

# Урок 2

Переменные и выражения

# В этом уроке Вы узнаете

- Базовый синтаксис
- Переменные
- Типы данных
- Константы
- Простые выражения

# Алфавит языка

- Алфавит C# включает:
  - прописные и строчные латинские буквы и знак подчеркивания;
  - арабские цифры от 0 до 9;
  - специальные знаки: " { } , [ ] ( ) + - / % \* . \ ? < = > ! & | # - ; ';
  - пробельные символы: пробел, символы табуляции, символы перехода на новую строку.
- Из символов алфавита формируются лексемы (token) языка:
  - идентификаторы;
  - ключевые (зарезервированные) слова;
  - знаки операций;
  - константы;
  - разделители (скобки, точка, запятая, пробельные символы).

# Идентификатор

- Идентификатор — это имя программного объекта
- В идентификаторе могут использоваться латинские буквы, цифры и знак подчеркивания.
- Прописные и строчные буквы различаются, например, name1, Name1 и NAME1 — три различных имени.
- Первым символом идентификатора может быть буква или знак подчеркивания, но не цифра. Пробелы внутри имен не допускаются.
- Идентификатор не должен совпадать с ключевыми словами
- Идентификатор создается на этапе объявления переменной, функции, типа и т. п., после этого его можно использовать в последующих операторах программы.

# Ключевые слова

Ключевые слова — это зарезервированные идентификаторы, которые имеют специальное значение для компилятора. Их можно использовать только в том смысле, в котором они определены.

<code>abstract</code>	<code>do</code>	<code>in</code>	<code>protected</code>	<code>true</code>
<code>as</code>	<code>double</code>	<code>int</code>	<code>public</code>	<code>try</code>
<code>base</code>	<code>else</code>	<code>interface</code>	<code>readonly</code>	<code>typeof</code>
<code>bool</code>	<code>enum</code>	<code>internal</code>	<code>ref</code>	<code>uint</code>
<code>break</code>	<code>event</code>	<code>is</code>	<code>return</code>	<code>ulong</code>
<code>byte</code>	<code>explicit</code>	<code>lock</code>	<code>sbyte</code>	<code>unchecked</code>
<code>case</code>	<code>extern</code>	<code>long</code>	<code>sealed</code>	<code>unsafe</code>
<code>catch</code>	<code>false</code>	<code>namespace</code>	<code>short</code>	<code>ushort</code>
<code>char</code>	<code>finally</code>	<code>new</code>	<code>sizeof</code>	<code>using</code>
<code>checked</code>	<code>fixed</code>	<code>null</code>	<code>stackalloc</code>	<code>virtual</code>
<code>class</code>	<code>float</code>	<code>object</code>	<code>static</code>	<code>void</code>
<code>const</code>	<code>for</code>	<code>operator</code>	<code>string</code>	<code>volatile</code>
<code>continue</code>	<code>foreach</code>	<code>out</code>	<code>struct</code>	<code>while</code>
<code>decimal</code>	<code>goto</code>	<code>override</code>	<code>switch</code>	
<code>default</code>	<code>if</code>	<code>params</code>	<code>this</code>	
<code>delegate</code>	<code>implicit</code>	<code>private</code>	<code>throw</code>	

# Структура языка

{

**Оператор1;**

**Оператор2; Оператор3;**

}

# Комментарии

```
/*Это комментарий*/  
/* Это многострочный  
   комментарий */  
//И это тоже комментарий
```

```
Console.WriteLine("Hello, World");//Вывод сообщения на  
   экран
```

```
Console.BackgroundColor/*Цвет заливки*/= ConsoleColor.Blue;
```

# Синтаксис языка

Регистрозависимый язык

```
Console.WriteLine("Hello, World");
```

```
console.WriteLine("Hello, World");
```

```
CONSOLE.WriteLine("Hello, World");
```

Среда разработки Visual Studio подсказывает, где ошибка...



# Переменные

- **Переменная — это именованная область памяти, в которой хранятся данные определенного типа.**
- У переменной есть имя и значение. Имя служит для обращения к области памяти, в которой хранится значение.
- Во время выполнения программы значение переменной можно изменять.
- Перед использованием любая переменная должна быть определена.

# Объявление переменной

**ТИП**    **ИМЯ;**

**ТИП**    **ИМЯ1, ИМЯ2, ИМЯ3;**

# Простые типы данных

C# включает две основные категории встроенных типов данных: **обычные типы** (или простые типы, типы по значению – value types) и **ссылочные типы (reference type)**. К ссылочным типам относятся классы

В C# для каждого типа данных строго определен диапазон значений и возможные операции над ними.

# Простые типы данных

Type	Signed size in Byte:		Possible Values(Range)
sbyte	Yes	1	-128 to 127
short	Yes	2	-32768 to 32767
int	Yes	4	-2147483648 to 2147483647
long	Yes	8	-9223372036854775808 to 9223372036854775807
byte	No	1	0 to 255
ushort	No	2	0 to 65535
uint	No	4	0 to 4294967295
ulong	No	8	0 to 18446744073709551615
float	Yes	4	Approximately +/- $1.5 \cdot 10^{-45}$ to +/- $3.4 \cdot 10^{38}$ with 7 significant figures
double	Yes	8	Approximately +/- $5.0 \cdot 10^{-324}$ to +/- $1.7 \cdot 10^{308}$ with 15 or 16 significant figures
decimal	Yes	12	Approximately +/- $1.0 \cdot 10^{-28}$ to +/- $7.9 \cdot 10^{28}$ with 28 or 29 significant figures
char	N/A	2	Any Unicode character(16 bit)
bool	N/A	1/2	true or false

# Инициализация переменных

При определении можно присвоить переменной начальное значение, это называется инициализацией.

# Пример

```
static void Main(string[] args)
{
    int x,n;
    n = 100;
    int y = 10;
    decimal percent = 1.25M;
    double total, sum = 0.0;
    Console.WriteLine(x);

    Console.ReadLine()
}
```

Ошибка, т.к переменная x не инициализирована

(local variable) int x

Error:

Use of unassigned local variable 'x'

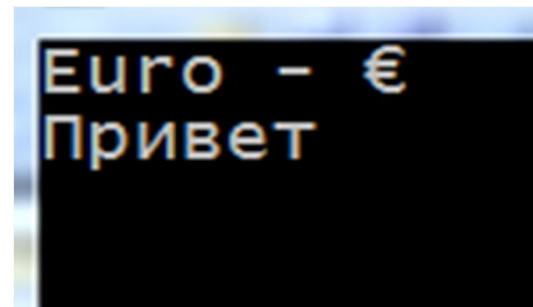
# Почему в C# используется Unicode?

- Unicode — стандарт кодирования символов, позволяющий представить знаки практически всех письменных языков (китайский, японский, греческий, грузинский, письменность майя, музыкальные и математические знаки)
- <http://unicode.org/charts/#scripts>

# Пример

```
static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.Unicode;
    char euroSign = '\u20AC';
    Console.WriteLine("Euro - " + euroSign);

    Console.WriteLine("\u041F\u0440\u0438\u0432\u0435\u0442"); //Привет
    Console.ReadLine();
}
```





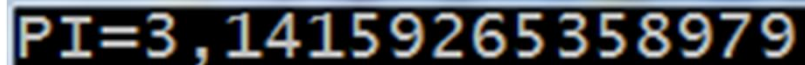
# Вопросы

1. Какие бывают типы данных?
2. Что такое переменная?
3. Необходимо обрабатывать и хранить следующие данные: 'a', 100, 12.5 – каким образом это можно организовать с помощью объявления переменных.
4. Почему тип данных `decimal` так важен для финансовых вычислений?
5. Что такое инициализация переменных?

# Константы

**Константами называют неизменяемые величины.**

```
double y;  
const double PI = 3.1415926535897932384626433832795;  
Console.WriteLine("PI=" + PI);  
y = 2 * PI * 10;  
//PI = 3.14; //Нельзя изменять значения
```

A screenshot of a console window showing the output of the code. The text "PI=3,14159265358979" is displayed in a yellow font on a black background. The comma is used as a decimal separator, which is typical for Russian locale settings.

PI=3,14159265358979

# Какие давать имена?

## Pascal Casing

При этом стиле, каждое логическое слово должно начинаться с заглавной буквы. (например: `TestCounter`) – **методы, классы**

## Camel Casing

Первый символ первого логического слова с маленькой буквы, остальные логические слова с большой (например: `testCounter`) - **переменные**

## Upper case

При этом соглашении в именовании используются только заглавные буквы. Этот стиль используется только при именовании **констант**

(например: `PI`, `MAX_VALUE`)

# Арифметические операторы

+ Сложение	<code>total = cost + tax;</code>
- Вычитание	<code>change = payment - total;</code>
* Умножение	<code>tax = cost * taxKate;</code>
/ Деление	<code>average = total / count;</code>
% Остаток от деления	<code>mod=10%2;</code>

**Оператор деления.** Если оба его аргумента имеют целочисленный тип, то этот оператор рассматривается, как оператор деления целых чисел без остатка.

Если же хотя бы один из операторов будет иметь дробный тип (вещественный), то оператор деления выполняется, как оператор деления десятичных дробей.

# Операции в выражениях

- Операции выполняются в соответствии с приоритетами.
- Для изменения порядка выполнения операций используются круглые скобки.
- Если в одном выражении записано несколько арифметических операций одинакового приоритета то выполнение происходит слева направо.

# Задание

1. Дана сторона квадрата *side*. Найти его периметр  $p = 4 \cdot \text{side}$ .
2. Даны стороны прямоугольника *a* и *b*. Найти его площадь  $\text{square} = a \cdot b$  и периметр  $\text{perimetr} = 2 \cdot (a + b)$
3. Найти длину окружности *L* и площадь круга *S* заданного радиуса *R*:  
 $L = 2 \cdot \pi \cdot R$ ,  $S = \pi \cdot R^2$ .  
В качестве значения  $\pi$  использовать константу



# Используемые ссылки

<http://simple-cs.ru/store/csharp/3/>

C#: Учебный курс. Г. Шилдт

