dobijen je prostim izračunavanjem za zadate promenljive  $n=2$  i  $m=3$ , pomoću formula:

$$SumN = 1 + 2 = 3,$$

$$SumM = 1 + 2 + 3 = 6,$$

$$k = 2 * 3 = 6,$$

$$SumK = 1 + 2 + 3 + 4 + 5 + 6 = 21,$$

$$F = SumN * SumM / SumK = 0.85714$$

### 5) Evaluacija rešenja

Prilikom izrade algoritma dobio sam sve potrebne vrednosti koje su mi neophodne da bih izračunao traženo rešenje formule F.

Izvršio sam zaštitu algoritma i celog programa od dobijanja iracionalnog rezultata tako što sam zadao proveru za vrednost promenljive k, odnosno sume proizvoda promenljivih n i m (kao:  $k < -1$ ?). Ova suma (SumK) figurira u imeniocu formule F, tako da ne sme imati vrednost 0.

Kao potprogram glavnog programa imam metodu, odnosno funkciju koja izračunava sumu za svaku zadatu promenljivu (n, m ili k) za koju je pozovem. To su pozivi funkcije:

- *IzracunavanjeSumN(n, SumN),*
- *IzracunavanjeSumN(m, SumM),*
- *IzracunavanjeSumN(k, SumK).*

Ovaj potprogram, tj. funkciju pozivam u glavnom programu (main) i vrednosti koje ta funkcija vraća upotrebljavam pri izračunavanju glavne formule zadatka, a to je formula za F, koja mi i daje rešenje problema koje ispisujem na kraju.

Formulu:

$$F = (1+2+ \dots + n)(1+2+ \dots + m) / (1+2+ \dots + n*m)$$

sam preformulisao u formulu:

$$F = \text{SumN} * \text{SumM} / \text{SumK}.$$

Prilikom praktične realizacije glavnog algoritma u programskom jeziku Java, primetio sam da je najpodesnije podprogram izući van glavne (main) metode programa i potom ga pozvati i vrednosti koje ova funkcija vraća smestiti u neke promenljive tipa double, koje sam dodatno uveo i na taj način osigurao da mi konačno rešenje bude realan a ne ceo broj. To je ujedno bila i najveća promena na celom programu, tj. algoritmu koja je učinila kod programa preglednijim i pojedinačne celine u programu lakše upotrebljivim i pogodnijim za korekciju i modifikaciju.

### 6) Dodatak

Java programski kod urađen na osnovu algoritma opisanog u projektu.

**Napomena:** Prema projektu nisam morao da ispišem ovaj kod, ali sam imao potrebe da pokažem da sam i praktično uspeo da implementiram algoritam iz zadatka.

```
package paket01;
```

```
public class projektni_zadatak02 {
```

```
    //Metoda za izracunavanje SumN
```

```
    static int IzracunavanjeSumN(int n, int SumN){
```

```
        for(int i=0; i<=n; i++){
```

```
            SumN += i;
```

```
        }
```

```
        return SumN;
```

```
    }
```

```
    //main metoda
```

```
    public static void main(String[] args) {
```

```
        int n=2, m=3, k, SumN=0, SumM=0, SumK=0;
```

```
        double x, y, z, F=0;
```

```
        x = IzracunavanjeSumN(n, SumN);
```

```
        y = IzracunavanjeSumN(m, SumM);
```

```
        System.out.println("SumN = " + x);
```

```
        System.out.println("SumM = " + y);
```

```
        k=n*m;
```

```
        System.out.println("k = " + k);
```

```
        if(k == -1){
```

```
            System.out.println("Broj k je -1, imenilac je 0 pa resenje nije  
racionalan broj. Unesite ponovo n ili m.");
```

```
        }
```

```
        z = IzracunavanjeSumN(k, SumK);
```

```
        System.out.println("SumK = " + z);
```

```
        F=x*y/z;
```

```
        System.out.println("F = " + F);
```

```
    }
```

```
}
```

MSc Branislav Manojlović