

[Programiranje]



Nastava: prof.dr.sc. Dražena Gašpar

Datum: 21.03.2017.

[Pripremiti za sljedeće predavanje]

- Sljedeće predavanje: 21.03.2017.
- Napraviti program koji koristi sve tipove podataka, osnovne operatore (+, -, /, * i modulo) i prikazuje rezultate izraza.
- Pripremiti:
 - Pripremiti se za test iz teorije:
 - Poglavlje 2 – Povijesni razvoj programskih jezika
 - Poglavlje 3 – Programske paradigme
 - Poglavlje 5 – Tipovi podataka

PONIJETI SVOJA RAČUNALA NA NASTAVU !!!

Upravljanje programskim tokom

***Svaka logička programska struktura
može se proizvesti od 3 osnovne:***

- sekvencija**
- selekcija**
- iteracija**



Simboli dijagrama toka

Početak i Kraj simboli koriste se za označavanje početka i završetka algoritma.

Početak

Kraj

Ulaz i Izlaz simbol koristi za učitavanje i/ili prikaz podataka.

Ulaz / Izlaz

Simbol točke odluke koristi se za prikaz grananja algoritma ovisno o rezultatu tj. ispunjenju ili neispunjenju zadanog uvjeta

DA

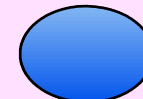
Uvjet
ispunjen?

NE

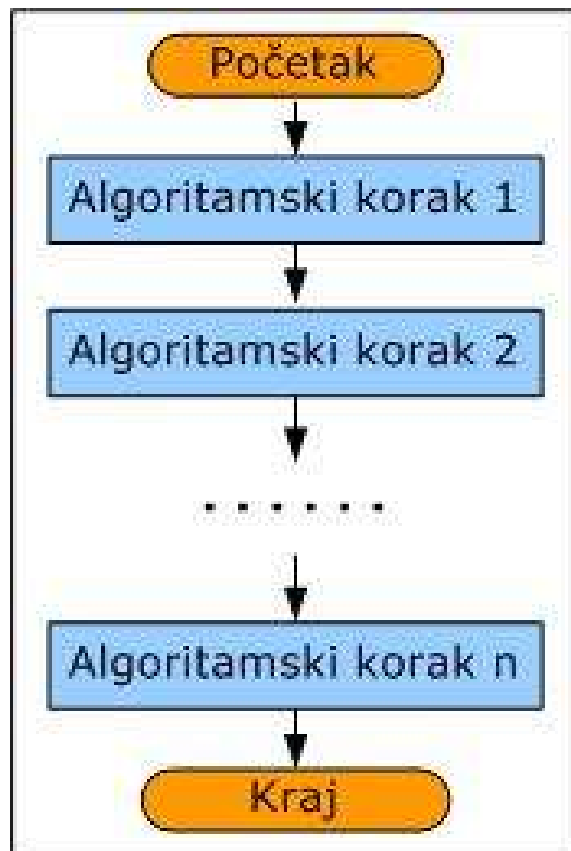
Simbol obrade koristi se za prikaz bilo kakvog izračuna ili obrade podataka.

Obrada

Simbol spajanja se koristi za povezivanje različitih dijelova blok dijagrama



[SEKVENCIJA]



The screenshot shows an IDE window with the title "ols Window Help". The "Source" tab is active, displaying the following Java code:

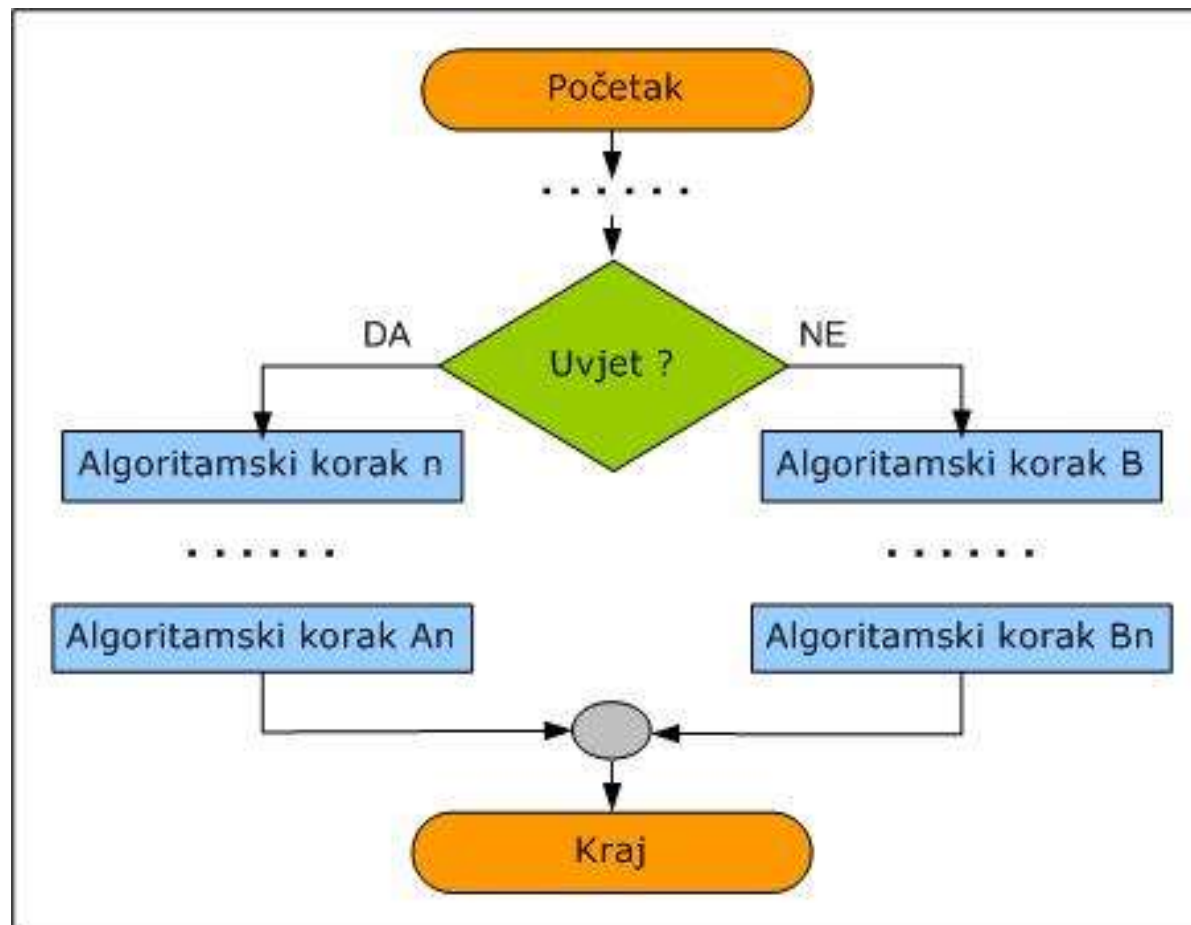
```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5   package javaapplication1;
6
7   /**
8    *
9    * @author user
10  */
11  public class PrimjerSekvencije {
12      public static void main(String[] args) {
13          int ocjena;
14          String student;
15          student = "Anica Aničić";
16          ocjena = 4;
17          System.out.println("Ime studenta je "+student);
18          System.out.println("Ocjena je "+ocjena);
19      }
20  }
21
```

The "Output - JavaApplication1 (run)" tab is also visible, showing the following output:

```
run:
Ime studenta je Anica Aničić
Ocjena je 4
BUILD SUCCESSFUL (total time: 0 seconds)
```

Upravljanje programskim tokom

Selekcija – uvjetno grananje (IF-THEN-ELSE)



Selekcija (uvjetno grananje)

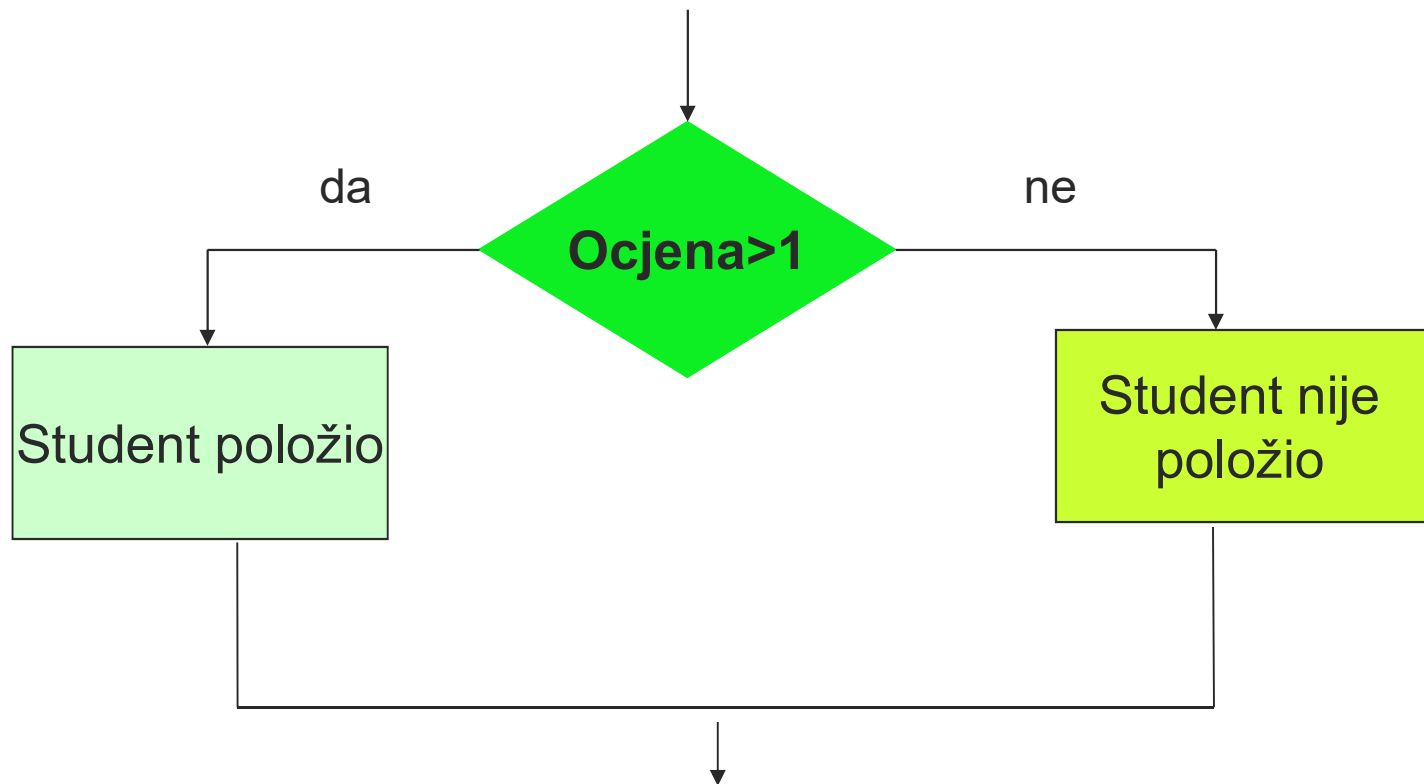
if ... else

```
if (uvjet tj.logički izraz) naredba1;  
else naredba2;
```

```
if(uvjet)  
    naredba;  
else if(uvjet)  
    naredba;  
else if(uvjet)  
    naredba  
...  
else  
    naredba;
```

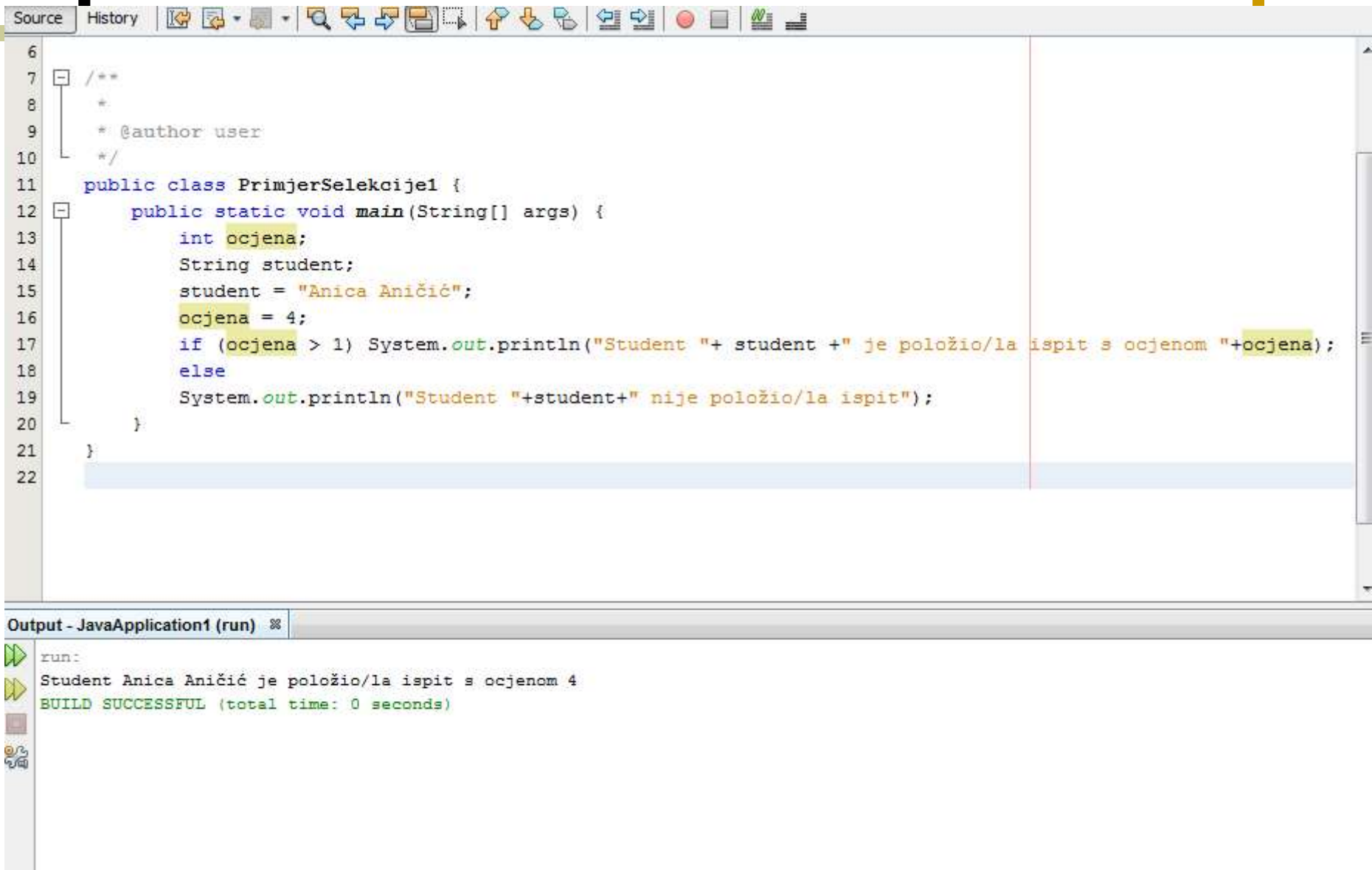
[Selekcija

if ... else]



Selekcija

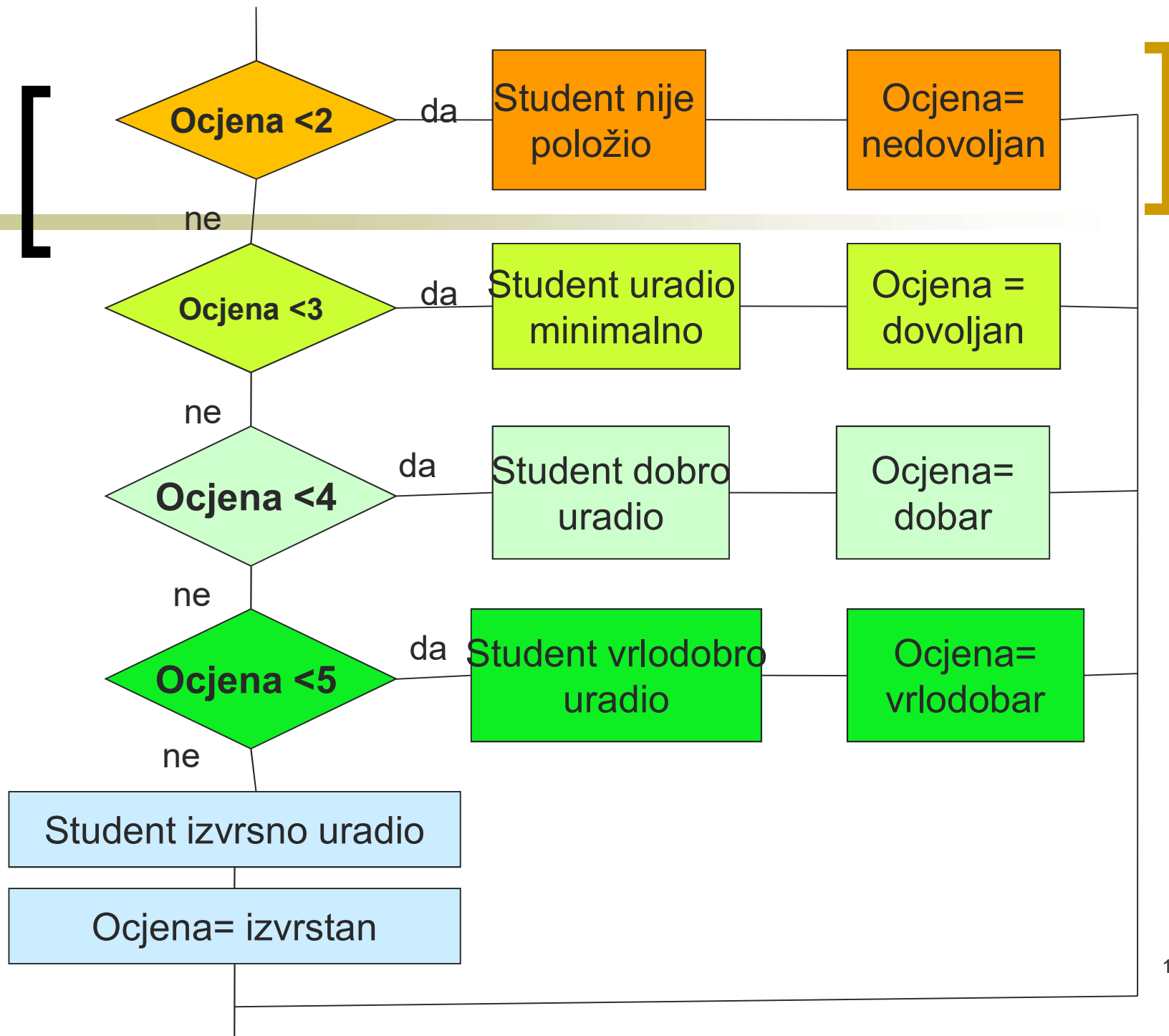
if ... else

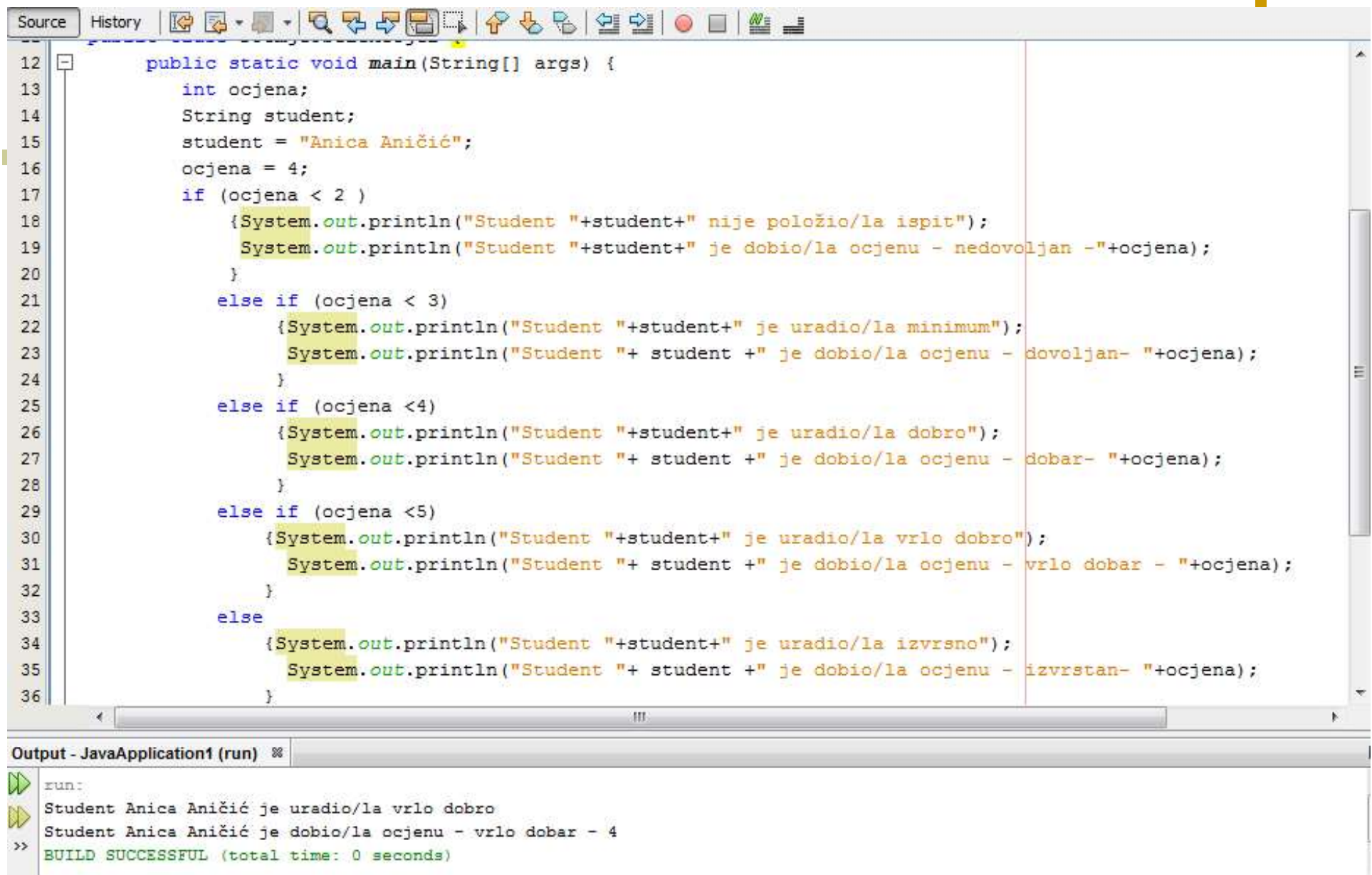


```
6
7 /**
8  *
9  * @author user
10 */
11 public class PrimjerSelekcije1 {
12     public static void main(String[] args) {
13         int ocjena;
14         String student;
15         student = "Anica Aničić";
16         ocjena = 4;
17         if (ocjena > 1) System.out.println("Student "+ student + " je položio/la ispit s ocjenom "+ocjena);
18         else
19             System.out.println("Student "+student+" nije položio/la ispit");
20     }
21 }
22
```

Output - JavaApplication1 (run) %

```
run:
Student Anica Aničić je položio/la ispit s ocjenom 4
BUILD SUCCESSFUL (total time: 0 seconds)
```





The screenshot shows an IDE window with a source code editor and an output console. The source code is a Java program that evaluates a student's grade based on a score. The score is 4, which falls into the 'vrlo dobar' (very good) category. The output console shows the program's execution, displaying the student's name and the corresponding grade description.

```
12 public static void main(String[] args) {
13     int ocjena;
14     String student;
15     student = "Anica Aničić";
16     ocjena = 4;
17     if (ocjena < 2 )
18     {System.out.println("Student "+student+" nije položio/la ispit");
19       System.out.println("Student "+student+" je dobio/la ocjenu - nedovoljan -"+ocjena);
20     }
21     else if (ocjena < 3)
22     {System.out.println("Student "+student+" je uradio/la minimum");
23       System.out.println("Student "+ student +" je dobio/la ocjenu - dovoljan- "+ocjena);
24     }
25     else if (ocjena <4)
26     {System.out.println("Student "+student+" je uradio/la dobro");
27       System.out.println("Student "+ student +" je dobio/la ocjenu - dobar- "+ocjena);
28     }
29     else if (ocjena <5)
30     {System.out.println("Student "+student+" je uradio/la vrlo dobro");
31       System.out.println("Student "+ student +" je dobio/la ocjenu - vrlo dobar - "+ocjena);
32     }
33     else
34     {System.out.println("Student "+student+" je uradio/la izvrsno");
35       System.out.println("Student "+ student +" je dobio/la ocjenu - izvrstan- "+ocjena);
36     }
```

Output - JavaApplication1 (run) %

```
run:
Student Anica Aničić je uradio/la vrlo dobro
Student Anica Aničić je dobio/la ocjenu - vrlo dobar - 4
>> BUILD SUCCESSFUL (total time: 0 seconds)
```

[If ... else]

U čemu je razlika:

??

```
int brojač=1, n=5;
```

```
...
```

```
if (brojač>0) {  
    Obrada();  
    brojač = n - 1;}
```

```
else
```

```
    Unos();  
    brojač = n;
```

```
int brojač=1, n=5;
```

```
...
```

```
if (brojač>0) {  
    Obrada();  
    brojač = n - 1;}
```

```
else
```

```
    {Unos();  
    brojač = n;}
```

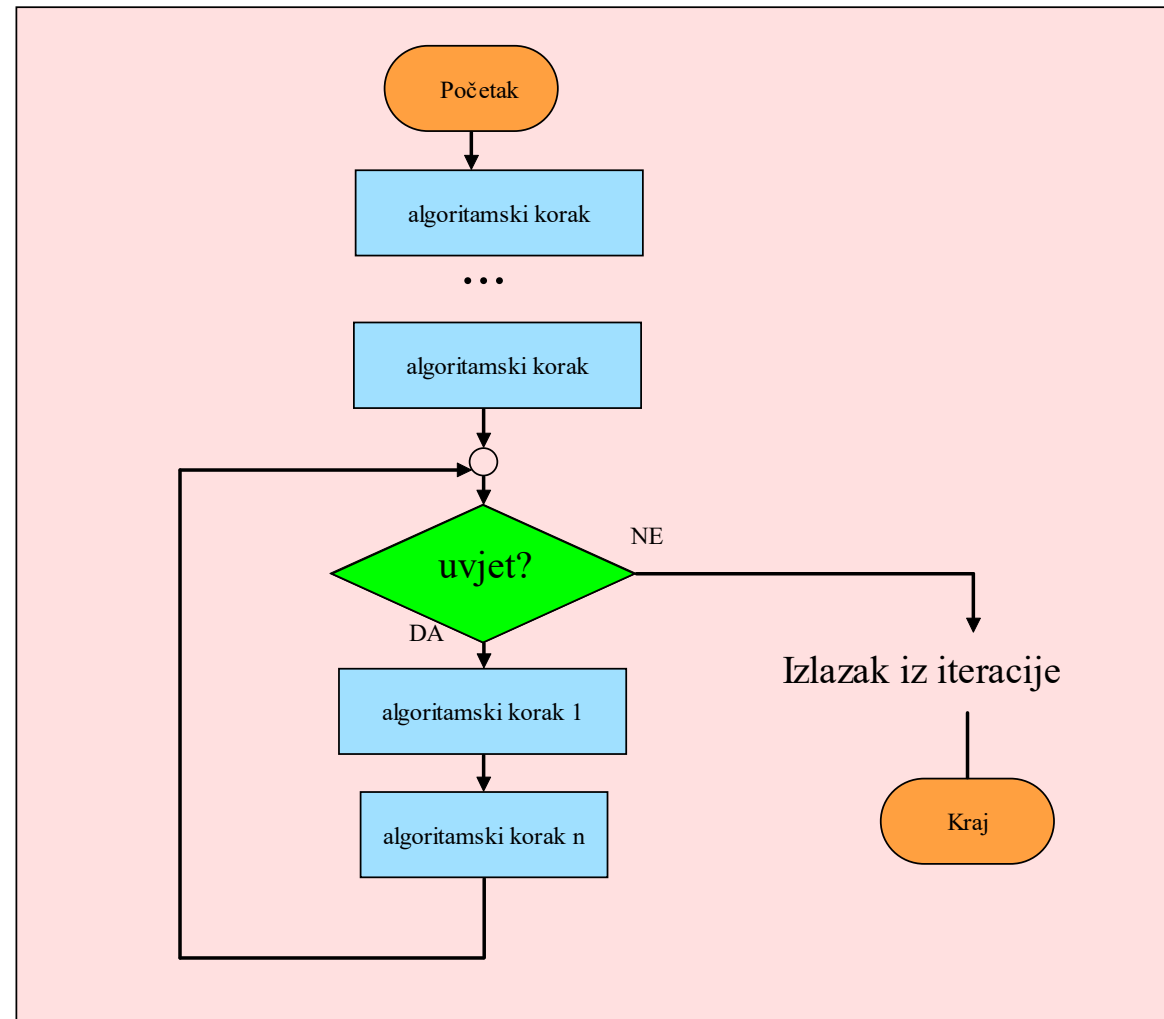
Upravljanje programskim tokom

**Iteracija – ciklična struktura – struktura
petlja - višestruko izvršavanje skupa
naredbi**

- 1. “DO WHILE” – pitaj pa radi**
- 2. “DO UNTIL” – radi pa pitaj**

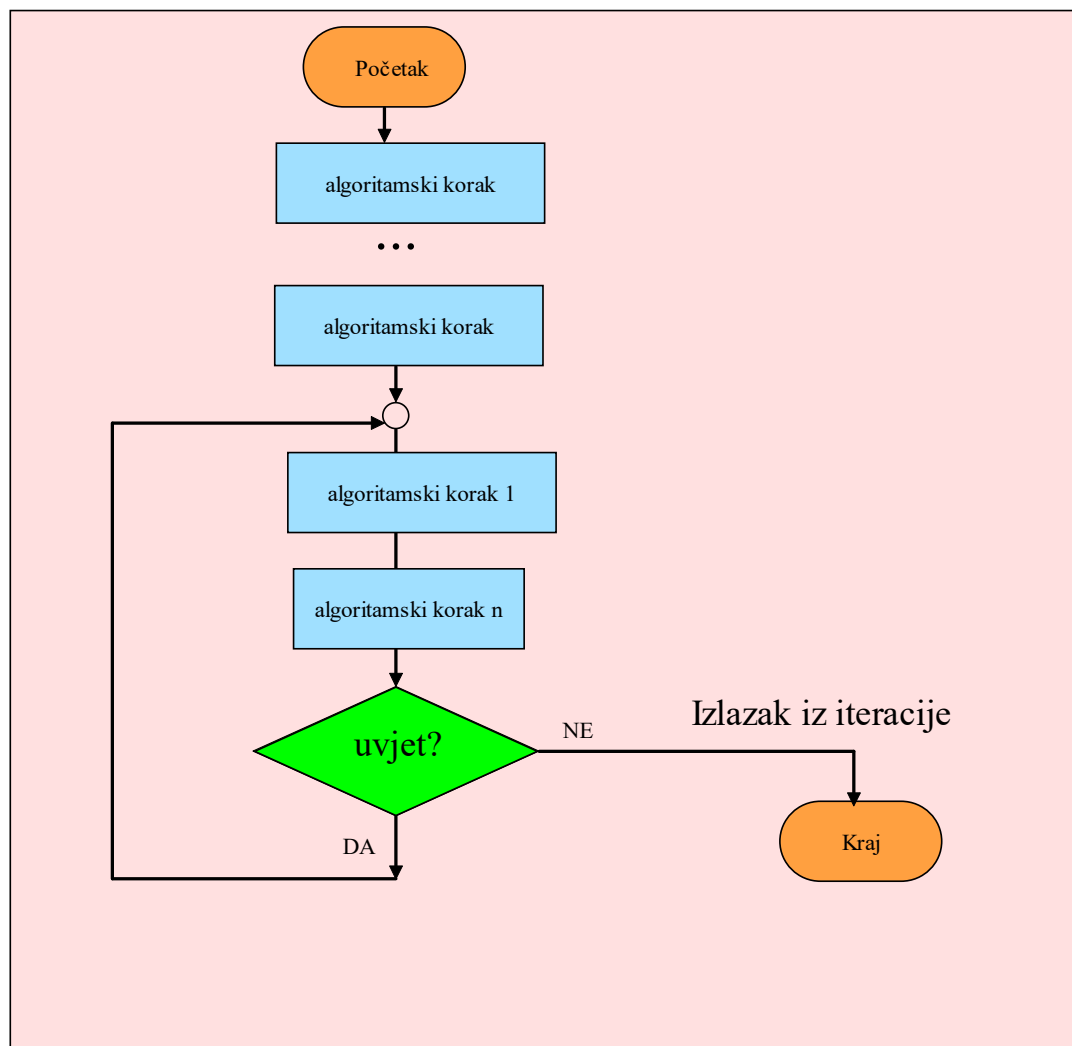
Upravljanje programskim tokom

Iteracija – “DO WHILE”



Upravljanje programskim tokom

Iteracija – “DO UNTIL”

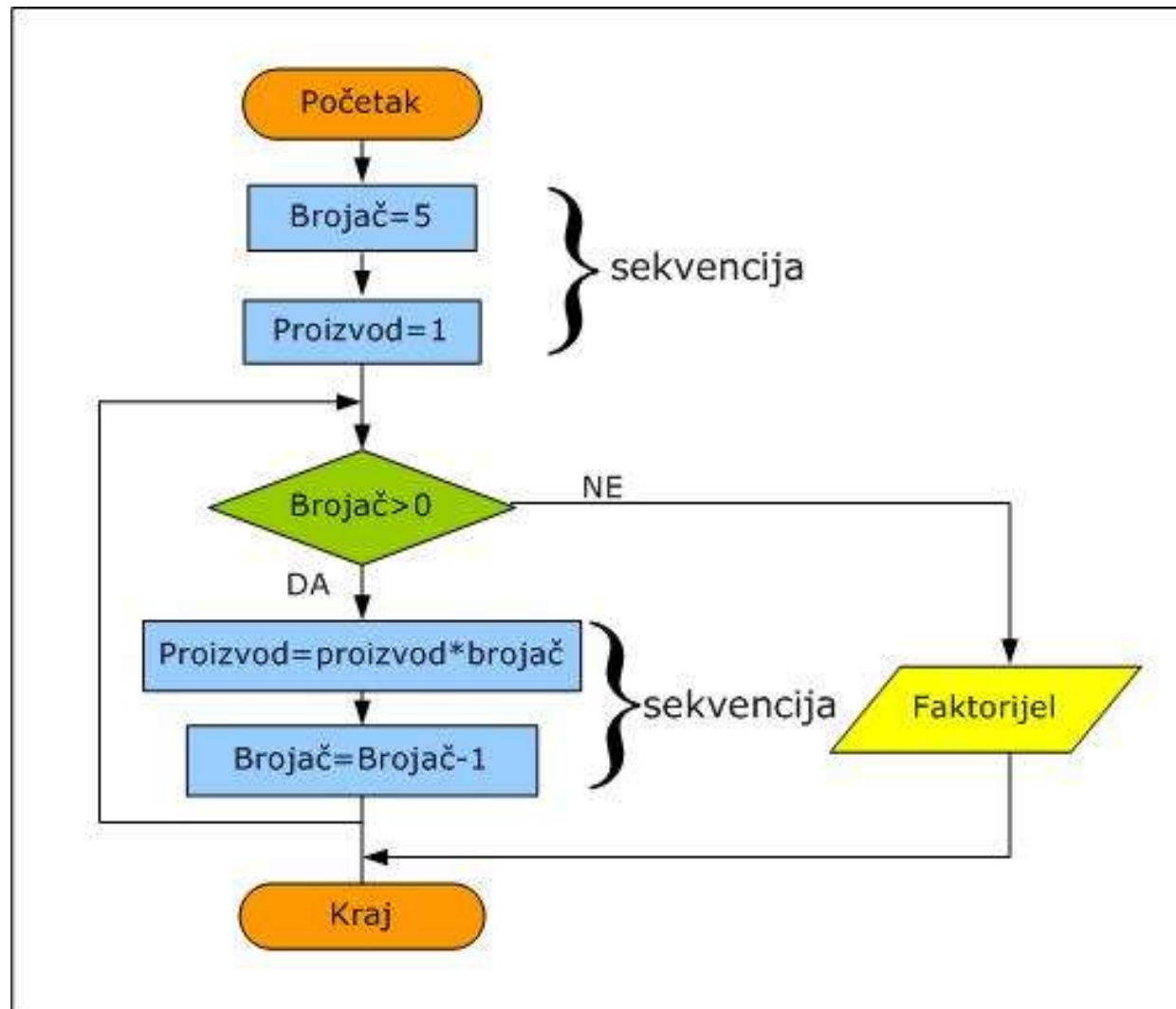


[Primjeri blok dijagrama]

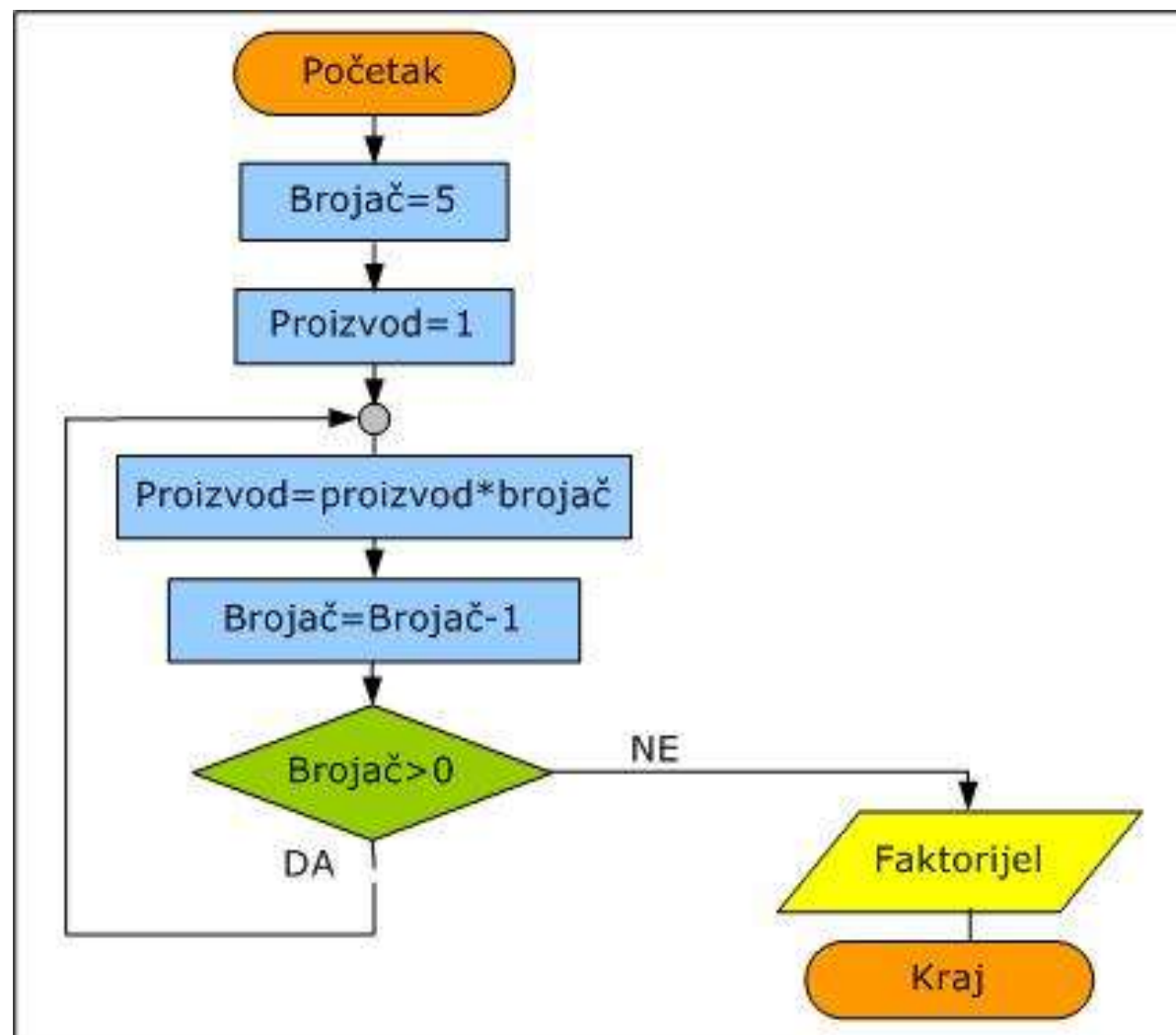
Primjer: Algoritam za izračunavanje faktoriijela broja 5 – $1 \times 2 \times 3 \times 4 \times 5$

1. Postaviti brojač na 5
2. Postaviti proizvod na 1
3. Sve dok je brojač veći od nule ponavljati slijedeće korake:
 - 3.1. pomnožiti proizvod i brojač
 - 3.2. umanjiti brojač za 1
4. Kada vrijednost brojača postane nula, zaustavlja se izvršavanje algoritamskih koraka.

Primjeri blok dijagrama



Primjeri blok dijagrama



[Switch]

- Alternativna sintaksa za if... else if ... else if
(ugniježdene if naredbe)

U situaciji kada svaki uvjet uspoređuje isti cjelobrojni izraz s različitom konstantom.

```
i = izraz
If i = 1 then naredba_A;
else if i = 2 then naredba_B;
else if i = 3 then naredba_C;
....
else naredba_N;
```

```
switch (izraz) {
    case 1:
        naredba_A
        break;
    case 2:
        naredba_B
        break;
    case 3:
        naredba_C
        break;
    default:
        naredba_N}
```

[Switch]

```
switch (izbor)
{
    case izbor1;
    ....
    break;
    case izbor2;
    break;
    case izbor3;
    break;
    case izbor4;
    break;
    ....
    default
    ....
    break;
}
```

Izbor tj.izraz koji se provjerava
MORA biti :

- byte
- char
- short ili
- int

(NE MOŽE biti boolean, long,
double, float, String)

Java 7 dozvoljava String i enumeracije

[Switch - enumeracije]

```
private static enum Smjerovi {SJEVER, JUG, ISTOK, ZAPAD};
```

```
Smjerovi smjer = Smjerovi.JUG;
```

```
Switch (smjer) {
```

```
    case SJEVER:
```

```
        System.out.println("Ići na sjever");
```

```
        break;
```

```
    case JUG:
```

```
        System.out.println("Ići na jug");
```

```
        break;
```

```
case ISTOK:
```

```
    System.out.println("Ići na istok");
```

```
    break;
```

```
case ZAPAD:
```

```
    System.out.println("Ići na zapad");
```

```
    break;
```

```
}
```

[Switch - String]

```
switch (grad) {  
    case "Mostar":  
        troškoviDostave = težina*0.20;  
        break;  
    case "Sarajevo":  
        troškoviDostave = težina*0.35;  
        break;  
    case "Banja Luka":  
        troškoviDostave = težina*0.60;  
        break;  
    default:  
        troškoviDostave = težina*0.40;  
}
```

Null vrijednost →
`java.lang.NullPointerException`

Osjetljivost na velika i
mala slova !!!



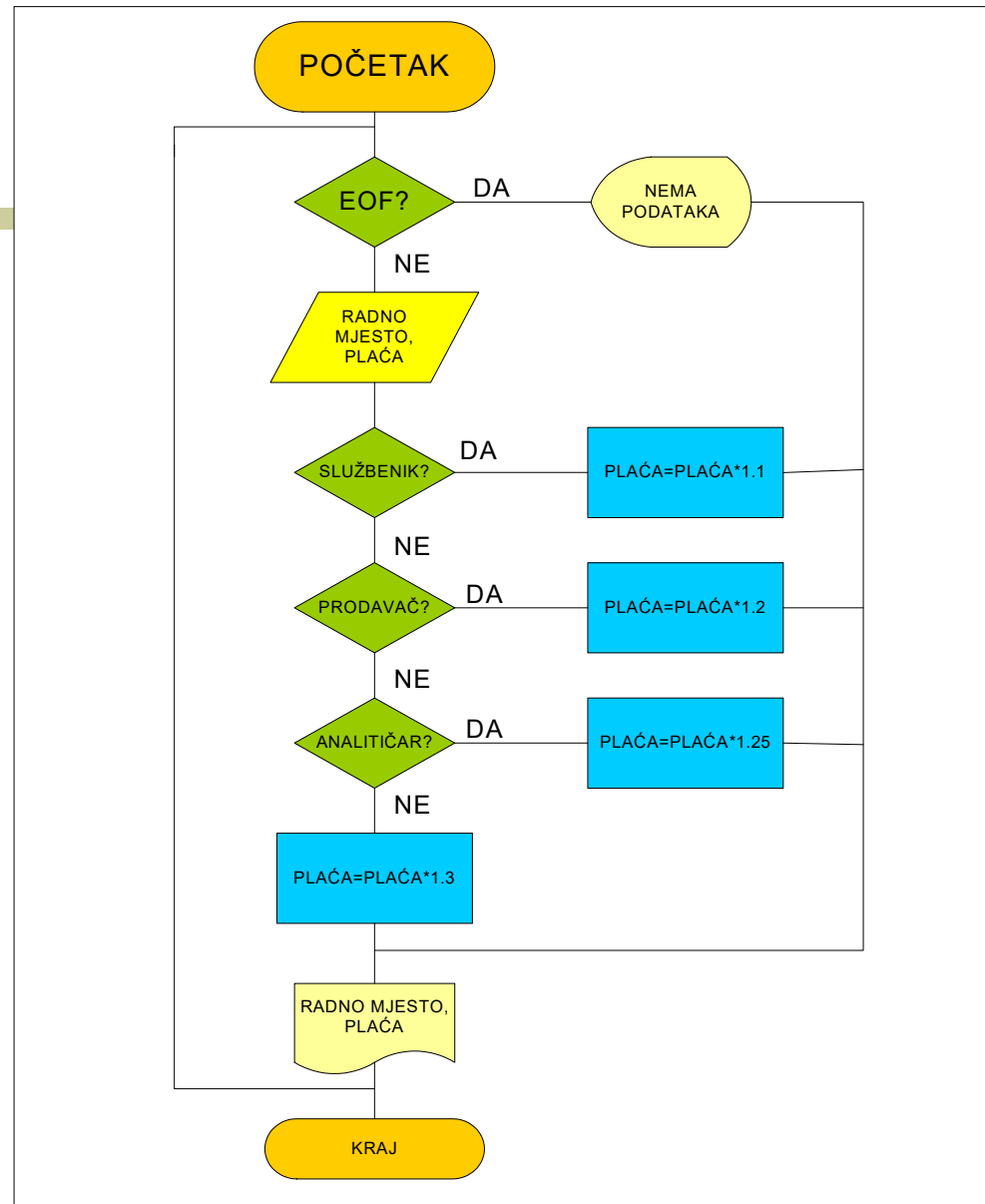
```
14     String student;
15     student = "Anica Aničić";
16     ocjena = 4;
17     switch (ocjena){
18         case 1:
19             {System.out.println("Student "+student+" nije položio/la ispit");
20              System.out.println("Student "+student+" je dobio/la ocjenu - nedovoljan -"+ocjena);
21              break; }
22         case 2:
23             {System.out.println("Student "+student+" je uradio/la minimum");
24              System.out.println("Student "+ student + " je dobio/la ocjenu - dovoljan- "+ocjena);
25              break; }
26         case 3:
27             {System.out.println("Student "+student+" je uradio/la dobro");
28              System.out.println("Student "+ student + " je dobio/la ocjenu - dobar- "+ocjena);
29              break; }
30         case 4:
31             {System.out.println("Student "+student+" je uradio/la vrlo dobro");
32              System.out.println("Student "+ student + " je dobio/la ocjenu - vrlo dobar - "+ocjena);
33              break; }
34         case 5:
35             {System.out.println("Student "+student+" je uradio/la izvrsno");
36              System.out.println("Student "+ student + " je dobio/la ocjenu - izvrstan- "+ocjena);
37              break; }
38         default:
39             {System.out.println("NEPOSTOJEĆA OCJENA"); }
40     }
```

Output - JavaApplication1 (run) »

```
>>> Student Anica Aničić je uradio/la vrlo dobro
>> Student Anica Aničić je dobio/la ocjenu - vrlo dobar - 4
BUILD SUCCESSFUL (total time: 0 seconds)
```

Primjeri dijagrama toka

Za djelatnike koji rade na radnom mjestu
službenika plaću pomnožiti koeficijentom
1.1,
za one na radnom mjestu prodavača s 1.2,
za analitičare s 1.25,
a za menadžere s 1.3.



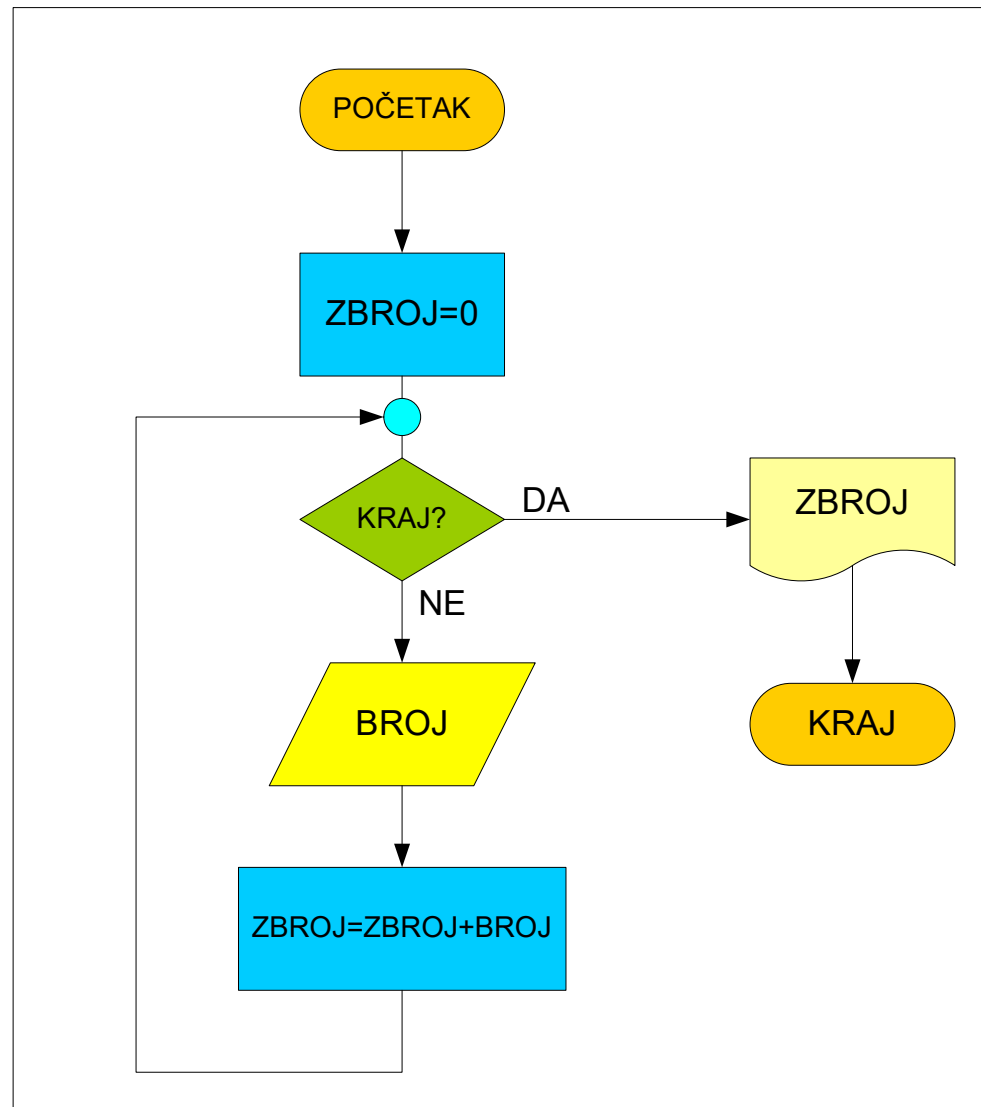
Primjeri dijagrama toka

Formirati algoritam za izračunavanje zbroja brojeva:

- a) Nepoznat broj članova(elementata) zbroja
- b) Za prvih 20 ulaznih brojeva

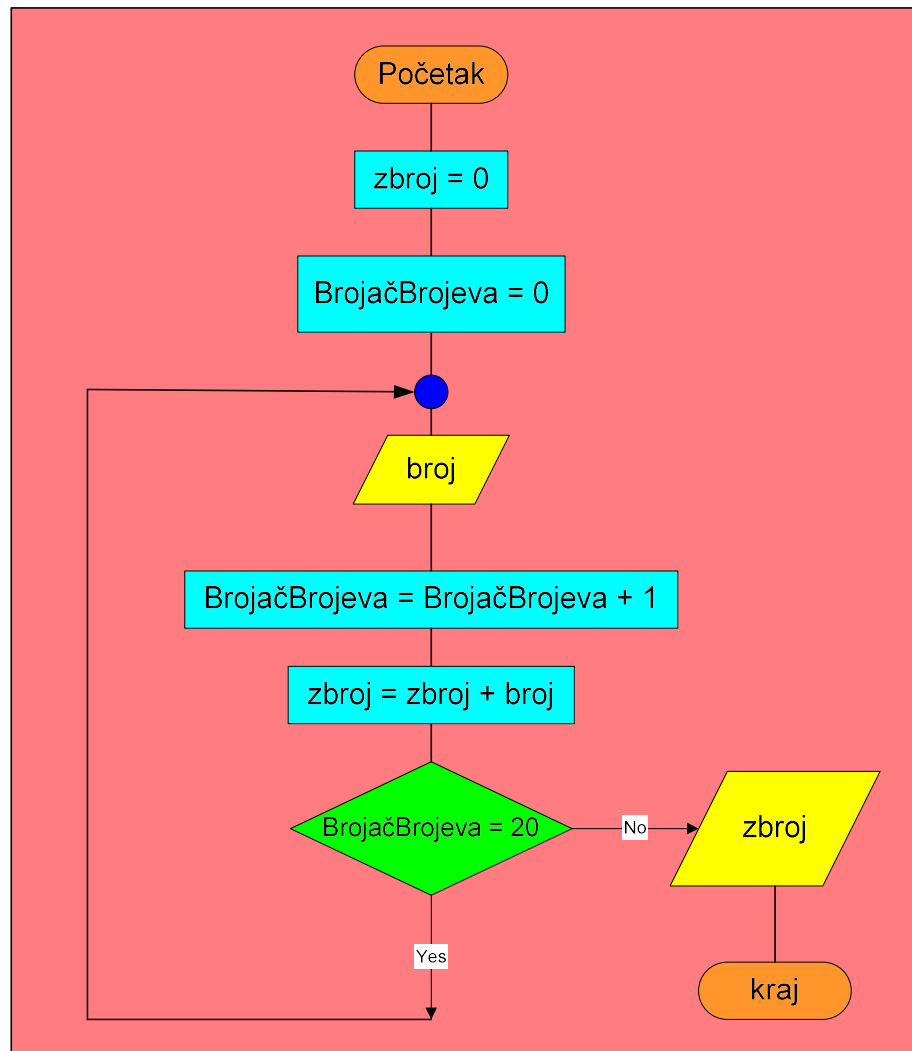
Primjeri dijagrama toka

a) Nepoznat broj članova(elementa) zbroja



Primjeri dijagrama toka

b) Za prvih 20 ulaznih brojeva

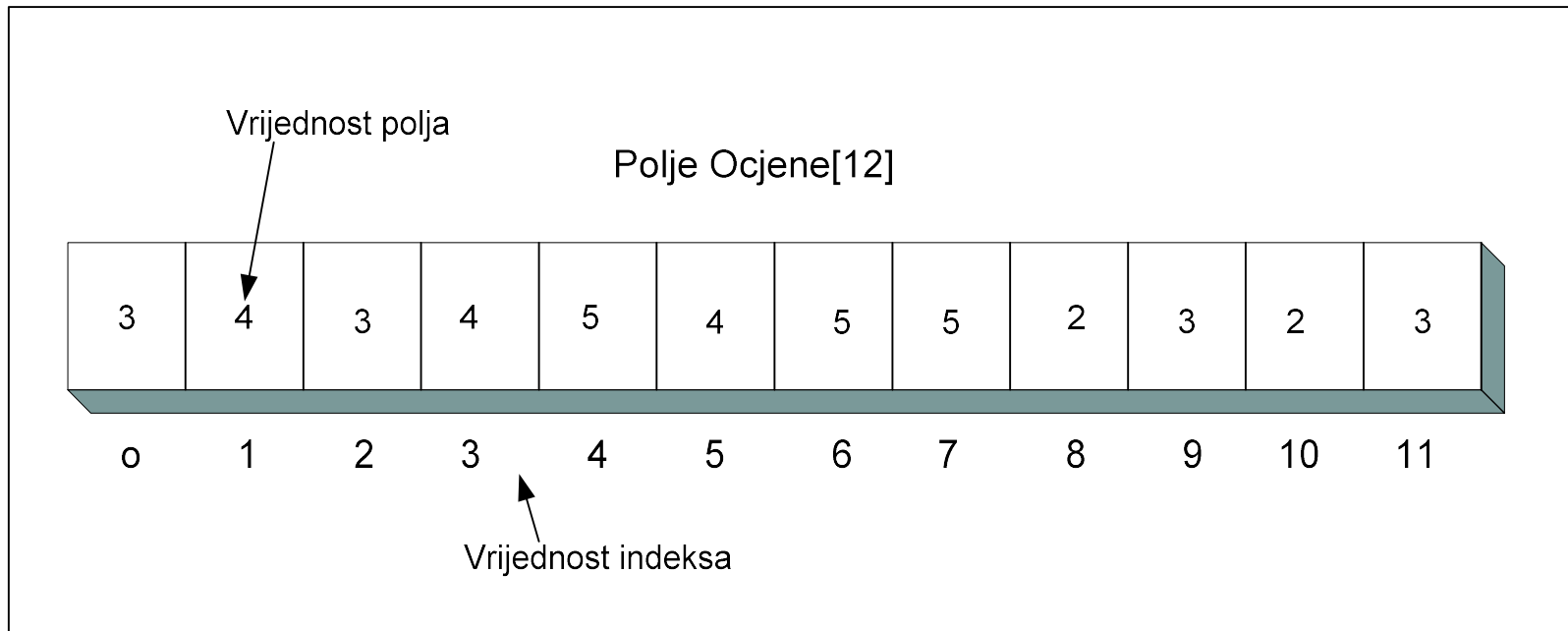


[Polje (array)]

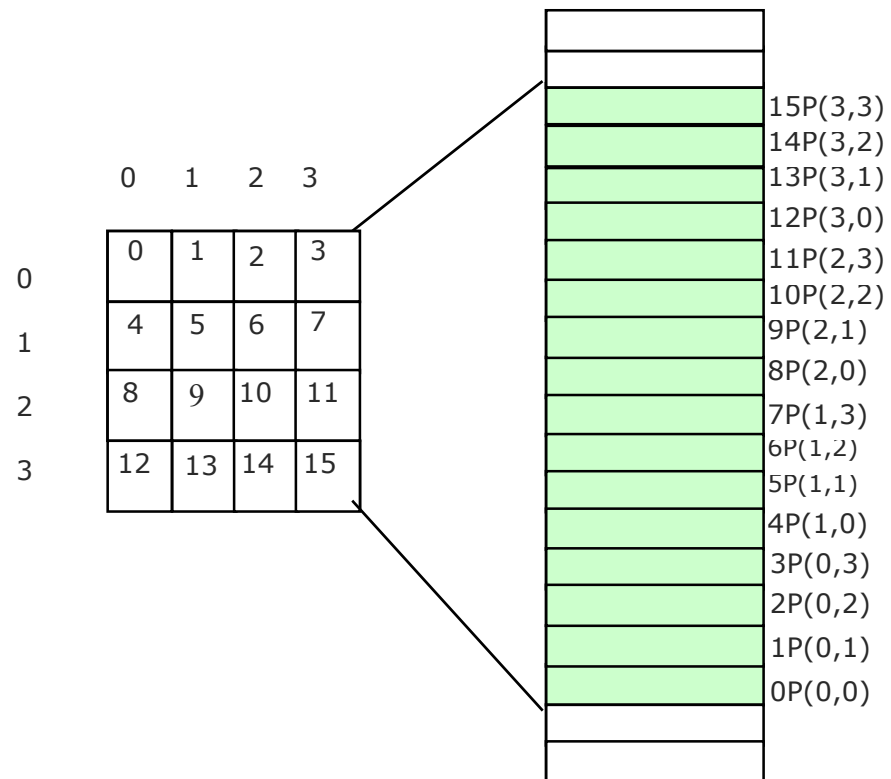
Podatkovna struktura gdje isto ime dijeli više podataka.

- uređeni skup podataka čiji su elementi istog tipa.
- najčešća struktura podataka
- indeks je osnovni elemenat pristupa polju koji definira uređenost polja.

[Polje (array)]



[Polje (array)]



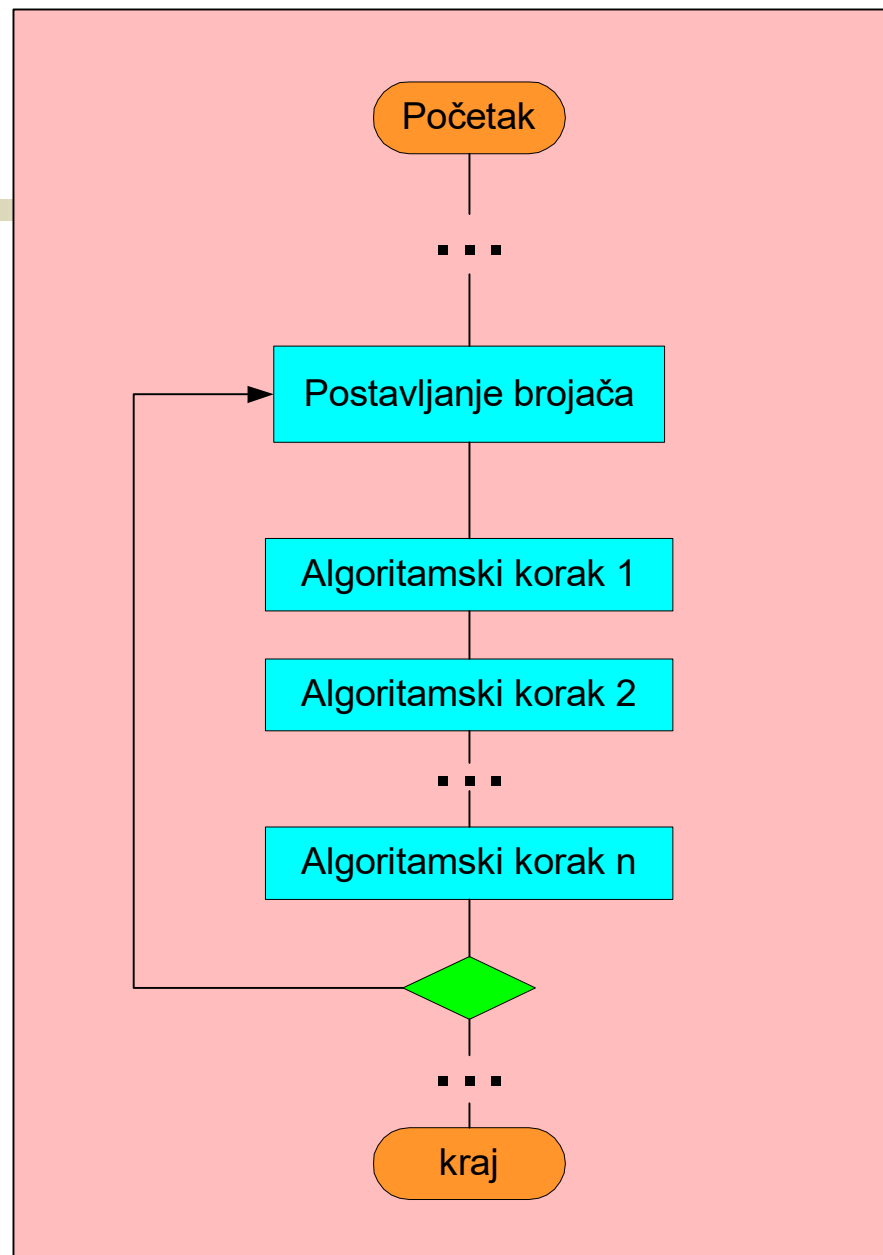
Preslikavanje elemenata po redovima

[FOR petlja]

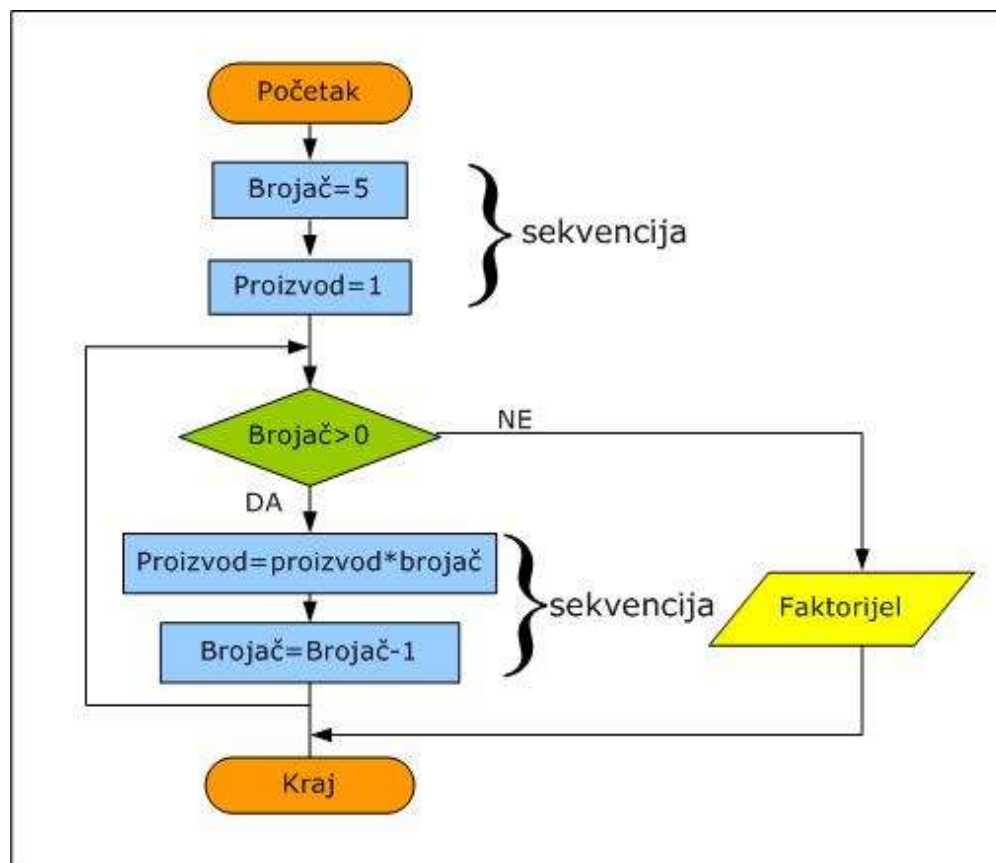
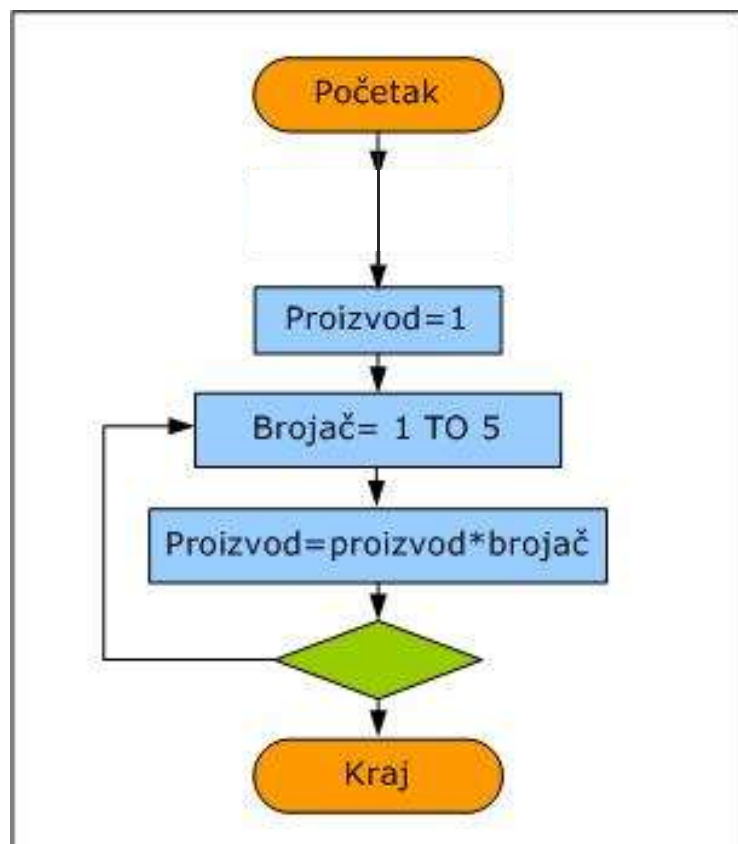
Ciklična struktura koja podrazumijeva da je broj iteracija unaprijed poznat:

- određena se vrijednost broji od neke početne vrijednosti pa sve do neke krajnje vrijednosti,
- rad s područjima (jednodimenzijska - liste, dvo i višedimenzijska područja).

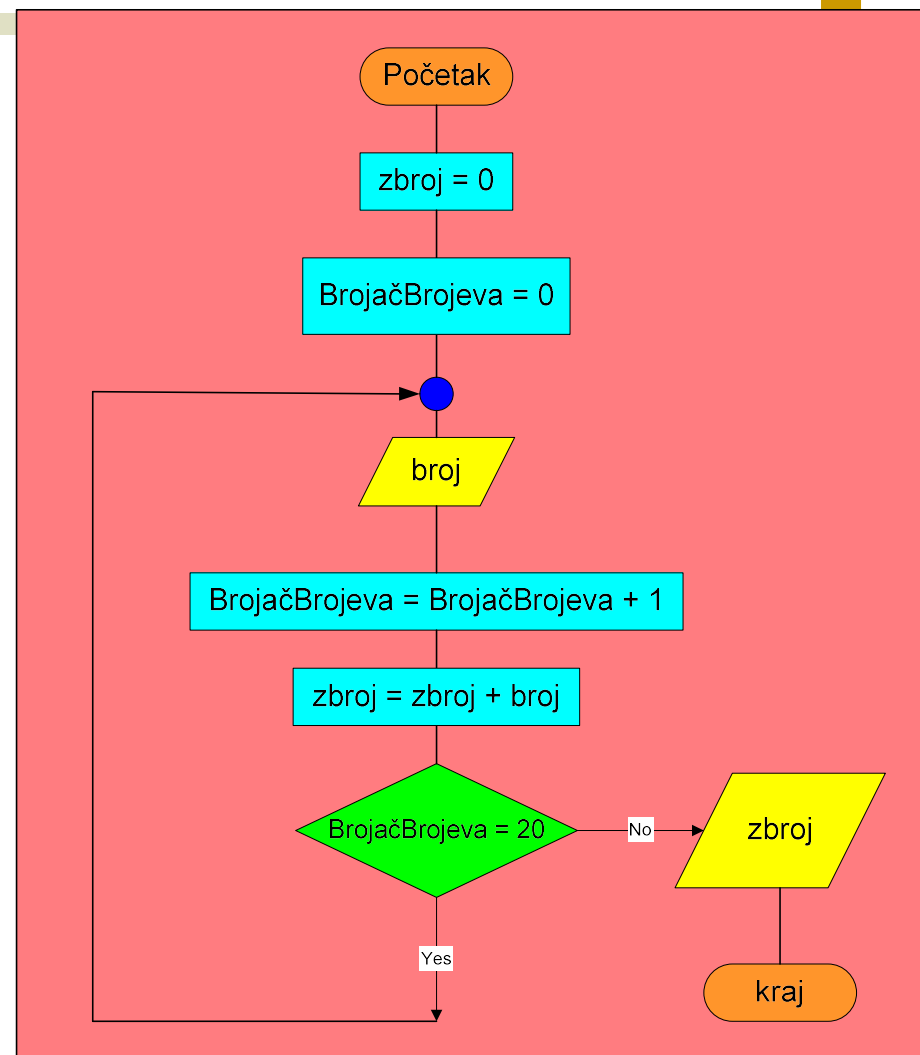
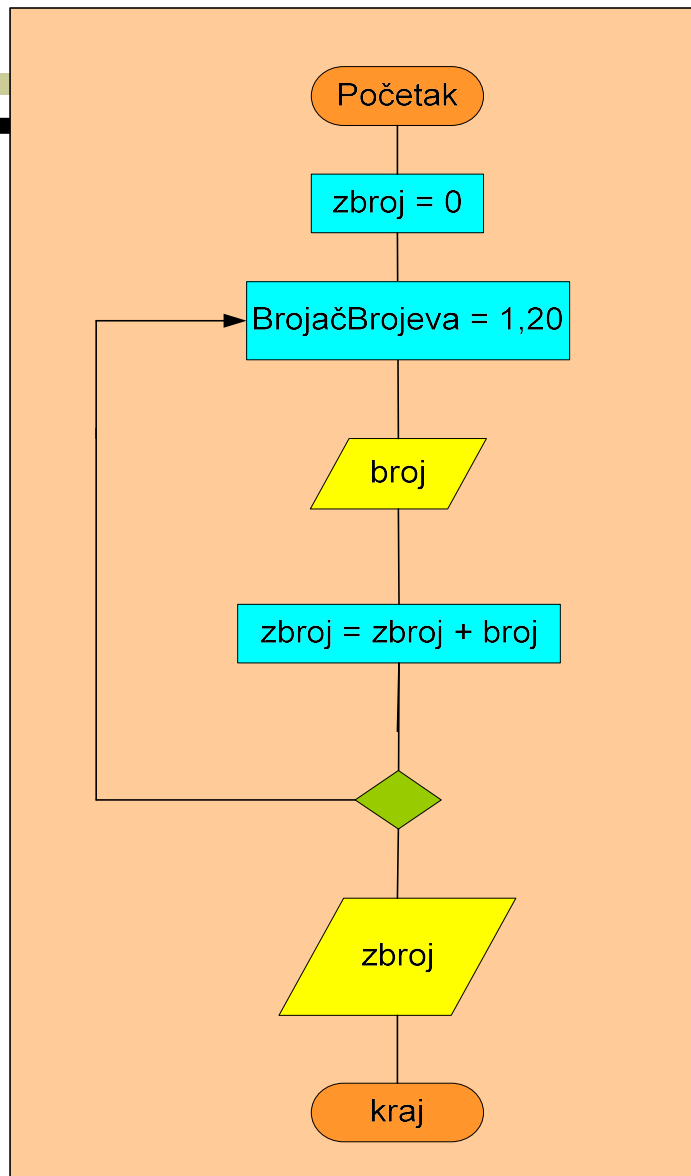
[FOR petlja



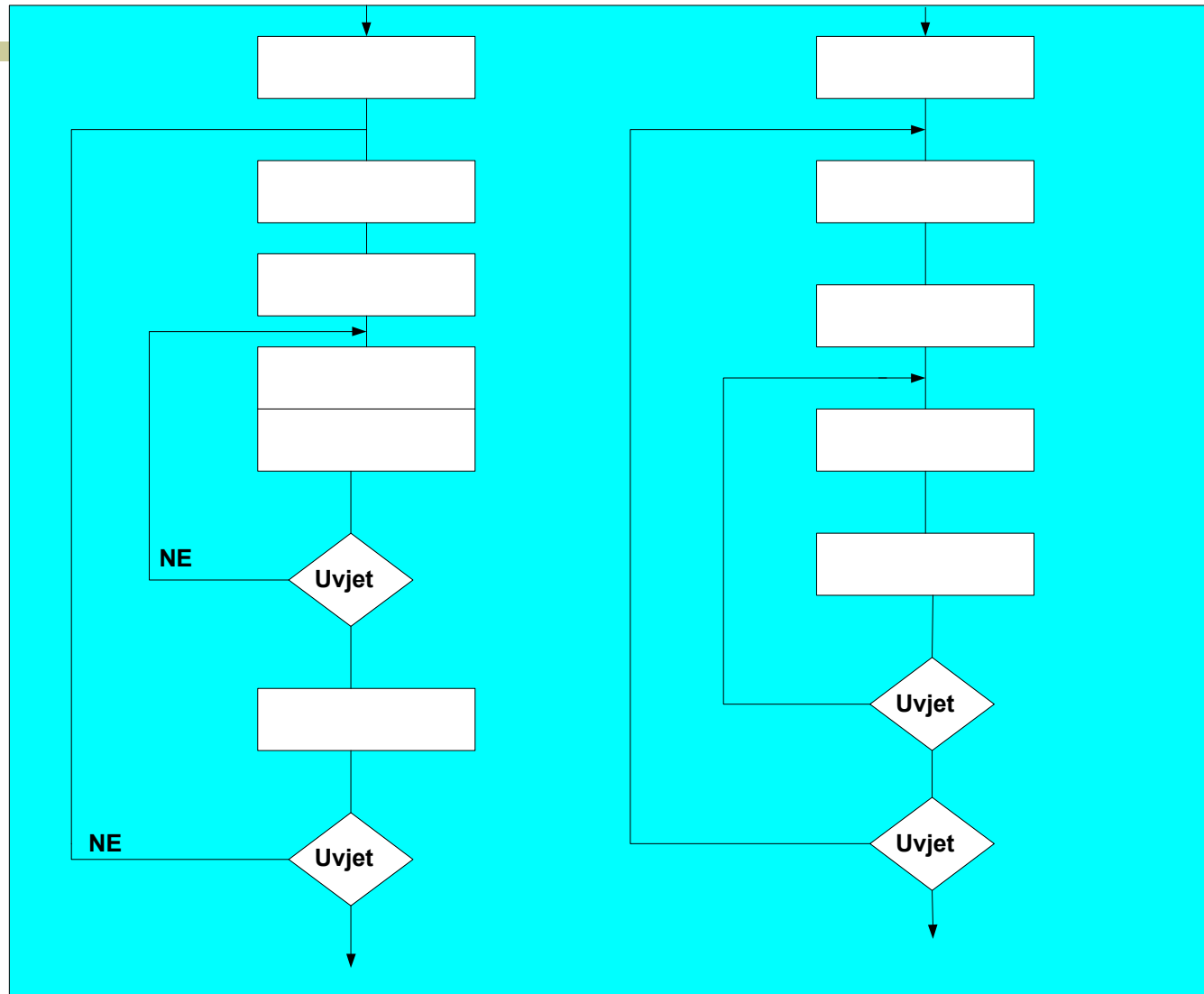
FOR petlja – primjer faktorzela



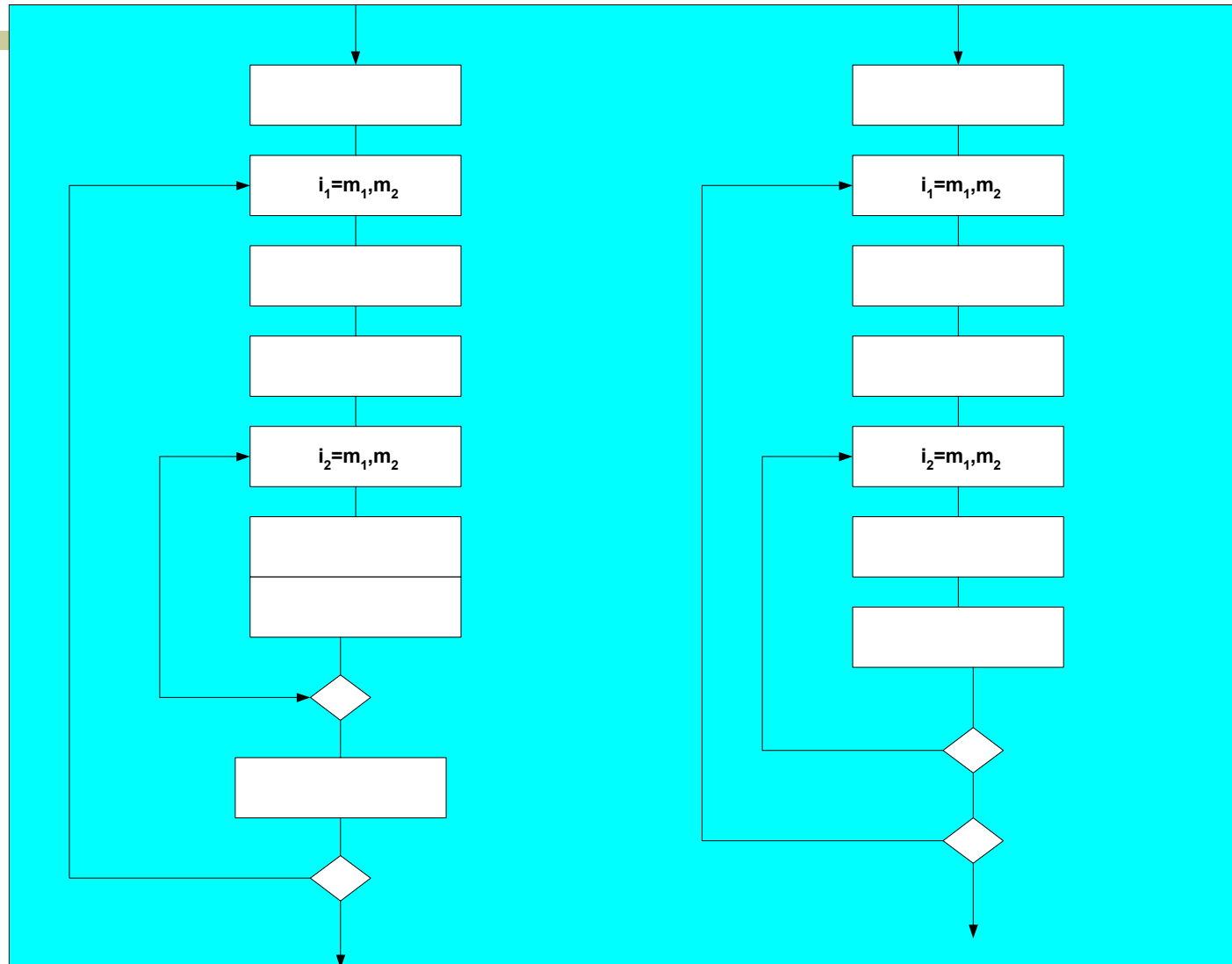
Formirati algoritam za izračunavanje zbroja brojeva za prvih 20 brojeva



Dvostruko ugnježđene ciklične strukture (1)



Dvostruko ugnježdene ciklične strukture (2)



[Sortiranje – bubble sort]

Algoritam “bubble” sort je jednostavan za početnik, ali i veoma spor.

Logika: elementi niza se uspoređuju i zamjenjuju po 2 istovremeno.

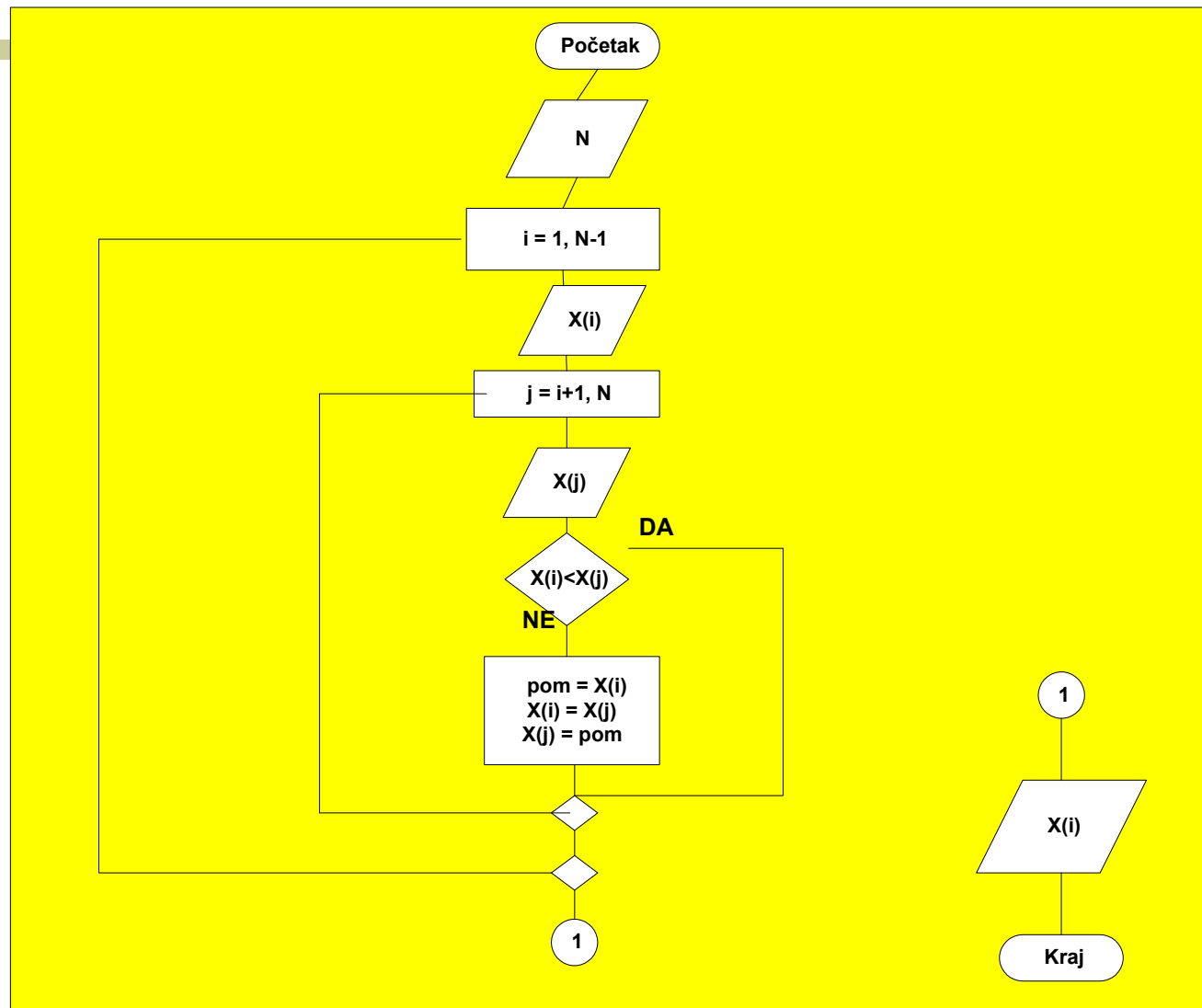
Za vrijeme svakog prolaza, “bubble” sortiranje postavlja najmanju (najveću) vrijednost u prvi element niza.

[Primjer]

Dani niz $X(i)$ sortirati:

- a) u rastućem redoslijedu
- b) u opadajućem redoslijedu

[Primjer]



Za Predavanje 04.04.2017.

Za Test 3 :

- Poglavlje 4
- Poglavlje 6

Kreirati dijagram toka i napraviti program koji izračunava najmanju prodaju za svaku prodavaonu (smješta ih u polje M) te prosječnu prodaju po prodavaonicama (smješta ih u polje P)

PRODAJA	Artikl 1	Artikl 2	Artikl 3	Artikl 4
Prodavnica 1	14	12	14	15
Prodavnica 2	5	4	4	4
Prodavnica 3	23	33	31	24

[PITANJA]

