



CODE@FEIT

ЧАС 3: ФУНКЦИИ КОИ ВРАЌААТ ВРЕДНОСТ

WHILE, DO-WHILE И FOR СТРУКТУРИ ЗА ПОВТОРУВАЊЕ

ПРИМЕРИ НА ПОСЛОЖЕНИ ФУНКЦИИ



ПОВТОРУВАЊЕ₁: ДА СЕ НАПИШЕ ФУНКЦИЈА КОЈА ВО ЗАВИСНОСТ ОД ПОЕНИТЕ ЌЕ ЈА ИСПЕЧАТИ ОЦЕНКАТА НА СТУДЕНТОТ

```
public class TestVoidMethod {
    public static void main(String[] args) {
        printGrade(78.5); // повик на функцијата
    }
    public static void printGrade(double score) {
        if (score >= 90.0) {
            System.out.println('10');
        }
        else if (score >= 80.0) {
            System.out.println('9');
        }
        else if (score >= 70.0) {
            System.out.println('8');
        }
        else if (score >= 60.0) {
            System.out.println('7');
        }
        else if (score >= 50.0) {
            System.out.println('6');
        }
        else {
            System.out.println('5');
        }
    }
}
```

Која ќе биде оценката?

8



ФУНКЦИИ КОИ ВРАЌААТ ВРЕДНОСТ

- Дефинирање:

```
public static tip_funkcija ime_funkcija(tip_param ime_param)
{
    //наредби
    return vrednost;
}
```

- Повикување:

```
tip_promenлива ime_promenлива =ime_funkcija(ime_param)
```



ПРОГРАМА I: ПРЕСМЕТУВАЊЕ НА АРИТМЕТИЧКА СРЕДИНА

```
public class Test{  
public static int presmetajAS(int x, int y, int z) //дефинирање на функција со три аргументи  
{  
    return (x+y+z)/3;  
}  
public static void main(String[] args) {  
    int v1=3, v2=5, v3=10, aSredina;  
    aSredina= presmetajAS( v1,v2,v3 ); //повик на функција со три аргументи  
    System.out.println( aSredina);  
    System.out.println( presmetajAS(6,7,9) );  
}  
}
```



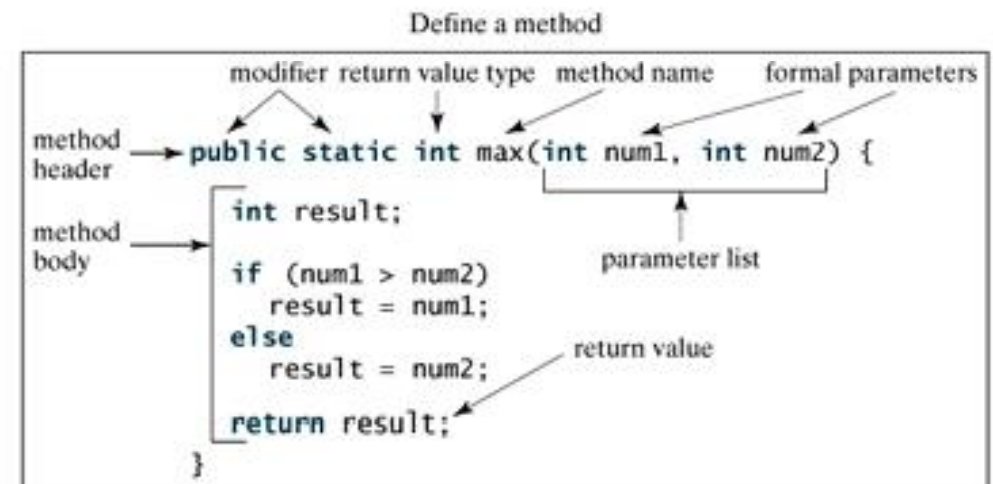
ПРОГРАМА₂: ПРЕСМЕТУВАЊЕ НА ПЛОШТИНА НА ПРАВОАГОЛНИК

```
Public class Test{
public static int presmetajPlostina(int a, int b) //дефинирање на функција со три аргументи
{
    return a*b;
}
public static void main(String[] args) {
    int v1=3, v2=5, Plostina;
    Plostina= presmetajPlostina( v1,v2); //повик на функција со два аргументи
    System.out.println(Plostina);
}
}
```



ПРОГРАМА 3: НАОЃАЊЕ НА МАКСИМУМ ПОМЕЃУ ДВА БРОЈА

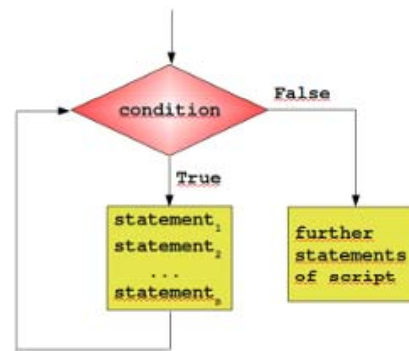
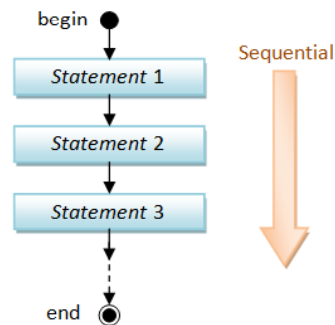
```
public class TestMax {  
    /** Main method */  
    public static void main(String[] args) {  
        int i = 5;  
        int j = 2;  
        int k = max(i, j);  
        System.out.println("The maximum between " + i +  
            " and " + j + " is " + k);  
    }  
  
    /** Return the max between two numbers */  
    public static int max(int num1, int num2) {  
        int result;  
        if (num1 > num2)  
            result = num1;  
        else  
            result = num2;  
  
        return result;  
    }  
}
```





ВОВЕДУВАЊЕ НА ЈАМКИ

- Секвенцијално програмирање
- Гранење со помош на `if` наредба
- Извршување на група инструкции во јамка се додека определен услов е исполнет
 - While
 - For





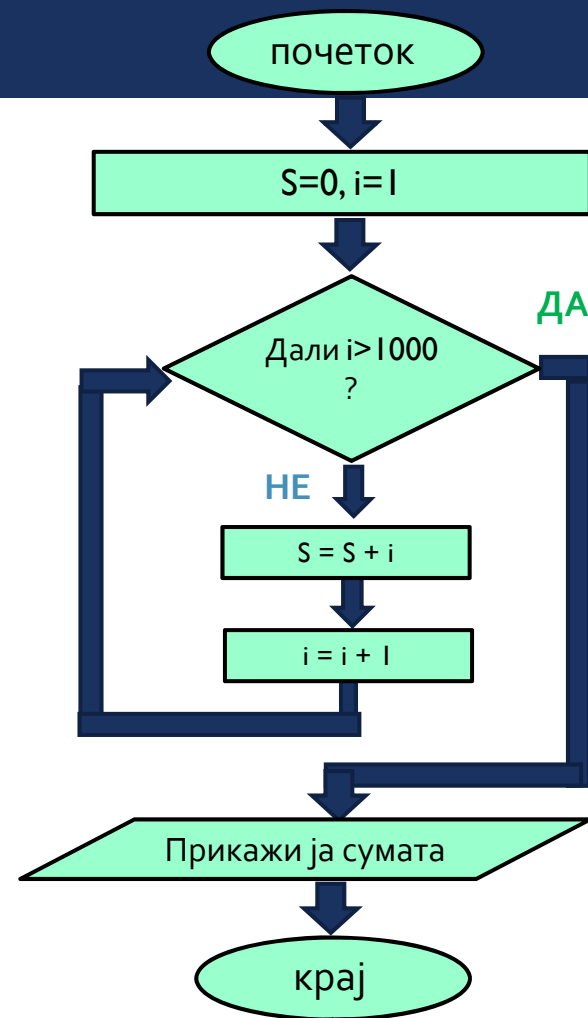
ЕДНОСТАВЕН ПРИМЕР₁

- Собирање на првите 10 цели броеви (од 1 до 10):
 - `int addition = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;`
- Проблем₁:
 - **Собирање на првите 1000 цели броеви (од 1 до 1000)**
- Решение:
 - Употреба на јамка за циклично собирање на елементите



АЛГОРИТАМ

1. Постави сума на 0
2. Постави бројач i на 0
3. **Ако $i > 1000$** оди на чекор 7 (крај на повторување), инаку оди на чекор 4.
4. На сумата од претходниот чекор додај го i .
5. Зголеми го бројачот i за 1.
6. Оди на чекор 3.
7. Прикажи ја сумата
8. Крај





WHILE ЈАМКА

иницијализација

while (услов) {

Блок на наредби

}

- На почеток секогаш се проверува дали условот е исполнет
 - **True:** изврши го блокот од наредби дадени во **while** циклусот
 - **False:** излези од **while** циклусот и изврши ги останатите наредби, дадени после јамката
- **While** јамката **се повторува** се додека условот е исполнет. Во моментот кога условот нема да се исполни, јамката прекинува.



РЕШЕНИЕ НА ПРИМЕР₁ СО WHILE ЈАМКА

ПРИМЕР₁: Собирање на првите 1000 цели броеви (од 1 до 1000)

```
public class Sumanabroeви{
    public static void main(String[] args) {
        int i = 1, suma = 0;
        while(i<=1000){
            suma=suma+i;
            i++;
        }
        System.out.println("Sumata iznesuva " + suma);
    }
}
```

ПРИМЕР₂: Да се прошири претходната задача така што ќе се пресмета и испечати аритметичка средина на сумата



ПРИМЕР 3: КАКОВ ИЗЛЕЗ ГЕНЕРИРА СЛЕДНАТА ПРОГРАМА

```
public class TestPrograma{  
    public static void main(String[] args) {  
        int loopVal = 0;  
        while ( loopVal < 5) {  
            System.out.println("Printing Some Text" + loopVal );  
            loopVal++;  
        }  
    }  
}
```

Што ќе се случи доколку двете наредби во `while` блокот си ги заменат местата? Каков излез ќе се генерира?



ПРИМЕР 4: ДА СЕ НАПИШЕ ПРОГРАМА КОЈА ЌЕ ГИ ПЕЧАТИ СИТЕ НЕПАРНИ БРОЕВИ, ВО ОПСЕГ ОД 1 ДО 1000

```
public class Neparnibroevi1 {  
    public static void main(String[] args) {  
        int i = 1;  
        while(i<=1000){  
            if(i%2==1){  
                System.out.println("Neparen broj: " + i);  
            }  
            i++;  
        }  
    }  
}
```

```
public class Neparnibroevi2{  
    public static void main(String[] args) {  
        int i = 1;  
        while(i<=1000){  
            System.out.println("Neparen broj: " + i);  
            i=i+2;  
        }  
    }  
}
```



DO - WHILE ЈАМКА

иницијализација

do {

Блок на наредби

} while (услов) ;

- На почеток секогаш се извршува блокот на наредби, а потоа се проверува дали условот е исполнет
 - **True:** повторно изврши го блокот од наредби дадени во телото на јамката
 - **False:** излези од while циклусот и изврши ги останатите наредби, дадени после јамката
- Условот се испитува на крајот, поради што блокот на наредби се извршува **најмалку еднаш!**
- **Do-While** јамката **се повторува** се додека условот е исполнет. Во моментот кога условот нема да се исполни, јамката прекинува.



ПРИМЕР 5: КАКОВ ИЗЛЕЗ ЌЕ ГЕНЕРИРААТ СЛЕДНИТЕ ДВЕ ПРОГРАМИ

Колку пати ќе се испечати пораката: **Printing Some Text** ?

```
int loopVal = 0;
while ( loopVal < 5 ) {
    System.out.println("Printing Some Text");
    loopVal++;
}
```

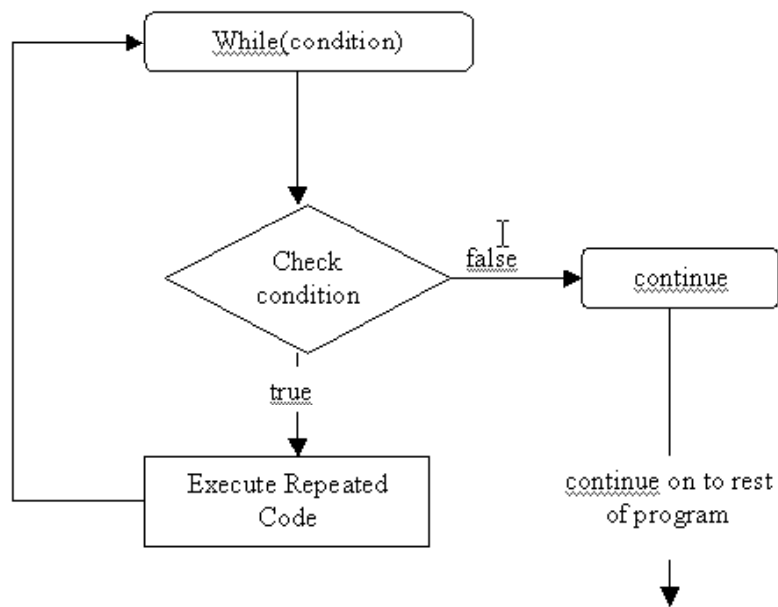
```
int loopVal = 0;
do {
    System.out.println("Printing Some Text");
    loopVal++;
}
while ( loopVal < 5 );
```

За вежба: Претходните примери може да се решат со do – while јамка!

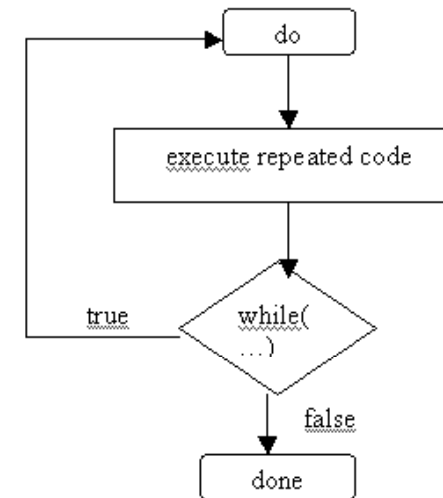


WHILE VS DO-WHILE ЦИКЛУСИ

Flow Diagram of a while loop



Flow Diagram of do .. while LOOP





WHILE VS FOR ЦИКЛУСИ

WHILE циклус

иницијализација

while (услов) {

Блок на наредби

инкремент/декремент

}

FOR циклус

for (иницијализација; услови; инкремент/декремент) {

Блок на наредби

}

- ОВИЕ НАРЕДБИ НЕМААТ ; НА КРАЈОТ!!!



ИНИЦИЈАЛИЗАЦИЈА

- Доделување на почетни вредности на бројачите
 - `for(i = 1; ...`
- Во делот за иницијализација може да се доделат вредности на повеќе бројачи
 - Бројачите кои се иницијализираат се одделуваат со запирка
 - `for(i = 1, j=0; ...`
- Делот за иницијализација од FOR циклусот, се извршува само еднаш на почетокот, пред влезот во циклусот



ПРОВЕРКА НА УСЛОВ

- Проверката на услови содржи најмалку еден логички или релациски израз
- Проверка на услови се врши пред секое повторување на блокот на наредби зададени во `for` структурата
 - `True`: Се извршува блокот на наредби од `for` структурата
 - `False`: Излези од `FOR` циклусот и изврши ги останатите наредби, дадени после јамката
- Пример за проверка на услови:
 - `for (j = 0; j < 30000; ...`
 - `for (j = 0, i=6; j < 30000 && i>5; ...`

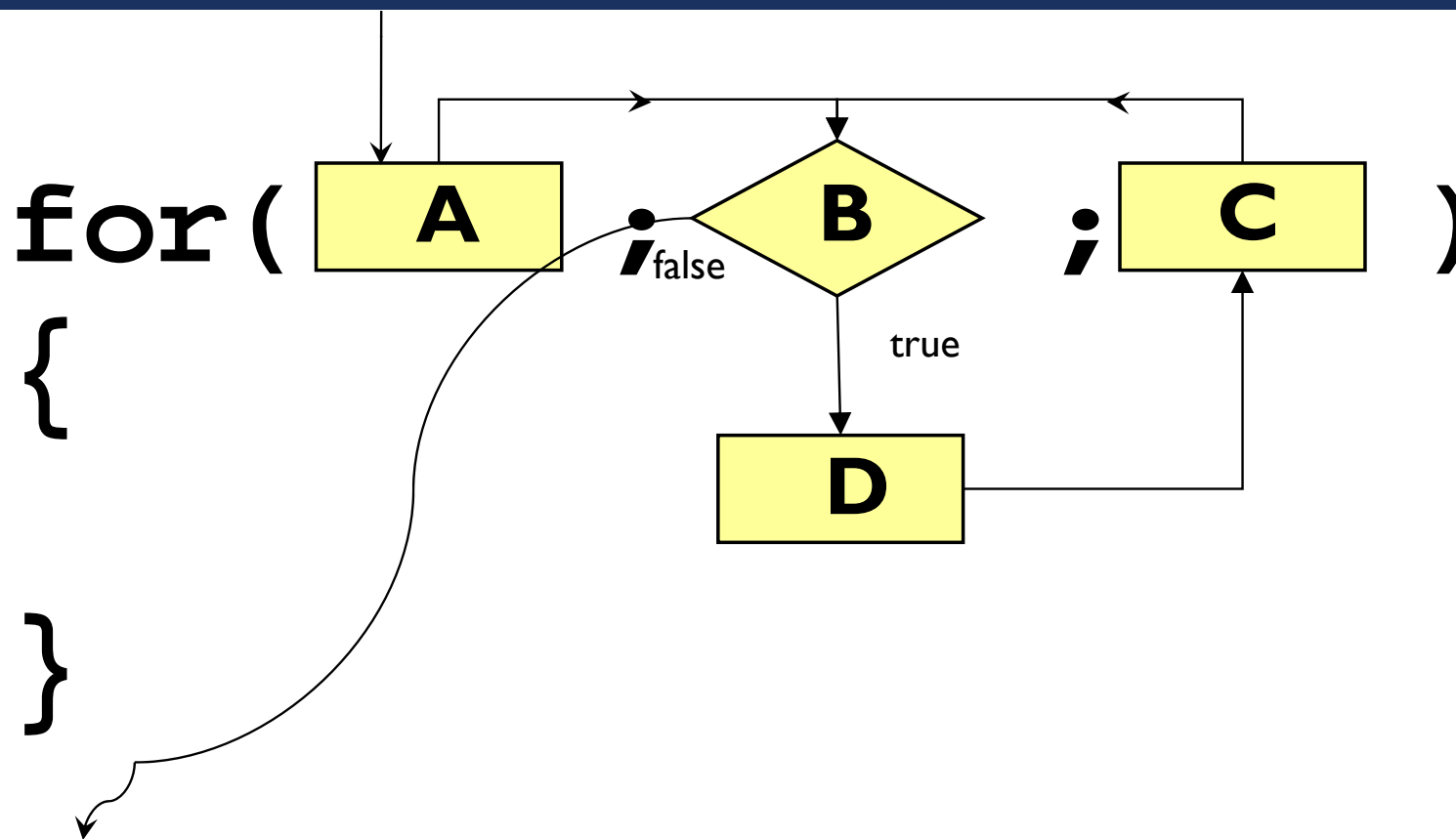


ИНКРЕМЕНТ/ДЕКРЕМЕНТ

- По завршување на блокот на наредби од for структурата, се извршува третиот дел од for наредбата
 - Инкремент/декремент на бројачи
 - `for (j = 0; j < 30000; j++)`
- Се додека условот е исполнет се повторува блокот наредби од for структурата и се врши инкремент/декремент на бројачите
- Овој чекор не мора буквално да значи инкремент/декремент
 - `for (x = 0, j = 0; j < 100; j++, x+=5)`



ИЗВРШУВАЊЕ НА FOR НАРЕДБА(1)





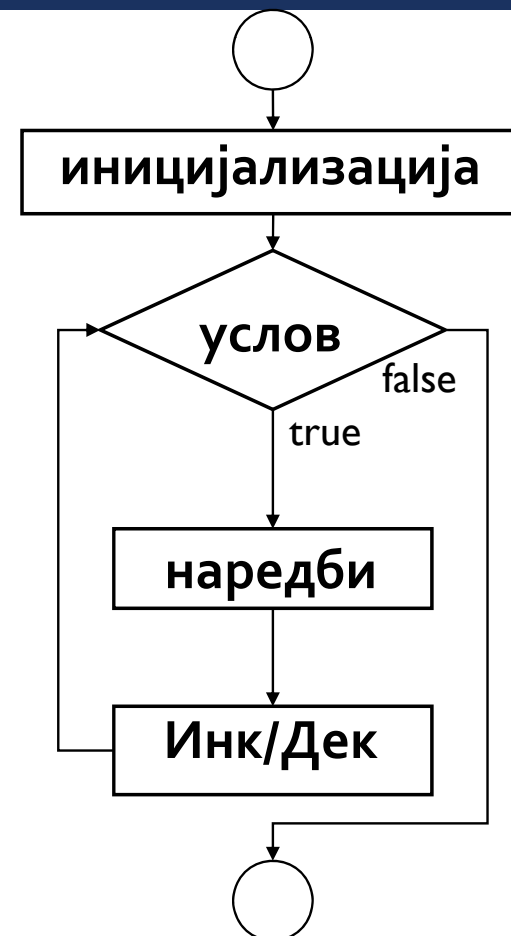
ИЗВРШУВАЊЕ НА FOR НАРЕДБА(2)

FOR циклус

for (иницијализација; услови; инкремент/декремент) {

Блок на наредби

}





ПРИМЕР 6: ДА СЕ НАПИШЕ ПРОГРАМА КОЈА ЌЕ ГИ ПЕЧАТИ БРОЕВИТЕ ОД 0 ДО 10 НА МОНИТОР

```
public class ForLoops {  
  
    public static void main(String[] args) {  
  
        int loopVal;  
        int end_value = 11;  
  
        for (loopVal = 0; loopVal < end_value; loopVal++) {  
  
            System.out.println("Loop Value = " + loopVal);  
        }  
    }  
}
```




ИЗЛЕЗ ОД ПРОГРАМАТА (ПРИМЕР6)

- Почетна вредност на бројачот во јамката: 0
Проверка на услов: бројачот треба да биде помал од 11
Инкремент/декремент: Зголемување на бројачот за 1
- Излез од програмата:

```
run:  
Loop Value = 0  
Loop Value = 1  
Loop Value = 2  
Loop Value = 3  
Loop Value = 4  
Loop Value = 5  
Loop Value = 6  
Loop Value = 7  
Loop Value = 8  
Loop Value = 9  
Loop Value = 10  
BUILD SUCCESSFUL (total time: 0 seconds)
```



СПЕЦИФИЧНИ СЛУЧАИ ВО ЈАМКИ

- Некои делови во FOR наредбата може да бидат празни

```
num = 1;
```

```
for( ; num<10; num+=2)
```

```
    System.out.println(num);
```

- Бесконечна јамка

```
for(num=15; num>10; num+=5)
```

```
    System.out.println(num);
```



ПРИМЕР 7: ДА СЕ НАПИШЕ ПРОГРАМА КОЈА ПРЕСМЕТУВА КВАДРАТ НА БРОЕВИ ОД 1 ДО НЕКОЈА МАКСИМАЛНА ВРЕДНОСТ ВНЕСЕНА ПРЕКУ ТАСТАТУРА

```
import java.util.Scanner;
public class ForLoopExample {
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    System.out.print("Input the maximum number to square: ");
    int maxNum = keyboard.nextInt();
    for(int currentNum = 1; currentNum <= maxNum; currentNum ++){
        int square = currentNum * currentNum;
        System.out.print(square + " ");
    }
    System.out.println();
}
}
```

Излез, за влез 6:

```
Input the maximum number to square: 1 4 9 16 25 36
```



ПРИМЕР8: ПРОГРАМА ВО КОЈА СЕ КОРИСТАТ ПОВЕЌЕ БРОЈАЧИ

```
public class Test {  
    public static void main(String args[]) {  
        for(int x = 10,y = 5;x < 20;x = x+1,y=y+1) {  
            System.out.print("x is: "+x+", and y is: "+y+ ",Value of x*y is : " + x*y );  
            System.out.print("\n");  
        }  
    }  
}
```

Излез:

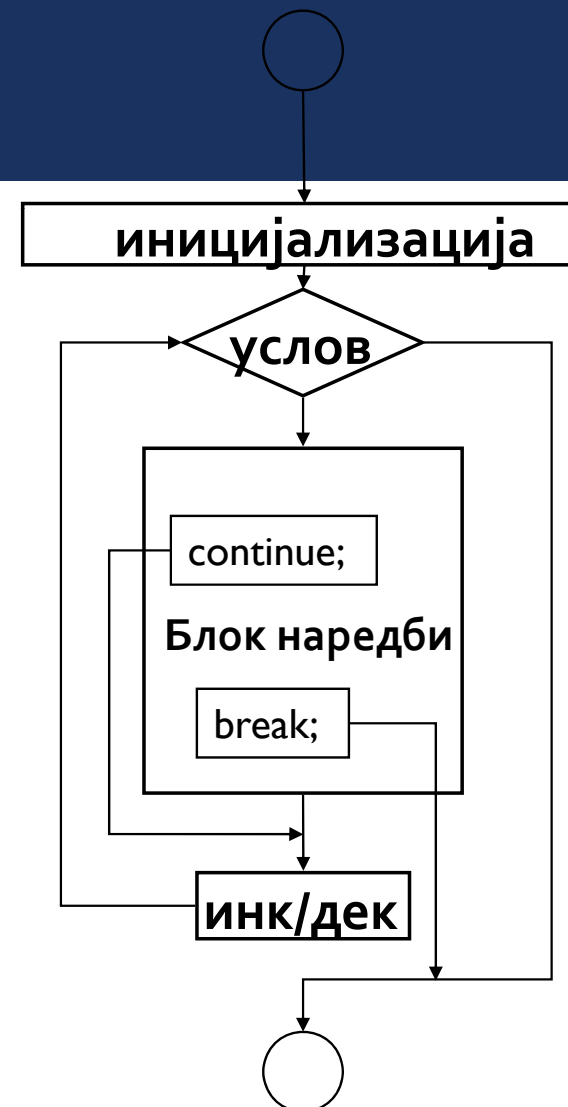
```
x is: 10, and y is: 5, Value of x*y is : 50  
x is: 11, and y is: 6, Value of x*y is : 66  
x is: 12, and y is: 7, Value of x*y is : 84  
x is: 13, and y is: 8, Value of x*y is : 104  
x is: 14, and y is: 9, Value of x*y is : 126  
x is: 15, and y is: 10, Value of x*y is : 150  
x is: 16, and y is: 11, Value of x*y is : 176  
x is: 17, and y is: 12, Value of x*y is : 204  
x is: 18, and y is: 13, Value of x*y is : 234  
x is: 19, and y is: 14, Value of x*y is : 266
```



ПРЕКИНУВАЊЕ НА ЈАМКИ

```
...  
for(иницијализација ; услов ; инк/дек)  
{  
    ...  
    break;  
    ...  
    continue;  
    ...  
}  
...
```

Блок од наредби





ПРЕКИНУВАЊЕ НА ЈАМКА СО BREAK

- Наредба BREAK
 - Овозможува **моментно излегување од циклус** реализиран со for, while, do-while или switch пред условот за напуштање на циклусот да биде исполнет.
 - Извршувањето продолжува со наредбите кои се веднаш после циклусот
- Примена
 - Предвремено завршување на јамка



ПРИМЕР ЗА ПРЕКИНУВАЊЕ НА ЈАМКА СО BREAK

```
public class Test {  
    public static void main(String args[]) {  
        for(int a = 1; a < 10; a++){  
            System.out.print(a + " ");  
            if(a == 5) {break;}  
        }  
        System.out.print("You have exited the loop");  
    }  
}
```

Излез:

```
1 2 3 4 5 You have exited the loop
```



ПРЕКИНУВАЊЕ НА ЈАМКА СО CONTINUE

■ Наредба CONTINUE

- Наредбата `continue` овозможува **директно извршување на следниот чекор на јамката** запоставувајќи ги наредбите до крајот на јамката.
- Кај **while** и **do/while** циклуси по извршување на **CONTINUE** веднаш се извршува проверка на условот од наредбата
- Кај **FOR** циклуси по извршување на **CONTINUE** веднаш се извршува `increment/dekrement` изразот, а потоа се проверува условот



ПРИМЕР ЗА ПРЕКИНУВАЊЕ НА ЈАМКА СО CONTINUE

```
public class TestExample {  
    public static void main(String[] args) {  
        for(int a = 1; a < 10; a++){  
            if(a == 5){continue;}  
            System.out.print(a + " ");  
        }  
    }  
}
```

Излез:

1 2 3 4 6 7 8 9

Се забележува дека на излез не се печати бројот 5



ПРИМЕР 9: КРЕИРАЊЕ НА ТРИАГОЛНИК СО БРОЕВИ, СО ПОМОШ НА ВГНЕЗДЕНИ FOR ЈАМКИ

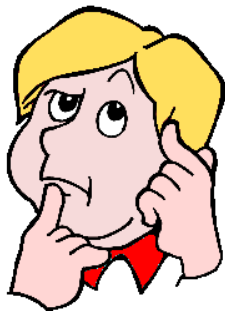
```
public class ForLoopExample {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 6; i++) {  
            for (int j = 1; j <= i; j++)  
                System.out.print(j + " ");  
            System.out.println();  
        }  
    }  
}
```

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6
```



ВЕЖБАЊЕ НА ЗАДАЧИ

- Најпрво треба да се размислува како да се реши проблемот, а потоа да се програмира!
- Најдобро се учи со вежбање и пробување на компјутер
- Паметни софтверски алатки



```
void foo(int a) {  
    if (a == 1) {  
        ...  
    } else if (a == 2 || a == 3) {  
        System.out.println("2 or 3");  
    } else if (a == 4) {  
        System.out.println("4");  
    } else {  
        System.out.println("default");  
    }  
}
```

- Convert 'if-else' to 'switch'
- Invert 'if' statement
- Change 'if' block to statement
- Change 'if-else' blocks to statements

```
...  
void foo(int a) {  
    switch (a) {  
    case 1:  
        System.out.println("1");  
        break;  
    case 2:  
    case 3:  
        System.out.println("2 or 3");  
        break;  
    }  
}
```

Press 'Tab' from prog



ЗАДАЧА₁: ДА СЕ НАПИШЕ ПРОГРАМА КОЈА ЧИТА БРОЈ N ОД ТАСТАТУРА И ПОТОА ПРЕСМЕТУВА СУМА НА N-ТЕ ЕЛЕМЕНТИ, НИВНИОТ КВАДРАТ И НИВНИОТ КУБ

```
import java.util.Scanner;

public class WhileLoopExample {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Input n: ");
        int n = keyboard.nextInt();

        int currentNum = 1;
        int sum = 0;
        int sumSquare = 0;
        int sumCube = 0;
        while (currentNum <= n)
        {
            sum = sum + currentNum;
            sumSquare = sumSquare + currentNum*currentNum;
            sumCube = sumCube + currentNum*currentNum*currentNum;
            currentNum = currentNum + 1;
        }
        System.out.println("The sum of the numbers from 1 to " + n + " is " + sum + ".");
        System.out.println("The sum of the squares of the numbers from 1 to " + n + " is " +
sumSquare + ".");
        System.out.println("The sum of the cubes of the numbers from 1 to " + n + " is " +
sumCube + ".");
    }
}
```

Излез, за влез 6:

```
Input n: The sum of the numbers from 1 to 6 is 21.
The sum of the squares of the numbers from 1 to 6 is 91.
The sum of the cubes of the numbers from 1 to 6 is 441.
```



ЗАДАЧА2: ДА СЕ НАПИШЕ ПРОГРАМА КОЈА ПРЕСМЕТУВА СУМА НА ЕЛЕМЕНТИ КОИ СЕ ВНЕСУВААТ ОД ТАСТАТУРА. ВЛЕЗОТ НА БРОЕВИТЕ ЗАВРШУВА КОГА КОРИСНИКОТ ЌЕ ВНЕСЕ -1

```
import java.util.Scanner;

public class WhileLoopExample {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);

        int currentNum = 0;
        int sum = 0;
        while (currentNum != -1)
        {
            sum = sum + currentNum;

            System.out.print("Input the next positive number to add (-1 to quit): ");
            currentNum = keyboard.nextInt();
        }
        System.out.println("The sum of the input numbers is " + sum + ".");
    }
}
```



ЗАДАЧА3: ДА СЕ НАПИШЕ ФУНКЦИЈА КОЈА ЌЕ ПРОВЕРУВА ДАЛИ НЕКОЈ БРОЈ Е ПРОСТ

```
import java.util.Scanner;
public class FindPrime {
public static void main(String args[]) {
    Scanner keyboard = new Scanner(System.in);
    System.out.print("Input n: ");
    int n = keyboard.nextInt();

    if(isprimenumber(n))
        System.out.println("Prime");
    else
        System.out.println("Not Prime");
}
public static boolean isprimenumber(int num){
    boolean isPrime = true;
    for(int i=2; i < num/2; i++) {
        if((num % i) == 0) {
            isPrime = false;
            break;
        }
    }
    return isPrime;
}
}
```



ЗАДАЧА4:

Да се напише програма која во зависност од влезот на корисникот (1 или 2) пресметува математички операции за правоаголник или круг, соодветно:

Влез 1:

Треба да се внесе висина и ширина на правоаголник и потоа да се пресмета плоштина и периметар на правоаголникот

Влез 2:

Треба да се внесе радиус на круг и потоа да се пресмета плоштина и периметар на круг

Програмата се извршува се додека корисникот не внесе валиден влез (1 или 2)

```
import java.util.Scanner;
public class WhileLoopExample {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);

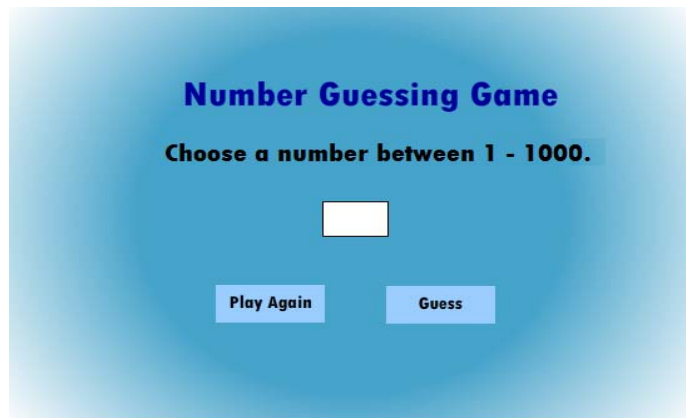
        int noMore = 0;
        while (noMore == 0)
        {
            System.out.println("Please select from the following menu:");
            System.out.println("1. Rectangle Properties");
            System.out.println("2. Circle Properties");
            System.out.println();
            System.out.print("Selection: ");
            int menuOption = keyboard.nextInt();
            System.out.println();

            if (menuOption == 1){
                System.out.print("Input the width of the rectangle: ");
                double width = keyboard.nextDouble();
                System.out.print("Input the height of the rectangle: ");
                double height = keyboard.nextDouble();
                double area = height * width;
                double perimeter = 2*height + 2*width;
                System.out.println("The area of the rectangle is " + area);
                System.out.println("The perimeter of the rectangle is " + perimeter);
                noMore = 1;
            }
            else if (menuOption == 2){
                System.out.print("Input the radius of the circle: ");
                double radius = keyboard.nextDouble();
                double area = Math.PI * Math.pow(radius, 2);
                double circumference = 2 * Math.PI * radius;
                System.out.println("The area of the circle is " + area);
                System.out.println("The circumference of the circle is " + circumference);
                noMore = 1;
            }
            else{
                System.out.println("Invalid Menu Selection!");
                System.out.println("Please make another selection.");
                System.out.println();
            }
        }
    }
}
```



ЗАДАЧА 5: ИГРА

Да се напише програма за играње со која се погодува случајно генериран број во осег од 0 до 1000. Корисникот внесува броеви преку тастатура се додека не го погоди случајно генерираниот број. Во моментот кога бројот ќе биде погоден играта завршува.



```
import java.util.Random;
import java.util.Scanner;

public class GuessingGame {

    public static void main(String[] args) {

        Random rand = new Random();
        int numberToGuess = rand.nextInt(1000);
        int numberOfTries = 0;
        Scanner input = new Scanner(System.in);
        int guess;
        boolean win = false;

        while (win == false) {

            System.out.println("Guess a number between 1 and 1000: ");
            guess = input.nextInt();
            numberOfTries++;

            if (guess == numberToGuess) {
                win = true;
            }
            else if (guess < numberToGuess) {
                System.out.println("Your guess is too low");
            }
            else if (guess > numberToGuess) {
                System.out.println("Your guess is too high");
            }
        }

        System.out.println("You win!");
        System.out.println("The number was " + numberToGuess);
        System.out.println("It took you " + numberOfTries + " tries");
    }
}
```