



# Osnove C#

## Poletna šola FRI

Laboratory for Computer Graphics and Multimedia  
Faculty of Computer and Information Science  
University of Ljubljana, Slovenia



# Unity Hello World!

## Primer preprostega programa:

```
using UnityEngine;  
  
public class Main : MonoBehaviour {  
    void Start () {  
        Debug.Log("Hello World !!");  
    }  
}
```

## Rezervirane besede

- so besede, ki so del jezika, in jih ne moremo uporabiti v drugačne namene
- besede določajo pomen posameznih delov programov, tipe spremenljivk, funkcij, potek izvajanja, ...
- primeri:
  - `int, double, float, ...`
  - `private, public, ...`
  - `while, for, ...`
  - `if, switch, ...`

## Osnovni podatkovni tipi

- bitni:
  - byte (`byte bit = 100;`)
- celoštevilski:
  - short, int, long (`int n = 90;`)
- decimalni:
  - float, double (`double d = 0.1;`)
- znakovni:
  - char (`char c = 'a';`)
- logični:
  - bool (`bool flag = false;`)

## Sestavljeni podatkovni tipi

- polja:

- `int n[] = new int[100];`

- `double stevilke[] = {0.1, 0.4, 0.3};`

- `char abeceda[] = {'a', 'b', 'c', 'd'};`

- razredi (string, Object, Mathf, ...)

- `string abc = "abeceda";`

- `abc == "abeceda";` (vrne true)

- `abc != "abeceda";` (vrne false)

- `string[] besede = new string[100];`

## Operatorji

- aritmetični
- inkrementacija, dekrementacija
  - `++x; x++; --x; x--;`
  - seštevanje, odštevanje, množenje, deljenje, modula
  - `z=x+y; z=x-y; z=x*y; z=x/y; z=x%y;`
- logični
- logični in, logični ali, negacija
  - `a && b      a || b      !a`
- primerjalni
- je enako, ni enako, večji, manjši, primerek razreda
- `a == b    a != b      x < y    x > y`
  - `abc is string`

## Operatorji

- bitni
  - bitni in, ali, xor, pomik v levo, pomik v desno, eniški komplement
    - `c = a&b; c = a|b; c = a^b; x = x<<1;`
- prireditveni
- prireditev, prireditev z operacijo
  - `x = 5; x += 5; x -= 5; x *= 5; x /= 5; x%=5;`
- posebni
- dostop do elementov razreda
  - `d = Mathf.Abs(x); d = Mathf.Sin(x);`
  - dostop do elementov polja
  - `x = stevila[10];`
  - pretvorba tipa
  - `d = (double)x;`

## Stavki

- osnovni stavek
  - `stavek;`
- prazni stavek
  - `;`
- blok stavkov
  - `{`
    - `stavek1;`
    - `stavek2;`
  - `}`



## Stavki

- odločitveni stavki

```
if (pogoj)
    stavek1;
else if (pogoj2)
    stavek2;
else
    stavek3;
```

## Stavki

- odločitveni stavki

```
switch (byte, short, char, ali int spr.)  
case vred1:  
    stavek1;  
    break;  
case vred2:  
    stavek2;  
  
...  
default:  
    stavek;
```

## Zanke

- while zanka

```
while (pogoj)
    stavek;
```

- for zanka

```
for (inicijalizacija; pogoj; korak)
    stavek;
```

- do while zanka

```
do{
    stavek;
} while (pogoj);
```

- dodatne možnosti

```
break, continue
```

## Izjeme

```
try {  
    stavek;  
} catch (Exception e) {  
    // kaj narediti ob izjemi;  
} finally {  
    // kaj narediti v vsakem primeru;  
}
```

## Metode

- Metoda, ki ne vrne ničesar:

```
(dostopnost) void Ime(parametri) {  
    stavek;  
}
```

- Metoda, ki na koncu vrne določen tip:

```
(dostopnost) tip Ime(parametri) {  
    return tip;  
}
```

Dostopnosti: **public**, **protected** ali **private**

## Komentarji

Uporabimo jih, da si zabeležimo kaj določena koda naredi, ali pa da onemogočimo določen del kode namesto da bi jo izbrisali

- **Enovrstični komentar:**

- `//Debug.Log("te kode ne bo izpisalo");`
- 

- **Večvrstični komentar:**

- `int x = 2;`
- `/* x = 10;`
- `x++; */`