

# С# Строки Дата Время

Мухортова Н.Н.



# Цель

Цель - познакомиться с методами обработки строк

Научиться работать с датой и временем.



# Создание строк

```
string s1 = "hello";
```

```
string s2 = null;
```

```
string s3 = new String('a', 6); // результатом будет строка "aaaaaa"
```

```
string s4 = new String(new char[]{'w', 'o', 'r', 'l', 'd'});
```



# Строка как набор символов

`public char this[int index] {get;}` индексатор строки, определен в классе `System.String`

```
string s1 = "hello";
```

```
char ch1 = s1[1]; // символ 'e'
```

```
Console.WriteLine(ch1);
```

```
Console.WriteLine(s1.Length);
```



# Метод Concat

Конкатенация строк или объединение может производиться как с помощью операции +, так и с помощью метода Concat:

```
string s1 = "hello";
```

```
string s2 = "world";
```

```
string s3 = s1 + " " + s2; // результат: строка "hello world"
```

```
string s4 = String.Concat(s3, "!!!"); // результат: строка "hello world!!!"
```



# Метод Join

```
string s6 = "a day";
```

```
string s7 = "keeps";
```

```
string s8 = "a doctor";
```

```
string s9 = "away";
```

```
string[] values = new string[] { s5, s6, s7, s8, s9 };
```

**Join:** соединяет элементы массива строк

```
String s10 = String.Join(" ", values);
```

```
// результат: строка "apple a day keeps a doctor away"
```



# Метод Compare

```
string s1 = "hello";
```

```
string s2 = "world";
```

```
int result = String.Compare(s1, s2);
```



# Метод IndexOf

Определить индекс первого вхождения отдельного символа или подстроки в строке

```
string s1 = "hello world";
```

```
char ch = 'o';
```

```
int indexOfChar = s1.IndexOf(ch); // равно 4
```

```
string subString = "wor";
```

```
int indexOfSubstring = s1.IndexOf(subString); // равно 6
```





# Методы StartsWith и EndsWith

```
string path = "C:\SomeDir";

string[] files = Directory.GetFiles(path);

for (int i = 0; i < files.Length; i++)
{
    if(files[i].EndsWith(".exe"))
        File.Delete(files[i]);
}
```



# Метод Split

Разделить строку на массив подстрок

```
string text = "И поэтому все так произошло";
```

```
string[] words = text.Split(new char[] { ' ' });
```

```
foreach (string s in words){
```

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

```
}
```

удаляет повторные пробелы

```
string[] words = text.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);
```



# Метод Trim

Для обрезки начальных или конечных символов

```
string text = " hello world ";
```

```
text = text.Trim(); // результат "hello world"
```

```
text = text.Trim(new char[] { 'd', 'h' }); // результат "ello worl"
```



# Метод Substring

Обрезать определенную часть строки

```
text = text.Substring(2);
```

```
// результат "роший день"
```

// обрезаем сначала до последних двух символов

```
text = text.Substring(0, text.Length - 2);
```

```
// результат "роший де"
```



# Метод Insert

Для вставки одной строки в другую

```
string text = "Хороший день";
```

```
string subString = "замечательный ";
```

```
text = text.Insert(8, subString);
```

```
// "Хороший замечательный день"
```



# Метод Remove

Удалить часть строки

```
string text = "Хороший день";
```

```
int ind = text.Length - 1;
```

```
// вырезаем последний символ
```

```
text = text.Remove(ind);
```

```
// "Хороший ден"
```

```
// вырезаем первые два символа
```

```
text = text.Remove(0, 2);
```

```
// "роший ден"
```



# Метод Replace

Заменить один символ или подстроку на другую

```
string text = "хороший день";
```

```
text = text.Replace("хороший", "плохой");
```

```
// "плохой день"
```

```
text = text.Replace("о", "");
```

```
// "плхй день"
```



# Методы ToUpper() и ToLower()

Для приведения строки к верхнему и нижнему регистру

```
string hello = "Hello world!";
```

```
Console.WriteLine(hello.ToLower()); // hello world!
```

```
Console.WriteLine(hello.ToUpper()); // HELLO WORLD!
```





## Другие методы

1. `Contains`: определяет, содержится ли подстрока в строке
2. `CopyTo`: копирует часть строки или всю строку в другую строку
3. `LastIndexOf`: находит индекс последнего вхождения символа или подстроки в строке



# Форматирование строк

```
Person person = new Person { Name = "Tom", Age = 23 };
```

```
Console.WriteLine("Имя: {0} Возраст: {1}", person.Name, person.Age);
```

То же самое с помощью метода `String.Format`

```
string output = String.Format("Имя: {0} Возраст: {1}", person.Name, person.Age);
```

```
Console.WriteLine(output);
```



# Интерполяция строк

Знак доллара перед строкой указывает, что будет осуществляться интерполяция строк

```
Person person = new Person { Name = "Tom", Age = 23 };
```

```
Console.WriteLine($"Имя: {person.Name} Возраст: {person.Age}");
```

```
int x = 8;
```

```
int y = 7;
```

```
string result = $"{x} + {y} = {x + y}";
```

```
Console.WriteLine(result); // 8 + 7 = 15
```



# Интерполяция строк

Если `person` не равен `null`, то вывод имени, иначе вывод имени по умолчанию

```
string output = $"{person?.Name ?? "Имя по умолчанию"}";
```

Номер телефона в формате `+x xxx-xxx-xx-xx`

```
long number = 19876543210;
```

```
Console.WriteLine($"{number:+# ### ### ## ##}"); // +1 987 654 32 10
```



# Работа с датами и временем

## Структура DateTime

Для работы с датами и временем в .NET предназначена структура DateTime.

Представляет время, обычно выраженное как дата и время суток.

Она представляет дату и время от 00:00:00 1 января 0001 года до 23:59:59 31 декабря 9999 года



# Инициализация

```
DateTime date1 = new DateTime(2008, 5, 1, 8, 30, 52);
```

```
DateTime dat1 = new DateTime();
```

```
// The following method call displays 1/1/0001 12:00:00 AM.
```

```
Console.WriteLine(dat1.Equals(DateTime.MinValue));
```

```
DateTime date1 = DateTime.Now; //берет текущую дату и время компьютера
```

```
DateTime date2 = DateTime.UtcNow; // дата и время относительно времени по Гринвичу (GMT)
```

```
DateTime date3 = DateTime.Today; // только текущая дата
```



# Операции с DateTime

`Add(DateTime date)`: добавляет дату date

`AddDays(double value)`: добавляет к текущей дате несколько дней

`AddHours(double value)`: добавляет к текущей дате несколько часов

`AddMinutes(double value)`: добавляет к текущей дате несколько минут

`AddMonths(int value)`: добавляет к текущей дате несколько месяцев

и другие

```
Console.WriteLine(date1.AddHours(3));
```

```
Console.WriteLine(date1.Subtract(date2));
```



# Примеры методов форматирования

```
DateTime date1 = new DateTime(2015, 7, 20, 18, 30, 25);
```

```
Console.WriteLine(date1.ToLocalTime()); // 20.07.2015 21:30:25
```

```
Console.WriteLine(date1.ToUniversalTime()); // 20.07.2015 15:30:25
```

```
Console.WriteLine(date1.ToLongDateString()); // 20 июля 2015 г.
```

```
Console.WriteLine(date1.ToShortDateString()); // 20.07.2015
```

```
Console.WriteLine(date1.ToLongTimeString()); // 18:30:25
```

```
Console.WriteLine(date1.ToShortTimeString()); // 18:30
```





# Значения даты и времени и календарей

класс `ChineseLunisolarCalendar`;

класс `EastAsianLunisolarCalendar`;


класс `GregorianCalendar`;

класс `HebrewCalendar`;

класс `HijriCalendar`;

класс `JapaneseCalendar`;

класс `JapaneseLunisolarCalendar`; и другие ...



# Прямое форматирование

## Пример строковых форматов

- D Полный формат даты. Например, 17 июля 2015 г.
  - d Краткий формат даты. Например, 17.07.2015
  - F Полный формат даты и времени. Например, 17 июля 2015 г. 17:04:43
  - f Полный формат даты и краткий формат времени. Например, 17 июля 2015 г. 17:04
  - G Краткий формат даты и полный формат времени. Например, 17.07.2015 17:04:43
  - g Краткий формат даты и времени. Например, 17.07.2015 17:04
  - M, m Формат дней месяца. Например, 17 июля
- и другие...



# Пример использования

```
DateTime now = DateTime.Now;
```

```
Console.WriteLine("D: " + now.ToString("D"));
```

```
Console.WriteLine("d: " + now.ToString("d"));
```

```
Console.WriteLine("F: " + now.ToString("F"));
```

```
Console.WriteLine("f: {0:f}", now);
```

```
Console.WriteLine("G: {0:G}", now);
```

```
Console.WriteLine("g: {0:g}", now);
```

```
Console.WriteLine("M: {0:M}", now);
```

```
Console.WriteLine("O: {0:O}", now);
```



# Выводы

В C# реализованы методы работы со строками

Метод `Concat()`. Метод `Join()`. Метод `Compare()`. Метод `IndexOf()`. Метод `Trim()`. Метод `Replace()`.

C# поддерживает разные календари

В C# можно задать форматы вывода даты и времени.