

for петља

Циклична структура for петља се извршава одређени број пута и самостално одржава стање бројача циклуса.

Да би се дефинисала for петља, неопходно је иницијализовати бројач, поставити услов за наставак петље који укључује стање бројача, обезбедити промену стања бројача на крају сваког циклуса.

Ове вредности се постављају у првој линији кода структуре која се назива хедер for петље.

```
for (<inicijalizacijaBrojaca>; <uslov>; <promenaBrojaca>)
{
    <kod petlje>
}
```

Оваква синтакса одговара синтакси while петље:

```
<inicijalizacijaBrojaca>
while (<uslov>)
{
    <kod petlje>
    <promenaBrojaca>
}
```

Могуће је у оквиру хедера for петље и декларисати бројач али у том случају, остатак кода не препознаје бројач промењиву:

```
for (int <inicijalizacijaBrojaca>; <uslov>; <promenaBrojaca>)
```

Прекиди циклуса

Команде за реализацију прекида циклуса:

- break – безусловни излазак из цикличне структуре, прекид даљег извршавања цикличне структуре
- continue – тренутни циклус структуре се прекида са извршавањем и започиње следећи циклус (не прекида се даље извршавање цикличне структуре)
- return – скок из петље

Пример:

```
int i = 1;
while (i <= 10)
{
    if (i == 6)
        break;
    Console.WriteLine($"{i++}");
}
```

У примеру се прекида извршење цикличне структуре када бројач i достигне вредност 6.

Пример:

```
int i;  
for (i = 1; i <= 10; i++)  
{  
    if ((i % 2) == 0)  
        continue;  
    Console.WriteLine(i);  
}
```

У примеру, сваки пут када бројач добије парну вредност, извршава се команда continue, којом се прекида тренутни циклус и одмах започиње следећи циклус са промењеном вредности бројача.

Пример:

```
bool i = true;  
while(i)  
{  
    Console.WriteLine("OK");  
}
```

У примеру се користи while петља са условом који је увек испуњен (true = true увек).

То значи да смао по услову ова петља се никада неће прекинути са извршавањем – бесконачна петља.