

Petlje

for

Uvjet završetka se provjerava prije ulaska u petlju

Tijelo petlje se ne mora izvesti niti jednom

for petlji je potrebna varijabla

`int i=0`

vrijednost se može uvećavati ili umanjivati

`i++`

```
for (početna vrijednost; uvjet završetka; promjena vrijednosti)  
{  
    // tijelo petlje - naredbe koje se ponavljaju  
}
```

uvijek završetka je logička vrijednost (bool)

`x<10`

Izvor:

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

Petlje

for

```
for(int i = 0; i < 10; i++)  
{  
    Console.WriteLine("Osijek");  
}
```

```
int j;  
for (j = 0; j < 10; j++)  
{  
    Console.WriteLine("Osijek");  
}
```

```
for (j = 10; j > 0; j--)  
{  
    Console.WriteLine("Osijek");  
}
```

```
for(int k = 0; k < 20; k += 2)  
{  
    Console.WriteLine("Osijek");  
}
```

Terminal – Predavanje5

```
Osijek  
Osijek  
Osijek  
Osijek  
Osijek  
Osijek  
Osijek  
Osijek  
Osijek  
Osijek
```

Varijabla u petlji mijenja vrijednost

```
for(int x = 0; x < 10; x++)  
{  
    Console.WriteLine(x+1);  
}
```

```
bool uvjet = true;  
for(int x = 1; uvjet; x++)  
{  
    Console.WriteLine(x);  
    uvjet = x < 10;  
}
```

Terminal – Predavanje5

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Petlje

for

Ugnježdživanje petlje

```
for(int i = 0; i < 10; i++)
{
    for(int j = 0; j < 10; j++)
    {
        Console.Write("{0} ", (i + 1) * (j + 1));
    }
    Console.WriteLine();
}
```

Terminal – Predavanje5

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

```
string s;
for(int i = 0; i < 10; i++)
{
    for(int j = 0; j < 10; j++)
    {
        var b = (i + 1) * (j + 1);
        s = "    " + b;
        Console.Write("{0}", s[^4|.]);
    }
    Console.WriteLine();
}
```

Terminal – Predavanje5

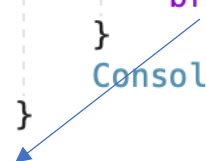
```
1  2  3  4  5  6  7  8  9 10
2  4  6  8 10 12 14 16 18 20
3  6  9 12 15 18 21 24 27 30
4  8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

Petlje

for

break nasilno prekidanje petlje – ne dolazi se do uvjeta izlaska iz petlje


```
for(int i = 0; i < 10; i++)  
{  
    if (i == 3)  
    {  
        break;  
    }  
    Console.WriteLine(i);  
}
```



> Terminal – Predavanje5

```
0  
1  
2
```

```
for (int i = 0; i < 5; i++)  
{  
    for (int j = 0; j < 5; j++)  
    {  
        if (j == 2)  
        {  
            goto labela;  
        }  
        Console.WriteLine("i={0},j={1}", i, j);  
    }  
}  
labela::
```



> Terminal – Predavanje5

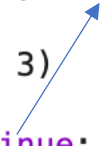
```
i=0,j=0  
i=0,j=1
```

Petlje

for

continue nastavak izvođenja (preskakanje petlje)


```
for(int i = 0; i < 10; i++)
{
    if (i == 3)
    {
        continue;
    }
    Console.WriteLine(i);
}
```



> Terminal – Predavanje5

```
0
1
2
4
5
6
7
8
9
```

```
for (int i = 0; i < 5; i++)
{
    for (int j = 0; j < 5; j++)
    {
        if (j == 2)
        {
            goto labela;
        }
        Console.WriteLine("i={0},j={1}", i, j);
    }
    labela:;
}
```



> Terminal – Predavanje5

```
i=0,j=0
i=0,j=1
i=1,j=0
i=1,j=1
i=2,j=0
i=2,j=1
i=3,j=0
i=3,j=1
i=4,j=0
i=4,j=1
```

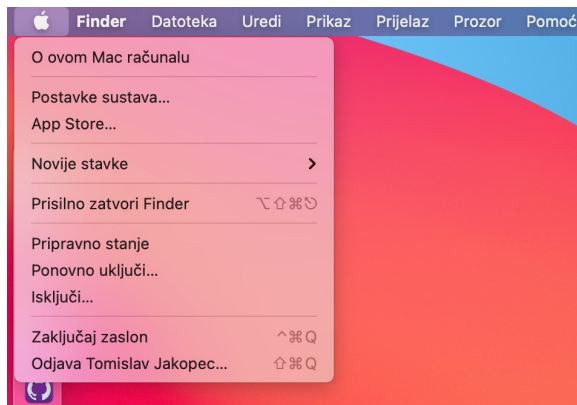
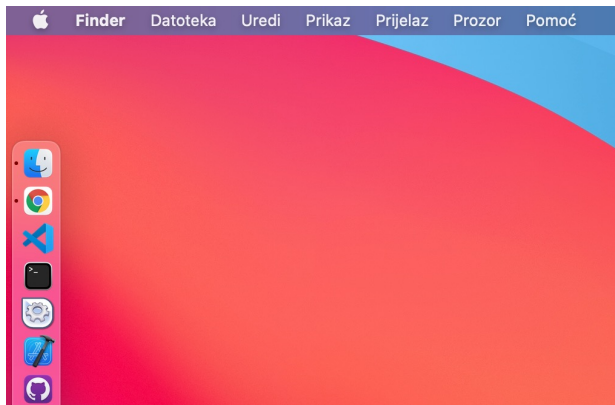
Petlje

for

beskonačna petlja

Jedan od osnovnih principa programiranja

Kako OS „zna“ da sam napravio klik?



Beskonačko pita da li si napravio klik?

Kada „konačno“ napravim klik prikaže izbornik

```
int i = 0;
for(;; )
{
    if (i++ > 10)
    {
        break;
    }
    Console.WriteLine("Osijek");
}
```

Petlje

primjer korištenja for petlje – što radi prikazani kod?

```
int i, s = 0; for (i = 1; i <= 100; i++) s += i;

int i, s; for (i = 1, s = 0; i <= 100; s += i, i++) ;

int i = 1, s = 0; for (; i <= 100; i++) { s += i; }

int i, s = 0; for (i = 1; ; i++) { if (i <= 100) s += i; else break; }

int i, s = 0; for (i = 1; i <= 100;) { s += i; i++; }

int i, s = 0; for (i = 1; ; ) { if (i <= 100) { s += i; i++; } else break; }

int i = 1, s = 0; for (; ; i++) { if (i <= 100) s += i; else break; }

int i = 1, s = 0; for (; ; ) { if (i <= 100) { s += i; i++; } else break; }
```

*IDE javlja greške jer ponovo deklariram postojeće varijable,
za testiranje izvoditi liniju po liniju

Petlje

while

Radi s bool tipom podatka

Sve dok uvjet ima vrijednost true nastavlja se izvođenje petlje

Uvjet se provjerava na ulazu u petlju

Tijelo petlje se ne mora izvesti niti jednom

ugnježđivanje, break, continue, jednako kao kod for petlje

```
bool uvjet=true;
int i = 0;
while (uvjet)
{
    Console.WriteLine("Osijek");
    uvjet = ++i < 10;
}
```

```
> Terminal – Predavanje5

Osijek
Osijek
Osijek
Osijek
Osijek
Osijek
Osijek
Osijek
Osijek
Osijek
Osijek
```

Izvor:

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

Kao i kod if izraz se češće koristi od bool varijable

```
int i = 0;
while (i++<10)
{
    Console.WriteLine("Osijek");
}
```

Beskonačna petlja

```
while (true)
{
    ...
}
```


Petlje

do

Radi s bool tipom podatka

Sve dok uvjet ima vrijednost true nastavlja se izvođenje petlje

Uvjet se provjerava na izlasku iz petlje

Tijelo petlje se mora izvesti barem jednom

ugnježđivanje, break, continue, jednako kao kod for petlje

```
int i = 0;
do
{
    Console.WriteLine("Osijek");
}
while (++i < 10);
```

```
int i = 10;
do
{
    Console.WriteLine("Osijek");
}
while (++i < 10);
```

> Terminal – Predavanje5

Osijek

Beskonačna petlja

```
do
{
}
while (true);
```

Izvor:

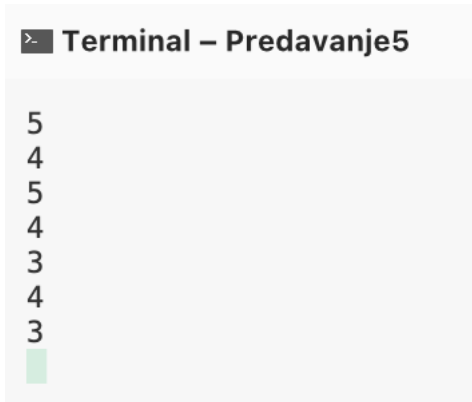
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/do>

Petlje

foreach

Ispisivanje svih elemenata niza
jedno ispod drugog „klasičnom” for petljom

```
int[] niz = { 5,4,5,4,3,4,3 };  
  
for(int i = 0; i < niz.Length; i++)  
{  
    Console.WriteLine(niz[i]);  
}
```



```
> Terminal - Predavanje5  
5  
4  
5  
4  
3  
4  
3
```

omogućuje iteraciju svakog elementa niza od indexa 0 do n
ugnježđivanje, break, continue, jednako kao kod for petlje

```
foreach(int i in niz)  
{  
    Console.WriteLine(i);  
}
```

Izvor:

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