В языке C# *оператор* — это элемент программы, который применяется для одного или нескольких *операндов* в выражении или инструкции. Операторы, в которых используется один операнд, например оператор инкремента (++) или new, называются *унарными*. Операторы, в которых используются два операнда, например арифметические операторы (+,-,\*,/), называются *бинарными*. Для одного оператора — условного (?:) — используются три операнда, и такой оператор является единственным троичным оператором в C#.

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 оператора 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[[1]](#footnote-1).

switch (expression)

{

case constant-expression:

statement

jump-statement

[default:

Statement

jump-statement]

}

**Основные операторы**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| x[.](https://msdn.microsoft.com/ru-ru/library/6zhxzbds.aspx)y | Доступ к членам |
| f[(x)](https://msdn.microsoft.com/ru-ru/library/0z4503sa.aspx) | Вызов метода и делегата |
| a[[x]](https://msdn.microsoft.com/ru-ru/library/a3hd7ste.aspx) | Доступ к массиву и индексатору |
| x[++](https://msdn.microsoft.com/ru-ru/library/36x43w8w.aspx) | Постфиксный инкремент |
| x[--](https://msdn.microsoft.com/ru-ru/library/wc3z3k8c.aspx) | Постфиксный декремент |
| [new](https://msdn.microsoft.com/ru-ru/library/fa0ab757.aspx) T(...) | Создание объекта и делегата |
| **new** T(...){...} | Создание объекта с инициализатором. См. раздел [Инициализаторы объектов и коллекций (Руководство по программированию в C#)](https://msdn.microsoft.com/ru-ru/library/bb384062.aspx). |
| **new** {...} | Анонимный инициализатор объекта. См. раздел [Анонимные типы (Руководство по программированию в C#)](https://msdn.microsoft.com/ru-ru/library/bb397696.aspx). |
| **new** T[...] | Создание массива. См. раздел [Массивы (Руководство по программированию на C#)](https://msdn.microsoft.com/ru-ru/library/9b9dty7d.aspx). |
| [typeof](https://msdn.microsoft.com/ru-ru/library/58918ffs.aspx)(T) | Получение объекта System.Type для T |
| [checked](https://msdn.microsoft.com/ru-ru/library/74b4xzyw.aspx)(x) | Вычисление выражения в проверенном контексте |
| [unchecked](https://msdn.microsoft.com/ru-ru/library/a569z7k8.aspx)(x) | Вычисление выражения в непроверенном контексте |
| [default](https://msdn.microsoft.com/ru-ru/library/25tdedf5.aspx) (T) | Получение значения по умолчанию для типа T |
| [delegate](https://msdn.microsoft.com/ru-ru/library/900fyy8e.aspx) {} | Анонимная функция (анонимный метод) |

**Унарные операторы**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| [+](https://msdn.microsoft.com/ru-ru/library/k1a63xkz.aspx)x | Идентификация |
| [-](https://msdn.microsoft.com/ru-ru/library/wch5w409.aspx)x | Отрицание |
| [!](https://msdn.microsoft.com/ru-ru/library/f2kd6eb2.aspx)x | Логическое отрицание |
| [~](https://msdn.microsoft.com/ru-ru/library/d2bd4x66.aspx)x | Поразрядное отрицание |
| [++](https://msdn.microsoft.com/ru-ru/library/36x43w8w.aspx)x | Префиксный инкремент |
| [--](https://msdn.microsoft.com/ru-ru/library/wc3z3k8c.aspx)x | Префиксный декремент |
| [(T)](https://msdn.microsoft.com/ru-ru/library/0z4503sa.aspx)x | Явное преобразование x в тип T |

**Мультипликативные операторы**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| [\*](https://msdn.microsoft.com/ru-ru/library/z19tbbca.aspx) | Умножение |
| [/](https://msdn.microsoft.com/ru-ru/library/3b1ff23f.aspx) | Деление |
| [%](https://msdn.microsoft.com/ru-ru/library/0w4e0fzs.aspx) | Остаток |

**Аддитивные операторы**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| x [+](https://msdn.microsoft.com/ru-ru/library/k1a63xkz.aspx) y | Сложение, объединение строк, объединение делегатов |
| x [-](https://msdn.microsoft.com/ru-ru/library/wch5w409.aspx) y | Вычитание, удаление делегатов |

**Операторы сдвига**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| x [<<](https://msdn.microsoft.com/ru-ru/library/a1sway8w.aspx) y | Сдвиг влево |
| x [>>](https://msdn.microsoft.com/ru-ru/library/xt18et0d.aspx) y | Сдвиг вправо |

**Операторы отношения и типа**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| x [<](https://msdn.microsoft.com/ru-ru/library/z5wecxwa.aspx) y | Меньше |
| x [>](https://msdn.microsoft.com/ru-ru/library/yxk8751b.aspx) y | Больше |
| x [<=](https://msdn.microsoft.com/ru-ru/library/hx063734.aspx) y | Меньше или равно |
| x [>=](https://msdn.microsoft.com/ru-ru/library/a59bsyk4.aspx) y | Больше или равно |
| x [is](https://msdn.microsoft.com/ru-ru/library/scekt9xw.aspx) T | Возвращает значение true, если x относится к типу T, в противном случае возвращает значение false |
| x [as](https://msdn.microsoft.com/ru-ru/library/cscsdfbt.aspx) T | Возвращает x типа T или значение NULL, если x не относится к типу T |

**Операторы равенства**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| x [==](https://msdn.microsoft.com/ru-ru/library/53k8ybth.aspx) y | Равно |
| x [!=](https://msdn.microsoft.com/ru-ru/library/3tz250sf.aspx) y | Не равно |

**Логические, условные операторы и NULL-операторы**

|  |  |  |
| --- | --- | --- |
| **Категория** | **выражение** | **Описание** |
| Логическое И | x [&](https://msdn.microsoft.com/ru-ru/library/sbf85k1c.aspx) y | Поразрядное И для операндов целочисленного типа, логическое И для операндов логического типа |
| Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ | x [^](https://msdn.microsoft.com/ru-ru/library/zkacc7k1.aspx) y | Поразрядное исключающее ИЛИ для операндов целочисленного типа, логическое исключающее ИЛИ для операндов логического типа |
| Логическое ИЛИ | x [|](https://msdn.microsoft.com/ru-ru/library/kxszd0kx.aspx) y | Поразрядное ИЛИ для операндов целочисленного типа, логическое ИЛИ для операндов логического типа |
| Условное И | x [&&](https://msdn.microsoft.com/ru-ru/library/2a723cdk.aspx) y | Равно y, только если x имеет значение true |
| Условное ИЛИ | x [||](https://msdn.microsoft.com/ru-ru/library/6373h346.aspx) y | Равно y, только если x имеет значение false |
| Объединение со значением NULL | x [??](https://msdn.microsoft.com/ru-ru/library/ms173224.aspx) y | Равно y, если x имеет значение NULL, в противном случае равно x |
| Условный оператор | x [?:](https://msdn.microsoft.com/ru-ru/library/ty67wk28.aspx) y : z | Равно y, если x имеет значение true, и z, если x имеет значение false |

**Операторы присваивания и анонимные операторы**

|  |  |
| --- | --- |
| **выражение** | **Описание** |
| [=](https://msdn.microsoft.com/ru-ru/library/sbkb459w.aspx) | Присваивание |
| x op= y | Составное присваивание. Поддерживает следующие операторы: [+=](https://msdn.microsoft.com/ru-ru/library/sa7629ew.aspx), [-=](https://msdn.microsoft.com/ru-ru/library/2y9zhhx1.aspx), [\*=](https://msdn.microsoft.com/ru-ru/library/s2bkaksf.aspx), [/=](https://msdn.microsoft.com/ru-ru/library/d31sybc9.aspx), [%=](https://msdn.microsoft.com/ru-ru/library/ydwa9zh0.aspx), [&=](https://msdn.microsoft.com/ru-ru/library/e669ax02.aspx), [|=](https://msdn.microsoft.com/ru-ru/library/h5f1zzaw.aspx), [!=](https://msdn.microsoft.com/ru-ru/library/3tz250sf.aspx), [<<=](https://msdn.microsoft.com/ru-ru/library/ayt2kcfb.aspx), [>>=](https://msdn.microsoft.com/ru-ru/library/23as4533.aspx) |
| (T x) [=>](https://msdn.microsoft.com/ru-ru/library/bb311046.aspx) y | Анонимная функция (лямбда-выражение) |

1. [↑](#footnote-ref-1)