

# Программирование на языке Си#

**Работа с символьными  
строками**

**Строки в Си-шарп** - это объекты класса `String`, значением которых является текст.

```
string s = "Hello, World!";  
Console.WriteLine(s);
```

**Чтобы использовать строку, ее нужно сначала создать – присвоить какое-либо значение!**

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

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

Для работы со строками в этом классе определено множество встроенных методов (функций).

- **Split()**-разбить строку по указанному символу на массив подстрок

Данный метод принимает один аргумент - символ, по которому будет разбита строка. Возвращает массив строк.

```
string s = "Arsenal,Milan,Real Madrid,Barcelona";  
string[] array = s.Split(','); // элементы массива - "Arsenal", "Milan", "Real  
Madrid", "Barcelona"
```

- **Remove()**- используется для обрезки строки, начиная с указанной позиции

Данный метод принимает один аргумент – позиция, начиная с которой обрезается строка.

```
string s = "Hello, World";  
Console.WriteLine(s.Remove(5)); // удаляем все символы, начиная с 5  
позиции, на экран 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.

Напишите программу, которая преобразует строку, содержащую имя, отчество и фамилию человека, к форме

<фамилия> <инициалы>

### Входные данные

Входная строка содержит имя, отчество и фамилию, разделённые одиночными пробелами.

### Выходные данные

Программа должна вывести в одной строке сначала фамилию, а потом (через пробел) – инициалы.

### Примеры

входные данные

Donald Ervin Knuth

выходные данные

Knuth D.E.

## Алгоритм:

1. Ввод имени, отчества и фамилии,
2. Разбить строку на 3 переменные, выбросив пробелы
3. Обрезать имя и отчество до одной буквы
4. Объединить фамилию, пробел, букву имени, точку, букву отчества и точку
5. Вывести новую строку

Ввод фамилии, имени и отчества

```
string s;  
s=Console.ReadLine();
```

Разбить строку на 3 переменные, выбросив пробелы

```
string [ ] arr=s.Split();  
    imy=Convert.ToString(arr[0]);  
    oth=Convert.ToString(arr[1]);  
    fam=Convert.ToString(arr[2]);
```

Обрезать имя и отчество до одной буквы

```
imy=imy.Remove(1);  
oth=oth.Remove(1);
```

Объединить фамилию, пробел, букву имени, точку, букву отчества и точку

```
s=fam+" "+imy+"."+oth+".";
```

Вывести новую строку

```
Console.WriteLine(s);
```

Свойство **Length** - возвращает количество элементов в массиве (во всех размерностях)

```
int[] numbers = new int[5];  
int size = numbers.Length; // size = 5
```



- **Replace()** -заменить в строке все подстроки указанной новой подстрокой

Данный метод принимает два аргумента – подстрока, которую нужно заменить и новая подстрока, на которую будет заменена первая.

```
string s = "Hello, World, Hello";  
Console.WriteLine(s.Replace("Hello", "World")); //выведет "World, World,  
World"
```

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

- **`Compare()`** - сравнение строк

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":

```
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
```

- **Substring()** -получить подстроку из строки, начиная с указанной позиции

Данный метод принимает один аргумент – позиция, с которой будет начинаться новая подстрока.

```
string s = "Hello, World";  
Console.WriteLine(s.Substring(7)); // получаем строку начиная с 7  
позиции, выведет "World"
```

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.

Напишите программу, которая заменяет в символьной строке все буквы *a* на буквы *b* (латинские, строчные).

### Входные данные

Входная строка содержит символы латинского алфавита, как строчные, так и заглавные.

### Выходные данные

Программа должна заменить во всей строке строчные буквы *a* на буквы *b* и вывести в первой строке получившуюся символьную строку, а во второй – количество выполненных замен.

### Примеры

<b>входные данные</b>
aabbccabade f
<b>выходные данные</b>
bbbbccbbbdef
4

1. Ввод строки (Read)
2. В цикле посчитать количество букв 'a'
3. Заменить 'a' на 'b' (Replase)
4. Вывести новую строку
5. Вывести количество

# Пример:

Ввод строки

Длина строки

Получить  
1 символ с  
номером  $i$

```
int K = 0;
```

```
string s; s = Console.ReadLine();
```

```
for (int i = 0; i < s.Length; i++)
```

```
{
```

```
    if (string.Compare("a", s.Substring(i, 1)) == 0)
```

```
        K = K + 1;
```

```
}
```

```
Console.WriteLine(s.Replace("a", "b"));
```

```
Console.WriteLine(K);
```

```
Console.ReadLine();
```

Сравнить  
1 символ с «a»

Заменить все  
«a» на «b»

```
string s;  
s = Console.ReadLine();
```

- **Insert()** - используется для вставки подстроки в строку, начиная с указанной ПОЗИЦИИ.

Данный метод принимает два аргумента – позиция и подстрока.

```
string s = "Hello World";  
Console.WriteLine(s.Insert(5,",")); // вставляет запятую на 5 позицию
```

- **Contains()** - проверка содержания подстроки в другой строке

Данный метод принимает один аргумент – подстроку. Возвращает True, если строка содержит подстроку, в противном случае – False.

**1 способ**

```
string s = "Hello, World";  
  
if (s.Contains("Hello") == true)  
    Console.WriteLine("Содержит");
```

**2 способ**

```
string s = "Hello, World";  
  
if (s.Contains("Hello"))  
    Console.WriteLine("Содержит");
```

- **IndexOf()** - возвращает индекс первого символа подстроки, которую содержит

Данный метод принимает один аргумент – подстроку. Если строка не содержит подстроки, метод возвращает "-1".

```
string s = "Hello, World";  
Console.WriteLine(s.IndexOf("H")); // 0  
Console.WriteLine(s.IndexOf("World")); // 7  
Console.WriteLine(s.IndexOf("Zoo")); // -1
```



- **ToCharArray()** - преобразовать строку в МАССИВ СИМВОЛОВ

```
string s = "Hello, World";  
char[] array = s.ToCharArray(); // элементы массива - 'H', 'e', 'l', 'l'...
```

Напишите программу, которая выводит первое слово переданной её символьной строки. Слово – это последовательность непробельных символов, ограниченная пробелами или концами строки.

### **Входные данные**

Входная строка содержит произвольную последовательность символов.

### **Выходные данные**

Программа должна вывести первое слово этой строки.

- 1. Ввод строки (Read)**
- 2. В цикле посчитать количество первых пробелов**
- 3. Вырезать новую строку (без первых пробелов!!) (Substring)**
- 4. Разбить на массив слов с разделителем пробел**
- 5. Вывести 0 элемент массива**

**Ввод строки**

```
string s = Console.ReadLine();  
  
int i= 0;  
while (String.Compare(" ", s.Substring(i, 1)) == 0)  
{  
    i = i + 1;  
}
```

**Количество  
первых  
пробелов**

**Сравнить  
символ с « »**

**Получить  
1 символ с  
номером i**

```
string s1 = s.Substring(i);  
string[] array = s1.Split(' ');
```

**Вырезать без  
первых пробелов**

**Разбить на массив**

```
Console.WriteLine(array[0]);  
Console.ReadKey();
```

**Вывод 0 элемента  
массива**