

Математические операции

В C# вы можете выполнять все стандартные арифметические действия, знакомые вам со школы. Основными операторами для работы с числовыми типами данных являются:

- **Сложение (+)**: `int a = 10 + 5; // a будет равно 15`
- **Вычитание (-)**: `int b = 10 - 5; // b будет равно 5`
- **Умножение (*)**: `int c = 10 * 5; // c будет равно 50`
- **Деление (/)**: `int d = 10 / 5; // d будет равно 2`

Важный нюанс при делении:

Если вы делите два целых числа, результат также будет целым, а дробная часть отбрасывается. Например:

```
int a = 9 / 5 //Результат будет 1, так как оба операнда целые числа
```

Чтобы получить точный результат, хотя бы один из операндов должен быть числом с плавающей точкой:

```
double a = 9.0 / 5 //Результат будет 1.8.
```

Остаток от деления (%): Этот оператор возвращает остаток от целочисленного деления.

```
'int e = 10 % 3;' // e будет равно 1 (10 делим на 3, в остатке 1)
```

Его часто используют для проверки числа на четность: если `число % 2` равно 0, то число четное.

Инкремент и декремент (++ , --): Эти унарные операторы увеличивают или уменьшают значение переменной на единицу.

```
int f = 5;  
f++; // f теперь равно 6  
int g = 5;  
g--; // g теперь равно 4
```

Они могут быть префиксными (`++f`) и постфиксными (`f++`), разница проявляется в том, какое значение возвращается в выражении.

Класс Math : Для более сложных математических вычислений, таких как возведение в степень, извлечение квадратного корня или тригонометрические функции (синус, косинус), в C# существует встроенный класс Math .

Логические операции

Логические операции используются для работы с булевым типом данных (bool), который может принимать только два значения: true (истина) или false (ложь). Они являются основой для создания условий и управления потоком выполнения программы.

Основные логические операторы:

- **Логическое "И" (&&)**: Выражение a && b истинно (true) только в том случае, если и a , и b истинны.

```
bool result = (5 > 3) && (10 > 5); // result будет true, так как оба условия верны.
```

- **Логическое "ИЛИ" (||)**: Выражение a || b истинно (true), если хотя бы одно из выражений, a или b , истинно.

```
bool result = (5 < 3) || (10 > 5); // result будет true, так как второе условие верно.
```

- **Логическое "НЕ" (!)**: Этот унарный (работает только с одним аргументом) оператор инвертирует (меняет на противоположное) логическое значение.

```
bool result = !(5 > 3); // result будет false, так как (5 > 3) – это true, а "НЕ" true – это false.
```

Операторы сравнения: Логические операторы используются вместе с операторами сравнения, которые сравнивают два значения и возвращают результат в виде bool .

- == (равно)
- != (не равно)
- < (меньше)
- > (больше)
- <= (меньше или равно)
- >= (больше или равно)

Условные и безусловные логические операторы: Стоит отметить, что существуют также операторы & и | (без удвоения). Главное отличие в том, что && и || являются "условными" или "сокращенными". Это означает, что они не будут вычислять правый

операнд, если результат уже ясен из левого. Например, в выражении `false && SecondOperand() SecondOperand()` никогда не будет вызван, так как результат всего выражения уже определенно `false`.

Приоритет операций

Как и в математике, в C# операции имеют свой приоритет. Например, умножение и деление выполняются раньше сложения и вычитания. Логические операции также имеют свой порядок выполнения. Для явного указания порядка вычислений всегда можно использовать круглые скобки `()`.