

**Лекция 3.
Ветвление.
Логический тип**

Ветвление

- Часто требуется, чтобы код выполнялся только если выполняется некоторое условие
- Например, если значение делителя равно 0, то выдать ошибку. Иначе – выполнить деление
- Для этого во многих языках есть конструкция **ветвление (условный оператор)**

Высказывания, истина и ложь

- **Логическим высказыванием** называется некоторое предложение, про которое можно сказать **ИСТИННО** оно или **ЛОЖНО**
- Примеры:
 - Сейчас идет дождь
 - Завтра четверг
 - Число x больше 5
- Эти высказывания можно вычислить в конкретный момент времени и получить результат – истина или ложь

Логический тип bool

- Используется для задания условий
- Имеет два возможных значения – `true` (истина) и `false` (ложь)
- Пример:
- `bool a = true;`
`bool b = false;`

Логические выражения

Выражение	Символ	Пример
Проверка на равенство	==	a == b
Проверка на неравенство	!=	a != b
Строгое сравнение	> и <	a > b a < b
Нестрогое сравнение	>= и <=	a >= b a <= b

- Пример:

```
double a = Convert.ToDouble(Console.ReadLine());  
Console.WriteLine(a > 5); // выдаст true, если  
// введенное число > 5, иначе - false
```

Не нужно путать проверку на равенство == с оператором присваивания =

Логические связи

Название	Синонимы	Операторы	Примеры
Логическое И	Конъюнкция	&&	<code>a >= 5 && b == 3</code>
Логическое ИЛИ	Дизъюнкция		<code>3 < 5 b > 4</code>
Логическое НЕ	Отрицание	!	<code>!(a == 4)</code>

Логическое И (конъюнкция)

a	b	a && b
false	false	false
false	true	false
true	false	false
true	true	true

- **Конъюнкция** истинна (результат - **true**) только если истинны все подвыражения (все подвыражения дают **true**)
- Пример:

```
int a = 5, b = 6;  
bool c = (a < b) && (a < b - 5);    // false  
bool d = (a < b) && (b != a);        // true
```

Логическое ИЛИ (дизъюнкция)

a	b	a b
false	false	false
false	true	true
true	false	true
true	true	true

- **Дизъюнкция** истинна если истинно хотя бы одно подвыражение

Логическое НЕ (отрицание)

a	!a
false	true
true	false

- Просто обращение значения выражения

Приоритет логических операторов

- Оператор && имеет больший приоритет, чем ||
- Поэтому эквивалентно:
 $a > 5 \ || \ c > 5 \ \&\& \ d < 3$
 $a > 5 \ || \ (c > 5 \ \&\& \ d < 3)$
- Приоритет логических операторов, ниже, чем у операторов сравнения. Поэтому сначала вычисляются подвыражения, например
 $a > 5, c > 5, d < 3.$
А уже для них производится комбинирование при помощи связок

Приоритет логических операторов

- Общее правило для скобок – поставь, если не уверен как это работает
- Кроме того, кое-где скобки имеет смысл ставить для повышения читаемости:
- `a > 5 || c > 5 && d < 3` // плохо читаемо
- `(a > 5) || (c > 5 && d < 3)` // читается лучше
- `(a > 5) || ((c > 5) && (d < 3))` // чрезмерное
// количество скобок

Свойства логических связей

- **Законы де Моргана:**

$$\neg(a \ \&\& \ b) \Leftrightarrow \neg a \ || \ \neg b$$

$$\neg(a \ || \ b) \Leftrightarrow \neg a \ \&\& \ \neg b$$

- **Дистрибутивность:**

$$a \ \&\& \ (b \ || \ c) \Leftrightarrow (a \ \&\& \ b) \ || \ (a \ \&\& \ c)$$

$$a \ || \ (b \ \&\& \ c) \Leftrightarrow (a \ || \ b) \ \&\& \ (a \ || \ c)$$

Обращение операторов сравнения

- Если хотим получить отрицание операторов сравнения

Оператор	Обращение
$a == b$	$a != b$
$a > b$	$a \leq b$
$a < b$	$a \geq b$

- Строгое сравнение заменяется нестрогим

Задача

- Практика у доски – обращение сложного логического выражения (с логическими связками и арифметическими операциями)

Свойства логических связей

- **Идемпотентность:**

$$A \&\& A \Leftrightarrow A$$

$$A || A \Leftrightarrow A$$

- **Двойное отрицание:**

$$!!A \Leftrightarrow A$$

- **Аксиомы:**

$$A \&\& !A \Leftrightarrow \text{false}$$

$$A || !A \Leftrightarrow \text{true}$$

Свойство && и ||

- Данные логические связки вычисляют результат слева направо, и прекращают вычисления, если результат уже определен
- Пример:
- $3 > 5 \ \&\& \ a > b \ \&\& \ b + c > 3$ // будет false, а // вторая и третья проверки даже не выполнятся
- $3 < 5 \ || \ a > 5$ // будет true, второе условие даже не // будет проверяться
- За счет этого, эти связки некоммутативны, т.е. $A \ \&\& \ B$ не всегда $\Leftrightarrow B \ \&\& \ A$

Свойство && и ||

- Свойство, что логические связки && и || могут не вычислять все операнды, часто используется на практике
- Пусть у нас есть функция $f(x)$, которая «падает», если значение x равно 0
- Тогда такое условие будет работать правильно:
- $x \neq 0 \ \&\& \ f(x)$
- А такое упадет, если x будет равен 0:
- $f(x) \ \&\& \ x \neq 0$

Условный оператор if

- `if (a > 5)`
 {
 `Console.WriteLine("Внутри if");`
 }
- Код внутри блока `if` выполняется только если условие в круглых скобках истинно (равно `true`)
- Т.е. когда сюда дошел код, то вычисляется выражение-условие. Если результат `true`, то выполняется код в фигурных скобках

Условный оператор if

- `if` (логическое выражение) инструкция
- Здесь инструкция – это либо 1 строка кода, либо целый блок кода внутри фигурных скобок
- Пример:
- `if (a > 5) b = 3; // внутри if одна команда`
- ```
if (a > 5)
{
 b = 3;
 c = 4;
}
```
- Внутри блока делается отступ размером в 1 табуляцию

# Условный оператор if

- Следует всегда использовать блок в фигурных скобках, даже для одной команды, чтобы не ошибиться
- `if (a > 5)`
  - `b = 3; // выполняется если a > 5`
  - `c = 4; // выполняется ВСЕГДА - ошибка`

# Условный оператор if и ;

- Пример:
- `if (a > 5) b = 3; // внутри if одна команда`
- ```
if (a > 5)
{
    b = 3;
    c = 4;
}
```
- Заметим, что после фигурных скобок `if` не ставится точка с запятой

Условный оператор if и ;

- Пример:
- `if (a > 5) b = 3; // внутри if одна команда`
- Заметим, что после условия в скобках точка с запятой тоже не ставится
- Это очень важно, потому что отдельно стоящая точка с запятой считается пустой командой, которая ничего не делает
- `if (a > 5) ; b = 3;`
- В таком коде `b = 3` выполнится всегда, а внутри `if` будет пустая инструкция

Задача

- Прочитать с консоли целое число
- Если оно положительное, то напечатать в консоль строку – «Данное число - положительное»
- * Напечатайте, что число четное, если оно четное. И что кратно 5, если кратно 5

if-else

- `if` может содержать необязательную часть `else` (иначе) – какой код выполнить, если условие в `if` является ложным
- `if` (логическое выражение)
инструкция1
`else`
инструкция2

Пример if-else

- ```
if (a > 5)
{
 Console.WriteLine("a > 5");
}
else
{
 Console.WriteLine("a <= 5");
}
```

# Составление цепочек if-else

- ```
if (a > 5)
{
    // код 1
}
else if (a == 5)
{
    // код 2
}
else
{
    // код 3
}
```

Вложение if'ов

- Ветвления можно вкладывать друг в друга сколько хочется

- `if (a > 5)`
 {
 `int b = a + 5;`
 `if (b < 33)`
 {
 `Console.WriteLine("1");`
 }
 `else`
 {
 `Console.WriteLine("2");`
 }
 }

Задача «Цепочки if»

- Написать программу, которая читает с консоли целое число и печатает, что данное число положительное, если оно положительное; что равно 0, если число равно 0; что оно отрицательное, если число отрицательное
- Использовать цепочку `if-else if-else`

Тернарный оператор

- Является альтернативой для `if-else`
- Запись:
логическое выражение ? выражение1 : выражение 2;
- Выражения 1 и 2 должны возвращать значения одного и того же типа (или совместимых типов)
- Пример - определение максимума из двух чисел:
- `int max;`
`if (x > y)`
`{`
 `max = x;`
`}`
`else`
`{`
 `max = y;`
`}`
`int max = (x > y) ? x : y;`

Задача на дом «Max/min»

- Прочитать из консоли два целых числа
- Вывести наименьшее и наибольшее из них
- Сделать данную задачу при помощи `if-else` и при помощи тернарного оператора

Длина строки

- `string s = "Hello";`
`int a = s.Length; // 5`
- Length – с англ. длина

Проверка строк на равенство

- В C# для строк можно использовать оператор ==
- Или можно использовать функцию строки Equals
- Можете проверить следующий код:
- ```
string s = "Hello";
string userLine = Console.ReadLine();
if (userLine == s)
{
 // true если ввести Hello
}
```



# Проверка строк на равенство

- Проверка при помощи команды Equals:

- `string s = "Hello";`  
`string userLine = Console.ReadLine();`

```
if (s.Equals(userLine))
{
 // true если ввели Hello
}
```

# Задача на дом «Пароль»

- В программе объявить строковую переменную, хранящую пароль
- С консоли прочитать строку, сравнить её с этим паролем. Если строка совпала, то выдать сообщение, что пароль верный
- Если строка не совпала с паролем, и её длина (использовать Length) больше длины пароля, то сказать что пароль неверный и строка слишком длинная
- Если строка не совпала с паролем, и её длина меньше, то сказать, что пароль неверный строка слишком короткая
- Иначе сказать, что пароль неверный

# Задача на дом «Високосный год»

- Прочитать с консоли год и вывести в консоль, является он високосным или нет
- Старайтесь использовать логические связки, если это возможно

# Задача на курс «Площадь треугольника»

- Прочитать с консоли координаты трёх точек на плоскости:  
(x1, y1), (x2, y2), (x3, y3)
- Вычислить и вывести на экран площадь данного треугольника
- Для вычисления площади можно воспользоваться формулой Герона:
- $$S_{\Delta} = \sqrt{p(p-a)(p-b)(p-c)},$$
где  $p$  – полупериметр треугольника  $p=(a+b+c)/2$ ,  
 $a, b, c$  – длины сторон треугольника
- Проверить на случай, когда эти точки лежат на одной прямой – в этом случае вычислять площадь не нужно, а нужно вывести сообщение об этом
- Для вычисления корня использовать команду  
`Math.Sqrt(значение)`

# Задача на курс «Возраст»

- Программа просит ввести пользователя свой возраст от 1 до 112 включительно, после чего выводит сообщение «Вам x лет»
- При этом учесть, что для разных чисел разные склонения
- Например, «3 года», «99 лет» и т.д.
- Если введут слишком малое или слишком большое число, то выведите, что «Вы слишком малы» или стары
- Старайтесь использовать логические связки, если это возможно

# Задача на курс «Квадратное уравнение»

- Прочитать с консоли коэффициенты  $a$ ,  $b$  и  $c$  квадратного уравнения  $ax^2 + bx + c = 0$  и найти решение этого уравнения
- Не забыть рассмотреть все 3 случая – когда есть 2 корня, 1 корень и нет решений
- Рассмотреть в том числе случаи когда  $a$ ,  $b$ ,  $c$  равны нулю

# Задача на курс «Следующая дата»

- Программа запрашивает сегодняшнюю дату, и выдает дату следующего дня
- Например, входные данные: 31 12 2015, на выходе: 01.01.2016
- День, месяц и год можно запрашивать у пользователя с консоли по очереди
- Еще сделать проверку даты на корректность