

C#

III

Strings & Files

Plan

1. Символы, строки
2. Создание строк
3. Неизменность строк
4. Основные функции для работы со строками
- 5. «Строительство»**

Символы и Строки

```
char a = 'b';  
char b = '\u0043';
```

Юникод^[1] (чаще всего) или Уникод^[2] ([англ. Unicode](#)) — стандарт [КОДИРОВАНИЯ СИМВОЛОВ](#), позволяющий представить знаки почти всех письменных [ЯЗЫКОВ](#)^[3].

```
string help = "Help me!";
```

Внутри себя string содержит массив символов

```
string str = "hello string";  
Console.WriteLine(str.Length); // мы можем узнать длину (12)  
char c = str[1]; // получить символ по конкретной позиции  
Console.WriteLine(c); // Выведет "e"
```

Создание строк

Строки ссылочные (они хранятся в куче)

```
string s = new string("hello"); // ERROR
```

```
string s = "hello"; // а вот это правильно
```

```
string s = "hello" + " world"; // конкатенация (слияние) на этапе компиляции  
создаст строку hello world, а затем присвоит ее переменной s
```

```
string path = "C:\\Study\\C#"; // требуется экранирование слэшей
```

```
string path = @"C:\Study\C#"; // строка распознается как есть (без управляющих символов)
```

Escape – последовательности: <https://msdn.microsoft.com/ru-ru/library/h21280bw.aspx>



Неизменность строк

```
string str = "some string";
```

```
str[3] = 'b'; // Ошибка (property or indexer string.this[int] cannot be  
              //assigned to it's read only)
```

Но мы можем присвоить переменной новое значение

```
str = "I like"; // это уже будет другая строка
```

```
str = str + " C#"; // в str будет храниться «I like C#» - это тоже новая строка
```

Основные функции для работы

```
string[] arr = str.Split('|'); // разбивает исходную строку в массив строк
```

```
string[] arr = item.Split(' ', StringSplitOptions.RemoveEmptyEntries);
```

```
string str = "Пама";  
str = str.Replace('P', 'Д'); // Дама
```

```
string str = "My string - unchangable type";  
str = str.Remove(8, 2); // My string unchangable type
```

```
str = str.ToUpper()  
str = str.ToLower()
```

```
str.ToUpper();  
str.ToLower();  
str.IndexOf();  
str.LastIndexOf();  
str.Insert();  
str.Trim(); // Удаляет пробелы  
// вокруг строки
```

```
int equal = string.Compare(str1, str2, StringComparison.OrdinalIgnoreCase);
```

StringBuilder

Нужно выполнить много операций со строками. При каждой операции создается новая строка (т.к. строки неизменяемы). В итоге создается куча объектов.

```
StringBuilder sb = new StringBuilder("hello");
    sb.Append(" world");
    sb.AppendLine("!");
    sb.AppendFormat("10 + 5 = {0}", 15); // существует еще string.Format("");
    sb.Remove(..);
    sb.Replace(..);
string str = sb.ToString();
```

Files

- Open file
- Action
- Close file

System.IO.StreamReader

System.IO.StreamWriter

Files - Writing

```
using System.IO;
```

```
StreamWriter sw;  
sw = new StreamWriter("my-file.txt");  
sw.Write("H");  
sw.WriteLine("i");  
sw.WriteLine("Man");  
sw.Close();
```

```
using(StreamWriter sw = new StreamWriter("my-file.txt"))  
{  
    sw.Write("H");  
    sw.WriteLine("i");  
    sw.WriteLine("Man");  
}
```

`using()` вызовет `sw.Close()` автоматически

Files - Reading

```
using System.IO;
```

```
using (StreamReader sr = new StreamReader("my-file.txt"))  
{  
    while (!sr.EndOfStream)  
    {  
        string s = sr.ReadLine();  
        Console.WriteLine(s);  
    }  
}
```

`using()` вызовет `sr.Close()` автоматически



Спасибо за внимание!