

Полезные классы .Net



Структура Complex

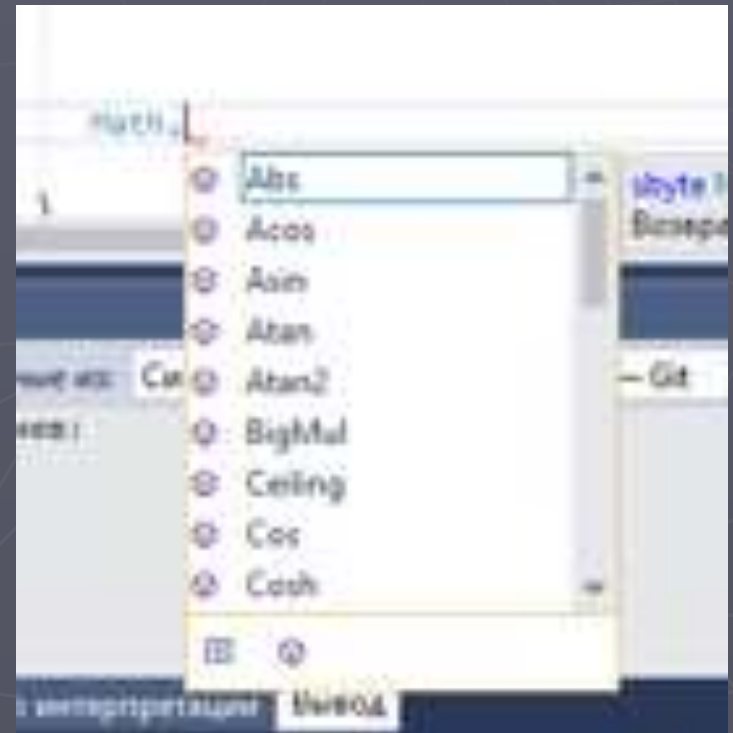
► представления комплексного числа

```
Complex z1 = new Complex(3, 5);  
Complex z2 = new Complex(-2, 10);
```

System.Math

- набор методов, соответствующих
ОСНОВНЫМ математическим функциям

Abs()	IEEERemainder()
Acos(), Asin(), Atan()	Log()
Atan2()	Log10()
BigMul()	Max(), Min()
Ceiling()	PI
Cos(), Sin(), Tan()	Pow()
Cosh(), Sinh(), Tanh()	Round()
DivRem()	Sign()
E	Sqrt()
Exp()	Truncate()
Floor()	



System.Random

- ▶ генерирует псевдослучайную последовательность значений byte, int или double.

```
Random rand = new Random(5);  
int intRand = rand.Next() +  
               rand.Next(10) +  
               rand.Next(-10, 10);  
double doubRand = rand.NextDouble();  
byte[] buffer = new byte[10];  
rand.NextBytes(buffer);
```

вещественное
число из
интервала [0, 1)

заполняет массив байтов
случайными значениям

Структура TimeSpan

- хранит интервал времени в виде отсчётов по 100 наносекунд

```
var t1 = new TimeSpan(50);  
// количество отсчётов по 100 нс  
var t2 = new TimeSpan(1, 20, 50);  
// часы, минуты, секунды
```

позволяют обратиться либо к отдельному временному компоненту интервала (свойства Days, Hours, Minutes, Seconds, Milliseconds), либо выразить весь интервал через указанную единицу времени (TotalDays, TotalHours и т.п.).


```
TimeSpan ts = TimeSpan.FromDays(5) - TimeSpan.FromSeconds(1);  
    Console.WriteLine(ts.Days);           // 4  
    Console.WriteLine(ts.Seconds);        // 59  
    Console.WriteLine(ts.TotalDays);      // 9.99998842592593
```

DateTime


- ▶ предназначена для хранения даты и времени (точность времени – 100 наносекунд)

год, месяц, день, часы, минуты, секунды, миллисекунды и тип времени (Unspecified, Local, Utc).

```
var dt1 = new DateTime(2020, 12, 11);  
var dt2 = new DateTime(2020, 12, 11, 12, 0, 0,  
                      DateTimeKind.Local);
```



```
var dt3 = new DateTime(2020, 12, 11,  
                      new PersianCalendar());
```



Определяет календарь –влияет на алгоритм вычисления некоторых свойств даты

- ▶ статические свойства Now, Today, UtcNow
- ▶ методы для увеличения
- ▶ Сравнения и аддитивные операции

```
var dt = DateTime.Today;  
    // текущая дата (без времени)  
Console.WriteLine(dt.Month);  
    // текущий месяц  
Console.WriteLine(dt.DayOfWeek);  
    // и день недели  
var modDt = dt.AddDays(-3.8);
```

```
Console.WriteLine(modDt);
```

```
11  
Tuesday  
23.11.2018 4:48:00  
Для продолжения нажмите
```


DateTimeOffset

- ▶ хранит смещение всемирного координированного времени (UTC offset),
 - возможность правильно сравнивать даты разных часовых поясов

```
var dtf = new DateTime(2019, 12, 11);  
var dto = new DateTimeOffset(  
    dtf, TimeSpan.FromHours(-6));
```

```
DateTimeOffset dto2 = DateTime.Now;  
    // неявное преобразование
```

Преобразование информации

- ▶ Статический класс `System.Convert` - взаимные преобразования данных базовых типов (числовые типы, `bool`, `string` и `DateTime`)
- ▶ Возможные исключения - `InvalidCastException`, `FormatException`, `OverflowException`

```
byte x = Convert.ToByte("100");  
        // x = 100  
bool y = Convert.ToBoolean(25);  
        // y = true  
int z = Convert.ToInt32(DateTime.Now);  
        // InvalidCastException
```

методы вида
`ТоИмяТипа()`, где `ИмяТипа`
является именем CLR для
базовых типов

- ▶ Все базовые типы явным образом реализуют интерфейс `System.IConvertible`
→
- ▶ набор методов вида `ToИмяТипа()`, где `ИмяТипа` – имя CLR для базовых типов

Статический класс System.BitConverter

- ▶ содержит набор методов для преобразования переменной числового или булевого типа в массив байт, а также методы обратного преобразования

```
byte[] data = BitConverter.GetBytes(1099);
```

```
int x = BitConverter.ToInt32(data, 0);
```

Класс CultureInfo

► поставщик формата

```
// создаём поставщик формата для русской культуры  
// список кодов: msdn.microsoft.com/en-us/globalization/bb896001  
var culture = new CultureInfo("ru-Ru");
```

идентификатор культуры

```
var s1 = dt.ToString("D", culture);
```

► Некоторые типы дополнительно перегружают Parse() и TryParse()

В случае неудачи генерирует исключение

В случае неудачи возвращает значение false

```
int x4 = Int32.Parse("1024");
```

```
DateTime dtP =  
DateTime.Parse("13/01/2020 16:45:06");
```

Базовый класс System.Text.Encoding

- ▶ поддерживает различные текстовые кодировки и наборы символов

```
Encoding utf8 = Encoding.GetEncoding("utf-8");  
Encoding ascii = Encoding.ASCII;
```

```
byte[] asciiBytes = ascii.GetBytes("TEXT");  
string s = ascii.GetString(asciiBytes);
```

для перевода строки или массива символов в массив байтов (коды символов) и обратно