

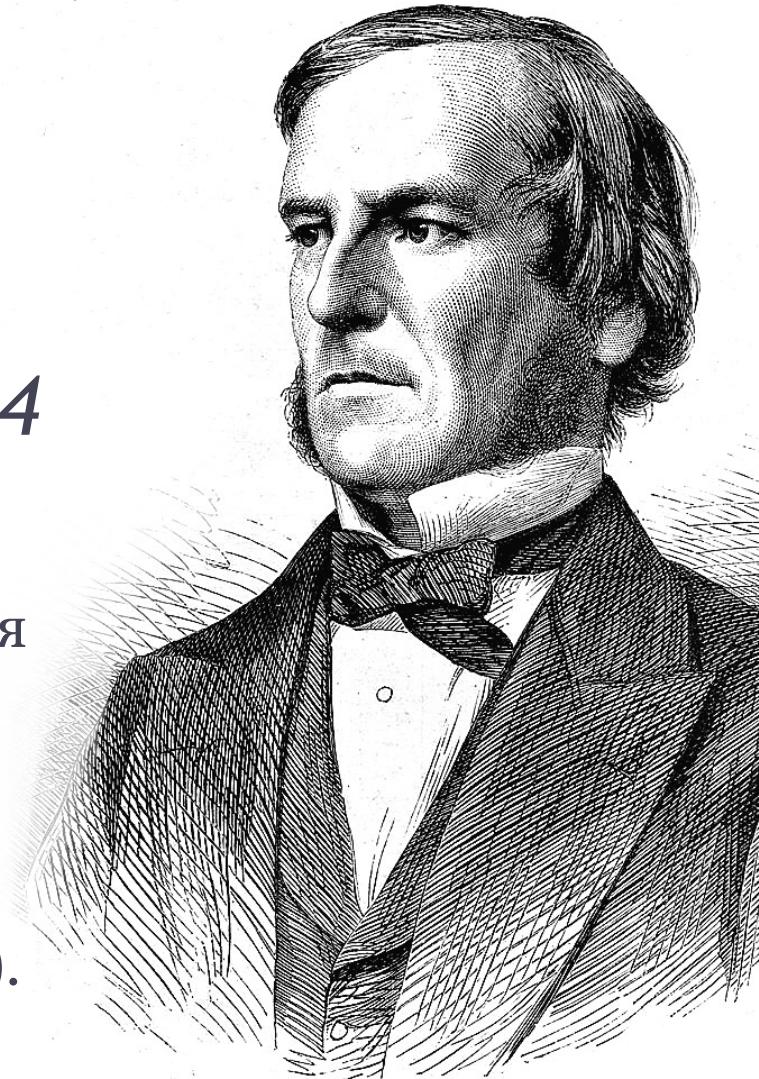
# Платформа .NET и язык программирования C#

Логический тип данных

# Джордж Буль (George Bool)

02.11.1815 – 08.12.1864

Джордж Буль по праву считается отцом математической логики. Его именем назван раздел математической логики — булева алгебра (алгебра логики).



# Логический тип данных

Переменная логического типа может принимать только два значения — "истина" и "ложь".

В языке C# для обозначения этих значений зарезервировано два слова — `true` и `false` соответственно.

Для хранения значения логического типа используется 1 бит памяти.

Для объявления переменной логического типа в C# используется зарезервированное слово `bool`.

Например,

```
bool flag = false;
```

# Операторы сравнения

Операторы сравнения позволяют сравнивать два значения любого типа, для которого они определены, например числовые типы данных, символы.

Результат получается типа `bool` (`true` или `false`).

`==` — равно

`!=` — не равно

`>` — больше

`<` — меньше

`>=` — больше или равно

`<=` — меньше или равно

Операторы `"=="` и `"!="` могут быть применимы к строкам.

# Логические операции

**Логические операции над операндами типа bool.**  
Результат логической операции тоже имеет тип bool.

| Операция                              | Знак операции | Пример       |
|---------------------------------------|---------------|--------------|
| Отрицание (НЕ, NOT)                   | !             | $! (x > 0)$  |
| Конъюнкция (И, AND)                   | &             | $x \& y$     |
| Дизъюнкция (ИЛИ, OR)                  |               | $x   y$      |
| Исключающая<br>дизъюнкция (ЛИБО, XOR) | $\wedge$      | $x \wedge y$ |

Исключающая дизъюнкция — это отрицание эквиваленции.

$$A \oplus B = \neg(A \leftrightarrow B) = (A \wedge \neg B) \vee (\neg A \wedge B)$$

# Таблицы истинности

0 – false, 1 – true

## Отрицание (NOT)

| <b>x</b> | <b>!x</b> |
|----------|-----------|
| 1        | 0         |
| 0        | 1         |

## Конъюнкция (AND)

| <b>x</b> | <b>y</b> | <b>x &amp; y</b> |
|----------|----------|------------------|
| 1        | 1        | 1                |
| 1        | 0        | 0                |
| 0        | 1        | 0                |
| 0        | 0        | 0                |

## Дизъюнкция (OR)

| <b>x</b> | <b>y</b> | <b>x   y</b> |
|----------|----------|--------------|
| 1        | 1        | 1            |
| 1        | 0        | 1            |
| 0        | 1        | 1            |
| 0        | 0        | 0            |

## Исключающая дизъюнкция (XOR)

| <b>x</b> | <b>y</b> | <b>x ^ y</b> |
|----------|----------|--------------|
| 1        | 1        | 0            |
| 1        | 0        | 1            |
| 0        | 1        | 1            |
| 0        | 0        | 0            |

# Логические операции (продолжение)

Помимо !, &, | и ^ имеются

| Операция                             | Знак операции           | Пример  |
|--------------------------------------|-------------------------|---|
| Ленивая конъюнкция<br>(уточненное И) | <code>&amp;&amp;</code> | <code>(x &gt;= 3) &amp;&amp; (x &lt;= 7)</code> |

В ленивой конъюнкции значение второго операнда не вычисляется, если значение первого операнда равно `false`.

|  |                 |                                   |
|--|-----------------|-----------------------------------|
| Ленивая дизъюнкция<br>(уточненное ИЛИ) | <code>  </code> | <code>(x == 0)    (x == 1)</code> |
|--|-----------------|-----------------------------------|

В ленивой дизъюнкции значение второго операнда не вычисляется, если значение первого операнда равно `true`.

# Логические операции над числами

**Логические операции над целыми числами** — производятся **поразрядно** в двоичном представлении числа:

| Операция                        | Знак операции | Пример       |
|---------------------------------|---------------|--------------|
| Инверсия                        | $\sim$        | $\sim x$     |
| Конъюнкция (И, AND)             | $\&$          | $x \& y$     |
| Дизъюнкция (ИЛИ, OR)            | $ $           | $x   y$      |
| Исключающая<br>дизъюнкция (XOR) | $\wedge$      | $x \wedge y$ |

**Примечание.** В C# есть еще две поразрядных операции: сдвиг влево  $x << y$  и сдвиг вправо  $x >> y$  (сдвиг разрядов числа x на y позиций). Эти операции не относятся к логическим.