

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        // Hello world program  
        System.out.println("Hello world!");  
    }  
}
```

Razred HelloWorld vsebuje metodo main, ki je glavna metoda programa.

Metoda main je obvezna. Obstajati mora samo ena metoda main v programu.

// pomeni, da je preostanek vrste pripomba.

Ostalo bomo razložili kasneje.

```
public class Hanoi {  
  
    public static void main(String[] args) {  
        // Hanojski stolpi  
        final int N = 3;  
        System.out.println("Start");  
        hanoi(N,1,2);  
        System.out.println("Finish");  
    }  
  
    public static void hanoi(int n, int start, int end) {  
        if (n > 0) {  
            int middle = other(start,end);  
            hanoi(n-1,start,middle);  
            System.out.println("move disk " + n + " from " + start  
                               + " to " + end);  
            hanoi(n-1,middle,end);  
        };  
    }  
  
    public static int other(int x, int y) {  
        return 6 - (x + y);  
    }  
  
}
```

Spremenljivke imajo vedno tipe, ki so specifikirani v deklaracijah spremenljivk.

- ▶ `int middle = other(start,end)`

Spremenljivka `middle` ima tip `int` in začetno vrednost `other(start,end)`. (Metoda `other` vrne vrednost tipa `int`.)

- ▶ `int start`

Spremenljivka `start`, ki je parameter metoda `hanoi`, ima tip `int`.

- ▶ `final int N = 3`

`N` je končna konstanta tipa `int`, ki ima vrednost 3. Končna pomeni, da se vrednost ne more spremeniti.

Primitivni tipi

boolean	1-bit truth values	true, false
char	16-bit characters	'a', 'A', '0', '#', ...
byte	8-bit integers	..., -1, 0, 1, ..., 127
short	16-bit integers	..., -1, 0, 1, ..., 32767
int	32-bit integers	..., -1, 0, 1, ...
long	64-bit integers	..., -1, 0, 1, ...
float	32-bit floating point numbers	...-3.14 ...0.0 ...3.14 ...
double	64-bit floating point numbers	...-3.14 ...0.0 ...3.14 ...

- ▶ `int other(int x, int y)`

Metoda `other` ima dva parametra, in sicer `x` in `y`, oba imata tip `int`. Metoda deluje kot funkcija in vrne vrednost tipa `int`.

- ▶ `void hanoi(int n, int start, int end)`

Metoda `hanoi` ima tri parametre `n`, `start`, `end`, ki imajo vsi tip `int`. Metoda ne vrne vrednosti.

`void` je poseben `return type`, ki pomeni, da metoda ne vrne vrednosti.

```
public class Hanoi {  
  
    public static void main(String[] args) {  
        // Imperative Hanoi  
        final int N = 3;  
        System.out.println("Start");  
        int numSteps = (int) Math.pow(2, N);  
        for (int step = 1; step < numSteps; step++) {  
            int disk = findDisk(step);  
            int direction = (disk+N)%2;  
            System.out.print("Move disk " + disk);  
            if (direction == 0)  
                {System.out.println(" clockwise");}  
            else {System.out.println(" anticlockwise");}  
        }  
        System.out.println("Finish");  
    }  
  
    public static int findDisk (int step) {  
        if (step%2 ==1) {return 1;}  
        else {return 1 + findDisk(step/2);}  
    }  
}  
  
}
```

Ukazi (Statements)

Posamezni ukazi (individual statements) se zaključijo s podpičjem (semicolon) ;

```
System.out.println("Start");
```

```
int disk = findDisk(step);
```

Blok ukazov je ukaz, ki se sestavlja zaporedja ukazov med zavitimi oklepaji { ... }

```
{final int N = 3;
  System.out.println("Start");
  hanoi(N,1,2);
  System.out.println("Finish");
}
```

Kontrolne strukture

- ▶ `if (condition) truebranch`
- ▶ `if (condition) truebranch else falsebranch`
- ▶ `for (initialization; condition; step) body`
- ▶ `while (condition) body`
- ▶ `do body while (condition)`

Pravila poimenovanja (naming conventions)

- ▶ `PrimerRazreda1`

Razredi se začnejo z veliko črko. Vsaka nova beseda v imenu se začne z veliko črko.

- ▶ `primerSpremenljivke1`

Spremenljivke se začnejo z majhno črko. Vsaka nova beseda v imenu se začne z veliko črko.

- ▶ `PRIMER_KONSTANTE1`

Konstante deklarirane kot `final` se pišejo z velikimi črkami, besede pa so ločene s podčrtaji.