

## Pisanje koda

### Osnovna sintaksna pravila

Izraz

varijabla

Operator

=

Vrijednost/Izraz

7;

← Oznaka kraja naredbe

! UVIJEK BITI BEZ SINTAKSNE GREŠKE !

```
varijabla = 4;
```

```
int varijabla = 4
```

Upozorenje

```
int varijabla = 4;
```

# Variable

Varijabla je prostor u memoriji

C# uz naziv zahtjeva za svaku varijablu poznavanje tipa podatka  
Na primjeru se koristi tip podatka int – cijeli broj

Gledajte na varijablu kao na „kutiju cipela”

## 1. Deklaracija

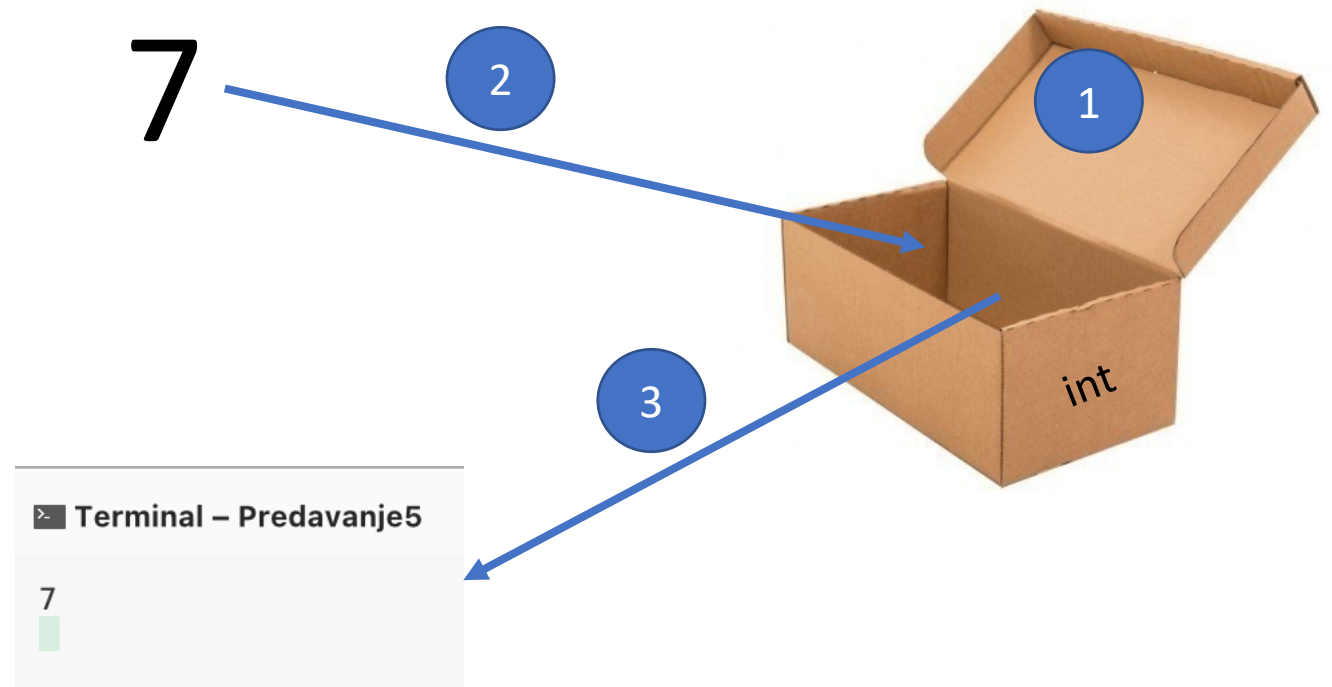
```
int varijabla;
```

## 2. Dodjeljivanje

```
varijabla = 7;
```

## 3. Korištenje

```
Console.WriteLine(varijabla);
```



## Varijable

Primjeri različitih načina deklaracija i korištenja varijabli

Kod

```
int i, j, k; i = 1; j = 1; k = 1;

int t = 1, m = 2;

int x, y = 1;

x = i + j;

bool logickaVrijednost = t == 1;

Console.WriteLine("{0},{1},{2},{3},{4},{5},{6},{7}",
    i, j, k, t, m, x, y, logickaVrijednost);
```

Rezultat izvođenja

```
1,1,1,1,2,2,1,True
```

## Pseudo kod

### Naredbe

ulaz (r)

$\pi = 3.14$

$o = 2 * r * \pi$

$p = r * r * \pi$

izlaz (o, p)

### Opis naredaba

Od korisnika traži unošenje vrijednosti i unesenu vrijednost pohranjuje u varijablu  $r$ .

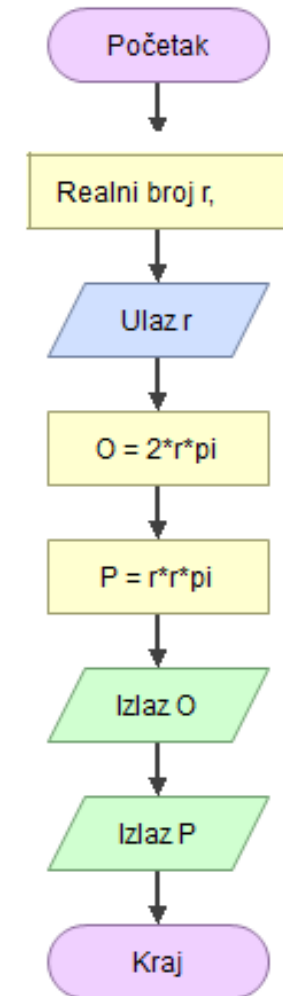
Varijabli  $\pi$  pridružuje vrijednost 3.14.

Računa vrijednost izraza  $i$  tu vrijednost pridružuje varijabli  $o$ .

Računa vrijednost izraza  $i$  dobivenu vrijednost pridružuje varijabli  $p$ .

Vrijednosti varijabli  $o$  i  $p$  ispisuje na zaslon.

## Blok dijagram



Izvor:

<https://loomen.carnet.hr/mod/book/view.php?id=645633&chapterid=127911>

## Varijable

### Implementacija zadatka u C# (IDE Visual Studio Mac)

#### Kod

```
double r = Double.Parse(Console.ReadLine());  
  
double pi = 3.14;  
  
double o = 2 * r * pi;  
  
double p = r * r * pi;  
  
Console.WriteLine("o={0},p={1}", o, p);
```

#### Rezultat izvođenja

```
> Terminal – Predavanje5  
  
2.75  
o=17.27,p=23.74625  
█
```

# Operatori

## Kategorije operatora

Operator dodjeljivanja =

Aritmetički operatori (Arithmetic operators)

unarni ( ++, --, +, - ) | binarni ( \* , / , % , + , - )

Operatori usporedbe (Comparison operators)

==, <, >, <=, >=

Logički operatori (Boolean logical operators)

!, &, &&, |, ||, ^

Operatori na bitovima (Bitwise and shift operators)

~, <<, >>, &, |, ^

Operatori jednakosti (Equality operators)

==, !=

## Primjeri koda

```
int i=2,j=3;
```

```
int k..= i + j;
```

```
k..= i - j;
```

```
k = i * j;
```

```
double t..= k / j;
```

```
bool b..= k == j;
```

```
b..= k > j;
```

```
b..= k != j;
```

```
bool p = true, r = false;
```

```
b..= p & r;
```

```
b..= p && r;
```

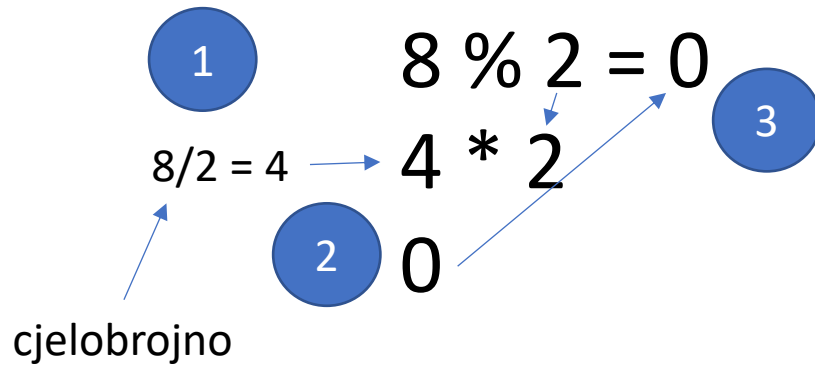
Izvor:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/>

## Operatori

### Modulo (%)

Ostatak nakon cjelobrojnog dijeljenja



### Increment i decrement

```
int i = 0;
//Increment
// prvo koristi vrijednost varijable
// pa ju onda uveća
i++; //i = 1
//i=1
++i; // i=2
//i=2

//Decrement
// prvo koristi vrijednost varijable
// pa ju onda umanjuje
i--; //i = 1
//i=1
--i; // i=0
// i=0
Console.WriteLine(i);
```

Tipovi podataka

Kategorije tipova podataka

Ugrađeni (Built-in) 

Referencirani (Reference)



| C# type keyword | .NET type     |
|-----------------|---------------|
| object          | System.Object |
| string          | System.String |
| dynamic         | System.Object |

\*referencirane obrađujemo u dijelu OOP

Izvor:  
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/built-in-types>

| C# type keyword | .NET type      |
|-----------------|----------------|
| bool            | System.Boolean |
| byte            | System.Byte    |
| sbyte           | System.SByte   |
| char            | System.Char    |
| decimal         | System.Decimal |
| double          | System.Double  |
| float           | System.Single  |
| int             | System.Int32   |
| uint            | System.UInt32  |
| long            | System.Int64   |
| ulong           | System.UInt64  |
| short           | System.Int16   |
| ushort          | System.UInt16  |



## Tipovi podataka

### Moguće vrijednosti cjelobrojnih tipova podataka

| C#<br>type/keyword  | Range                                                   | Size                    | .NET type                     |
|---------------------|---------------------------------------------------------|-------------------------|-------------------------------|
| <code>sbyte</code>  | -128 to 127                                             | Signed 8-bit integer    | <a href="#">System.SByte</a>  |
| <code>byte</code>   | 0 to 255                                                | Unsigned 8-bit integer  | <a href="#">System.Byte</a>   |
| <code>short</code>  | -32,768 to 32,767                                       | Signed 16-bit integer   | <a href="#">System.Int16</a>  |
| <code>ushort</code> | 0 to 65,535                                             | Unsigned 16-bit integer | <a href="#">System.UInt16</a> |
| <code>int</code>    | -2,147,483,648 to 2,147,483,647                         | Signed 32-bit integer   | <a href="#">System.Int32</a>  |
| <code>uint</code>   | 0 to 4,294,967,295                                      | Unsigned 32-bit integer | <a href="#">System.UInt32</a> |
| <code>long</code>   | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 | Signed 64-bit integer   | <a href="#">System.Int64</a>  |
| <code>ulong</code>  | 0 to 18,446,744,073,709,551,615                         | Unsigned 64-bit integer | <a href="#">System.UInt64</a> |

Izvor:

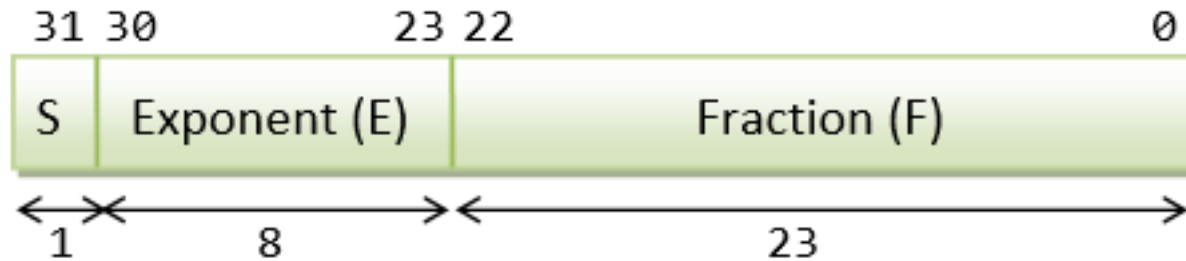
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/integral-numeric-types>

## Tipovi podataka

Moguće vrijednosti tipova podataka s pomičnom točkom (decimalni brojevi)

| C# type/keyword      | Approximate range                                        | Precision     | Size     | .NET type                      |
|----------------------|----------------------------------------------------------|---------------|----------|--------------------------------|
| <code>float</code>   | $\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$    | ~6-9 digits   | 4 bytes  | <a href="#">System.Single</a>  |
| <code>double</code>  | $\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$  | ~15-17 digits | 8 bytes  | <a href="#">System.Double</a>  |
| <code>decimal</code> | $\pm 1.0 \times 10^{-28}$ to $\pm 7.9228 \times 10^{28}$ | 28-29 digits  | 16 bytes | <a href="#">System.Decimal</a> |

Pomična točka



**32-bit Single-Precision Floating-point Number**

Izvor:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/integral-numeric-types>

<https://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html>

## Tipovi podataka – primjeri korištenja

```
bool logickaVrijednost = true;
byte najmanjiCijeliBroj = 1;
sbyte najmanjiPozitivniCijeliBroj = 0;
short drugiPoReduCijeliBroj = -1;
ushort drugiPoReduPozitivniCijeliBroj = 1;
int treciPoReduCijeliBroj = -1;
uint treciPoReduPozitivniCijeliBroj = 1;
long najvećiCijeliBroj = -1;
ulong najvećiPozitivniCijeliBroj = 1;
// M/m jer je D/d za double pa je m
// sljedeći smisleni znak u nazivu decimal
decimal najvećiDecimalniBroj = 3.14m;
// D/d kao double iako na double može i bez oznake
double srednjiDecimalniBroj = 3.14;
// F/f kao float
float najmanjiDecimalniBroj = 3.14f;
char znak = 'T';
Console.WriteLine("{0}, {1}, {2}, {3}, {4}, {5}, " +
    "{6}, {7}, {8}, {9}, {10}, {11}, {12}", logickaVrijednost,
    najmanjiCijeliBroj, najmanjiPozitivniCijeliBroj,
    drugiPoReduCijeliBroj, drugiPoReduPozitivniCijeliBroj,
    treciPoReduCijeliBroj, treciPoReduPozitivniCijeliBroj,
    najvećiCijeliBroj, najvećiPozitivniCijeliBroj,
    najmanjiDecimalniBroj, srednjiDecimalniBroj,
    najvećiDecimalniBroj, znak);
```

### > Terminal – Predavanje5

```
True, 1, 0, -1, 1, -1, 1, -1, 1, 3.14, 3.14, 3.14, T
```

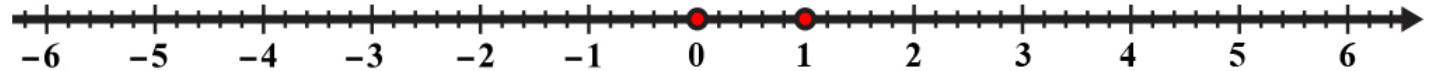
## Maksimalna i minimalna moguća vrijednosti

```
long i = Int64.MaxValue;  
Console.WriteLine(i);  
i++;  
Console.WriteLine(i);|
```

> Terminal – Predavanje5

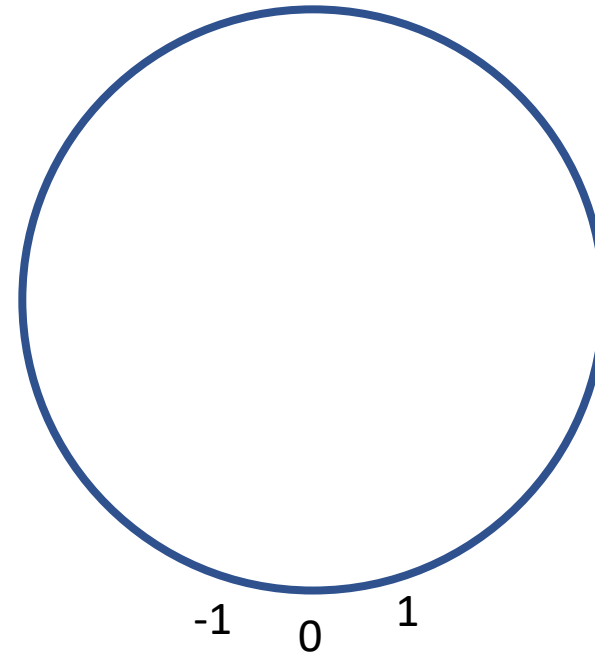
```
9223372036854775807  
-9223372036854775808  
█
```

Uobičajeno u matematici brojevi se prikazuju na brojevnom pravcu



Brojevi se ponašaju kao brojevi na brojevnoj kružnici

-9223372036854775808    9223372036854775807



## Variable

U C# postoji 7 kategorija varijabli

1. Statične varijable (static variables)
2. Varijable instance (instance variables)
3. Elementi niza (array elements)
4. Parametar vrijednosti (value parameters)
5. Parametar reference (reference parameters)
6. Izlazni parameter (output parameters)
7. Lokalna varijabla (local variables)

### Terminal – Predavanje5

1. Vrijednost statične varijable je 1
2. Vrijednost varijable instance je 2
3. Vrijednost varijable niza je 3
4. Vrijednost parametra vrijednosti je 4
5. Vrijednost parametra reference je 5
6. Vrijednost izlaznog parametra je 6
7. Vrijednost lokalne varijable je 7

### Izvori:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/variables>

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>

```
1 using System;
2
3 namespace Predavanje5
4 {
5     class Variable
6     {
7         static int staticnaVarijabla = 1;
8         int varijablaInstance = 2;
9         public Variable()
10         {
11             Console.WriteLine("2. Vrijednost varijable instance je "
12                               + varijablaInstance);
13             int[] niz = { 3, 3, 3 };
14             Console.WriteLine("3. Vrijednost varijable niza je "
15                               + niz[0]);
16             int parametarReference=0, parametarVrijednosti = 4,
17                 izlazniParametar;
18             metoda(ref parametarReference, parametarVrijednosti,
19                   out izlazniParametar);
20
21             Console.WriteLine("5. Vrijednost parametra reference je "
22                               + parametarReference);
23
24             Console.WriteLine("6. Vrijednost izlaznog parametra je "
25                               + izlazniParametar);
26
27             int lokalnaVarijabla = 7;
28             Console.WriteLine("7. Vrijednost lokalne varijable je "
29                               + lokalnaVarijabla);
30         }
31
32         void metoda(ref int parametarReference,
33                    int parametarVrijednosti,
34                    out int izlazniParametar)
35         {
36             Console.WriteLine("4. Vrijednost parametra vrijednosti je "
37                               + parametarVrijednosti);
38             parametarReference = 5;
39             izlazniParametar = 6;
40         }
41
42         static void Main(string[] args)
43         {
44             Console.WriteLine("1. Vrijednost statične varijable je "
45                               + staticnaVarijabla );
46             new Variable();
47         }
48     }
49 }
```