

do петља

Одмах се извршава код у блоку, проверава се тачност услова после while команде и ако је услов тачан, поново се извршава код у блоку.

То значи да ће се сигурно извршити макар једном код у блоку без обзира на тачност услова после while наредбе.

```
do
{
    < blok koda >
} while (< uslov >);
```

Пример: Написати програм помоћу do петље који израчунава потребан број година да би се уз тренутну каматну стопу достигла потребна свота новца у банци.

```
double stanje, kamatnaStopa, ciljnoStanje;
Console.WriteLine("Koje je tvoje stanje u banci?");
stanje = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Koja je trenutno kamatna stopa u tvojoj banci (u
    %)?");
kamatnaStopa = 1 + Convert.ToDouble(Console.ReadLine()) / 100.0;
Console.WriteLine("Koje je tvoje ciljno stanje u banci?");
ciljnoStanje = Convert.ToDouble(Console.ReadLine());
int ukupnoGodina = 0;
do
{
    stanje *= kamatnaStopa;
    ++ukupnoGodina;
} while (stanje < ciljnoStanje);
Console.WriteLine($"U {ukupnoGodina} godina imaces stanje u banci od
    {stanje}.")
```

while петља

Овде се тестирање услова врши на старту сваког циклуса.

Ако је услов тачан, извршава се блок кода, а ако је нетачан, излази се из цикличне структуре на следеће линије кода у програму.

```
while (< uslov >)
{
    < blok koda >
}
```

Пример: Написати програм помоћу while петље који израчунава потребан број година да би се уз тренутну каматну стопу достигла потребна свота новца у банци.

```
double stanje, kamatnaStopa, ciljnoStanje;
Console.WriteLine("Koje je tvoje stanje u banci?");
stanje = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Koja je trenutno kamatna stopa u tvojoj banci (u
    %)?");
```

```
kamatnaStopa = 1 + Convert.ToDouble(Console.ReadLine()) / 100.0;
Console.WriteLine("Koje je tvoje ciljno stanje u banci?");
ciljnoStanje = Convert.ToDouble(Console.ReadLine());
int ukupnoGodina = 0;
while (stanje < ciljnoStanje)
{
    stanje *= kamatnaStopa;
    ++ukupnoGodina;
}
Console.WriteLine($"U {ukupnoGodina} godina imaces stanje u banci od
    {stanje}.");
```