

Работа со строками в Си-шарп. Класс String

- **Строки в Си-шарп** - это объекты класса `String`, значением которых является текст. Для работы со строками в этом классе определено множество методов (функций) и в этом уроке мы рассмотрим некоторые из них.
-

Чтобы использовать строку, ее нужно сначала создать – присвоить какое-либо значение, иначе мы получим ошибку: "Использование локальной переменной "[имя переменной]", которой не присвоено значение". Объявим простую строку и выведем ее на экран:

- ```
static void Main(string[] args)
{
 string s = "Hello, World!";
 Console.WriteLine(s);
}
```

- ```
static void Main(string[] args)
{
    string s;
    Console.WriteLine(s); // ошибка, строка не создана
}
```

- Для **объединения** (конкатенации) строк используется оператор "+".

```
string s = "Hello," + " World!";
```

Оператор "[]" используется для доступа (только чтение) к символу строки по индексу:

```
string s = "Hello, World!";
char c = s[1]; // 'e'
```

Свойство **Length** возвращает длину строки.

- **Спецсимволы**

Символ "\" является служебным, поэтому, чтобы использовать символ обратного слэша необходимо указывать его дважды "\\".

Символ табуляции – "\t"

Символ перевода строки – "\r\n"

Двойные кавычки – "\""

Методы (функции) класса `String` для работы со строками в Си-шарп

- Как проверить, пуста ли строка?

Метод `IsNullOrEmpty()` возвращает `True`, если значение строки равно `null`, либо когда она пуста (значение равно `""`):

```
static void Main(string[] args)
{
    string s1 = null, s2 = "", s3 = "Hello";
    String.IsNullOrEmpty(s1); // True
    String.IsNullOrEmpty(s2); // True
    String.IsNullOrEmpty(s3); // False
}
```


- Метод **IsNullOrWhiteSpace()** работает как и метод `IsNullOrEmpty()`, только возвращает `True` еще и тогда, когда строка представляет собой набор символов пробела и/или табуляции ("`\t`"):

```
static void Main(string[] args)
{
    string s1 = null, s2 = "\t", s3 = " ", s4 = "Hello";
    String.IsNullOrEmpty(s1); // True
    String.IsNullOrEmpty(s2); // True
    String.IsNullOrEmpty(s3); // True
    String.IsNullOrEmpty(s4); // False
}
```

Как проверить, является ли одна строка "больше" другой?

- Для сравнения строк используется метод **Compare()**. Суть сравнения строк состоит в том, что проверяется их отношение относительно алфавита. Строка "a" "меньше" строки "b", "bb" "больше" строки "ba". Если обе строки равны - метод возвращает "0", если первая строка меньше второй – "-1", если первая больше второй – "1":

```
static void Main(string[] args)
{
    String.Compare("a", "b"); // возвращает -1
    String.Compare("a", "a"); // возвращает 0
    String.Compare("b", "a"); // возвращает 1
    String.Compare("ab", "abc"); // возвращает -1
    String.Compare("Romania", "Russia"); // возвращает -1
    String.Compare("Rwanda", "Russia"); // возвращает 1
    String.Compare("Rwanda", "Romania"); // возвращает 1
}
```

- Чтобы игнорировать регистр букв, в метод нужно передать, как третий аргумент true.

```
String.Compare("ab", "Ab"); // возвращает -1
String.Compare("ab", "Ab", true); // возвращает 0
```

Как перевести всю строку в верхний/нижний регистр?

- Для этого используются методы `ToUpper()` и `ToLower()`:

```
static void Main(string[] args)
{
    string s = "Hello, World";
    Console.WriteLine(s.ToUpper()); // ВЫВОДИТ "HELLO, WORLD"
    Console.WriteLine(s.ToLower()); // ВЫВОДИТ "hello, world"
    Console.ReadLine();
}
```


Как проверить, содержит ли строка подстроку?

- Для проверки содержания подстроки строкой используется метод **Contains()**. Данный метод принимает один аргумент – подстроку. Возвращает True, если строка содержит подстроку, в противном случае – False. Пример:

```
static void Main(string[] args)
{
    string s = "Hello, World";

    if (s.Contains("Hello"))
        Console.WriteLine("Содержит");
    Console.ReadLine();
}
```

- Данная программа выводит слово "Содержит", так как "Hello, World" содержит подстроку "Hello".

Как найти индекс первого символа подстроки, которую содержит строка?

- Метод **IndexOf()** возвращает индекс первого символа подстроки, которую содержит строка. Данный метод принимает один аргумент – подстроку. Если строка не содержит подстроки, метод возвращает "-1". Пример:

```
static void Main(string[] args)
{
    string s = "Hello, World";
    Console.WriteLine(s.IndexOf("H")); // 0
    Console.WriteLine(s.IndexOf("World")); // 7
    Console.WriteLine(s.IndexOf("Zoo")); // -1
    Console.ReadLine();
}
```


Как узнать, начинается/заканчивается ли строка указанной подстрокой?

- Для этого используются соответственно методы **StartsWith()** и **EndsWith()**, которые возвращают логическое значение. Пример:

```
static void Main(string[] args)
{
    string s = "Hello, World";
    Console.WriteLine(s.StartsWith("Hello")); // True
    Console.WriteLine(s.StartsWith("World")); // False
    Console.WriteLine(s.EndsWith("World")); // True
    Console.ReadLine();
}
```

Как вставить подстроку в строку, начиная с указанной позиции?

- Метод **Insert()** используется для вставки подстроки в строку, начиная с указанной позиции. Данный метод принимает два аргумента – позиция и подстрока. Пример:

```
static void Main(string[] args)
{
    string s = "Hello World";
    Console.WriteLine(s.Insert(5,",")); // вставляет запятую на 5 позицию
    Console.ReadLine();
}
```


Как обрезать строку, начиная с указанной позиции?

- Метод **Remove()** принимает один аргумент – позиция, начиная с которой обрезается строка:

```
static void Main(string[] args)
{
    string s = "Hello, World";
    Console.WriteLine(s.Remove(5)); // удаляем все символы, начиная с 5 позиции, на экран
    Console.ReadLine();
}
```

выведется "Hello"

- В метод **Remove()** можно передать и второй аргумент – количество обрезаемых символов. **Remove(3, 5)** – удалит из строки пять символов начиная с 3-го.

Как получить подстроку из строки, начиная с указанной позиции?

- Для этого используется метод **Substring()**. Он принимает один аргумент – позиция, с которой будет начинаться новая подстрока:

```
static void Main(string[] args)
{
    string s = "Hello, World";
    Console.WriteLine(s.Substring(7)); // получаем строку начиная с 7 позиции, выведет "World"
    Console.ReadLine();
}
```

- В метод **Substring()**, как в метод **Remove()** можно передать и второй аргумент – длина подстроки. **Substring (3, 5)** – возвратит подстроку длиной в 5 символов начиная с 3-й позиции строки.

Как заменить в строке все подстроки указанной новой подстрокой?

- Метод **Replace()** принимает два аргумента – подстрока, которую нужно заменить и новая подстрока, на которую будет заменена первая:

```
static void Main(string[] args)
{
    string s = "Hello, World, Hello";
    Console.WriteLine(s.Replace("Hello", "World")); //выведет "World, World,
World"
    Console.ReadLine();
}
```

Как преобразовать строку в массив СИМВОЛОВ?

- Метод **ToCharArray()** возвращает массив СИМВОЛОВ указанной строки:

```
static void Main(string[] args)
{
    string s = "Hello, World";
    char[] array = s.ToCharArray(); // элементы массива – 'H', 'e', 'l', 'l'...
}
```


Как разбить строку по указанному символу на массив подстрок?

- Метод **Split()** принимает один аргумент - символ, по которому будет разбита строка. Возвращает массив строк. Пример:

```
static void Main(string[] args)
{
    string s = "Arsenal,Milan,Real Madrid,Barcelona";
    string[] array = s.Split(','); // элементы массива – "Arsenal", "Milan",
    "Real Madrid", "Barcelona"
}
```

Неизменяемые строки

- Стоит знать, что объекты класса `String` представляют собой неизменяемые (`Immutable`) последовательности символов `Unicode`. Когда вы используете любой метод по изменению строки (например `Replace()`), он возвращает новую измененную копию строки, исходные же строки остаются неизменными. Так сделано потому, что операция создания новой строки гораздо менее затратна, чем операции копирования и сравнения, что повышает скорость работы программы. В Си-шарп также есть класс **`StringBuilder`**, который позволяет изменять строки. Об особенностях работы с ним мы поговорим в одном из дальнейших уроков.

- **Домашнее задание**

Задача 1. Есть некий текст. Необходимо заменить в этом тексте все слова "Nikolay" на "Oleg".

Задача 2. Дан текст – «Сегодня мы с вами рассмотрели, как работать со строками в Си-шарп. Были описаны основные операторы и методы, которые используются для работы со строками». Обрежьте этот текст так, чтобы осталась только часть «Были описаны основные операторы и методы».

Задача 3. Дана строка, которая содержит имена пользователей, разделенные запятой – "Login1,LOgin2,login3,loGin4". Необходимо разбить эту строку на массив строк (чтобы отдельно были логины), и перевести их все в нижний регистр.