

## Циклы

К *операторам циклов* относятся операторы `for`, `while`, `do-while`, `foreach`. Для циклов с известным числом итераций используется оператор `for`:

```
for ([<инициализатор>]; [<условие>]; [<итератор>]) <блок (оператор)>
```

Здесь *<инициализатор>* задаёт начальное значение счетчика (или счетчиков) цикла. Для счетчика может использоваться существующая переменная или объявляться новая переменная, время жизни которой будет ограничено циклом (при этом вместо типа переменной допустимо указать `var`). Цикл выполняется, пока булево *<условие>* истинно, а *<итератор>* определяет изменение счетчика цикла на каждой итерации.

В инициализаторе можно объявить и задать начальные значения для нескольких счетчиков одного типа. Если число итераций цикла заранее неизвестно, можно использовать цикл `while` или цикл `do-while`. Данные циклы имеют схожий синтаксис:

```
while (<условие>) <блок (оператор)>
```

```
do  
    <блок (оператор)>  
while (<условие>);
```

В обоих операторах цикла тело цикла выполняется, пока булево *<условие>* истинно. В цикле `while` условие проверяется в начале очередной итерации, а в цикле `do-while` – в конце. Таким образом, цикл `do-while` всегда выполнится, по крайней мере, один раз. Обратите внимание, *<условие>* должно присутствовать обязательно. Для организации бесконечных циклов на месте условия можно использовать литерал `true`:

```
while (true) Console.WriteLine("Endless loop");
```

Для перебора элементов объектов перечисляемых типов (например, массивов) в C# существует специальный цикл `foreach`:

```
foreach (<тип> <идентификатор> in <коллекция>) <блок (оператор)>
```

## Операторы перехода

К *операторам перехода* относятся `break`, `continue`, `goto`, `return`, `throw`. Оператор `break` используется для выхода из операторного блока циклов и оператора `switch`. Оператор `break` выполняет переход на оператор за блоком. Оператор `continue` располагается в теле цикла и применяется для запуска новой итерации цикла. Если циклы вложены, то запускается новая итерация того цикла, в котором непосредственно располагается `continue`.

Оператор `return` служит для завершения методов. Оператор `throw` генерирует исключительную ситуацию<sup>1</sup>.

```
switch (expression)  
{  
    case constant-expression:  
        statement  
        jump-statement  
[default:  
    statement  
    jump-statement]  
}
```

---

<sup>1</sup> Работа с методами и исключительными ситуациями рассматривается далее.