

NET.C#.02

Атрибуты. Описание и  
использование

## Атрибуты

**Атрибуты** позволяют добавить к метаданным дополнительную информацию, которая затем может извлекаться при помощи механизма **рефлексии**.

Компилятор создает атрибуты, когда вы объявляете экземпляры специальных классов, наследующих от **System.Attribute**

Атрибуты используются для:

1. Описания правил сериализации данных
2. Управления безопасностью на уровне сборки
3. Облегчения отладки кода
4. Управления поведением компонент
5. Управления видимостью элементов управления и классов при разработке форм пользовательского интерфейса
6. Комментирования кода
7. ...

## Применение атрибутов

Большинство атрибутов применяется к классам, методам, полям и свойствам

**Глобальные** атрибуты – воздействуют на всю сборку или модуль

## Применение атрибутов на уровне классов и методов.

1. Определяется новый или используется существующий в .net framework атрибут
2. Инициализируется конкретный экземпляр атрибута с помощью вызова конструктора атрибута
3. Атрибут помещается в метаданные при компиляции и становится доступен CLR
4. По соглашению имена всех атрибутов оканчиваются словом `Attribute`.  
**`System.ObsoleteAttribute`**

```
using System;
public class MainApp
{
    public static void Main()
    { //На этой строке компилятор выдаст предупреждение.
        int MyInt = Add(2,2);
    }

    //Этот атрибут применяется только к методу Add.
    [Obsolete("В следующей версии метод будет удален")]
    public static int Add(int a, int b)
    { return (a + b); }
}
```

**Атрибут `System.ObsoleteAttribute`, помечающий код как устаревший, заставляет компилятор выдать переданную строку как предупреждение при компиляции помеченного кода.**

На уровне сборок

```
using System.Reflection;  
  
[assembly:AssemblyName("Моя сборка")]
```

При компиляции кода строка "Моя сборка" помещается в манифест сборки в секции метаданных. Этот атрибут можно увидеть с помощью дизассемблера MSIL (Ildasm.exe) или с помощью пользовательских средств.

Для применения атрибута на уровне модуля используется ключевое слово `Module`.

## Пользовательские атрибуты

Пользовательские атрибуты – классы, наследующие от **System.Attribute**

Объявление начинается с **AttributeUsageAttribute**

```
[AttributeUsage(AttributeTargets.All, Inherited = false,  
AllowMultiple = true)]
```

Атрибут не будет наследоваться классами наследниками

Атрибут может применяться ко всем элементам программы

Атрибут может применяться к одному элементу многократно

## Пользовательские атрибуты

```
// По умолчанию Inherited = true.  
public class MyAttribute : Attribute { }  
  
// Явно задается false.  
[AttributeUsage(Inherited = false)]  
public class YourAttribute : Attribute { }
```

```
public class MyClass  
{  
    [MyAttribute][YourAttribute]  
    public void MyMethod() { //... }  
}
```

## Пользовательские атрибуты

```
public class MyClass
{ // Ошибка - дублирование не разрешено.
  [MyAttribute, MyAttribute]
  public void MyMethod() { //... }

// Это допустимо.
  [YourAttribute, YourAttribute]
  public void YourMethod() { //... }
}
```



## Типы данных, допустимые в атрибутах

- Bool
- Byte
- Char
- Double
- Float
- Int
- Long
- Short
- String
- Object
- System.Type
- Открытые перечислимые типы, вложенные (если вложены) в открытые типы

Thanks for your  
attention