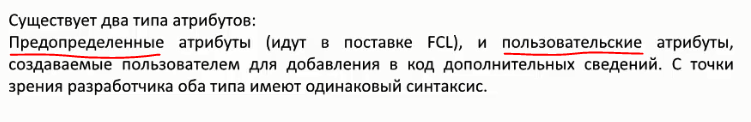
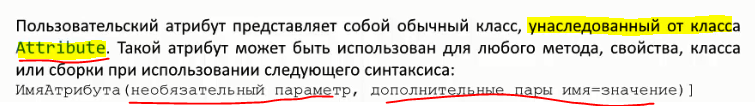
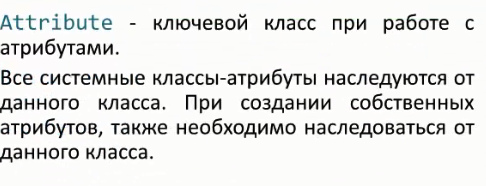
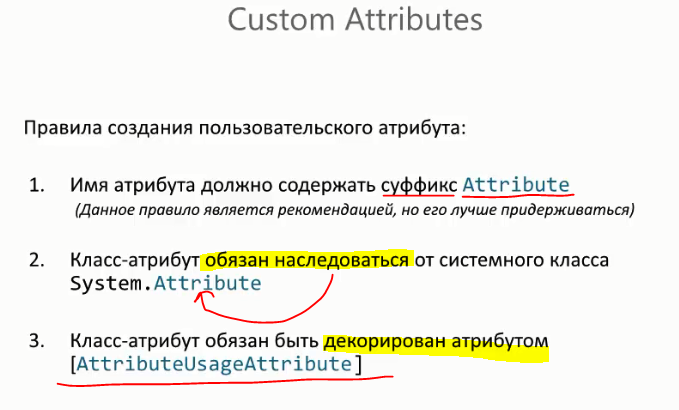
Дополнительная информация о классе, свойстве или методе









Атрибуты могут использоваться

перед объявлениями типов,

членов класса,

членов интерфейса

и перечислений

Виды атрибутов:

Предопределенные и Пользовательские

[ ....... ] - секция атрибута.

Объявление атрибута состоит

из Имени атрибута и

набора Параметров

Для атрибутов определяют два вида параметров –

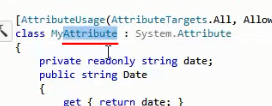
Позиционные и Именованные

Именованные параметры всегда располагаются после Позиционных

Создание собственного атрибута

1

Суфикс имени атрибута должен быть Attribute



2

Для создания атрибута необходимо создать класс производный от класса : System.Attribute.



3

Должен быть декорирован [AttributeUsage]

Атрибут - [AttributeUsage] - задает свойства пользовательских атрибутов.



Позиционный параметр - AttributeTargets, определяет элементы программы, для которых атрибут может быть применен. (Параметр - AttributeTargets.All - позволяет использовать атрибут - MyAttribute, для любого элемента.)

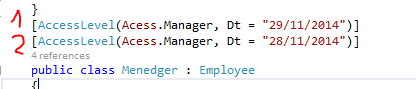
Именованный параметр - AllowMultiple = false, определяет сколько раз к одному элементу можно применять атрибут.

object[] attributes = type.GetCustomAttributes(typeof(MyAttribute), true);

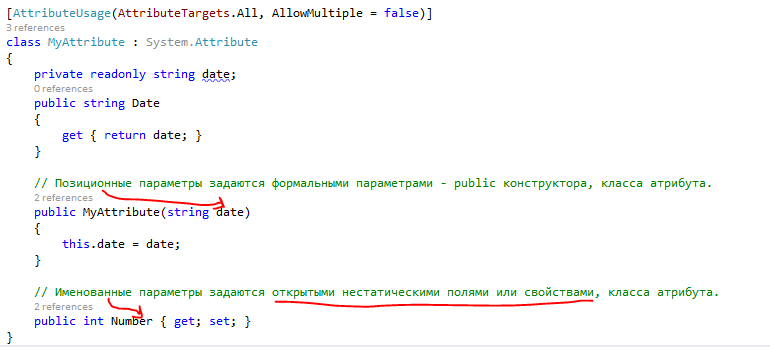
attributes.GetLength(0) – вернет количество элементов в первой размерности

AllowMultiple

[AttributeUsage(AttributeTargets.Class,AllowMultiple = true)]

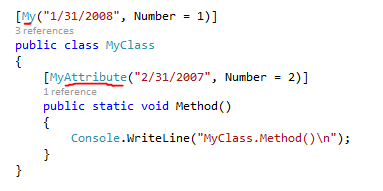


Разрешает применить множество пользовательский атрибутов одного типа к одному и то му же члену



Применение атрибута к классу и статическому методу

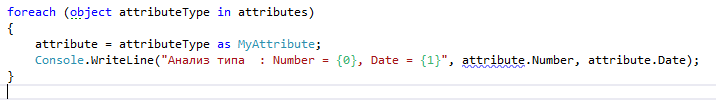
Передав в атрибут позиционный и именованный параметры



Анализ атрибутов ТИПА

Type type = typeof(MyClass); - получаем тип для проверки его атрибутов

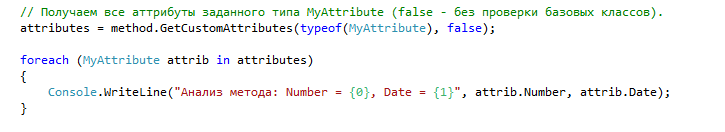
attributes = type.GetCustomAttributes(false); - Получаем все атрибуты заданного типа MyAttribute (false - без проверки базовых классов).



Анализ атрибутов МЕТОДА

MethodInfo method = type.GetMethod("Method", BindingFlags.Public | BindingFlags.Static); - Получаем public static метод - Method.

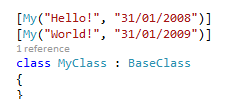
attributes = method.GetCustomAttributes(typeof(MyAttribute), false);- Получаем все аттрибуты заданного типа MyAttribute (false - без проверки базовых классов).

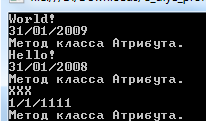




Вывод записей атрибутов в обратном порядке конструктору, от производных классов к базовому







Применение атрибута к сборке

Глобальные атрибуты для всей сборки.

[assembly: AssemblyVersionAttribute("1.0.2000.0")] // Версия сборки.

[assembly: AssemblyTitle("AssemblySmpl")] // Имя сборки.

[assembly: AssemblyDescription("")] // Описание сборки.

[assembly: AssemblyConfiguration("")] // Конфигурация, например, Release или Debug.

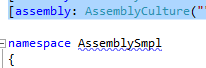
[assembly: AssemblyCompany("CyberBionic Systematics")] // Имя компании разработчика.

[assembly: AssemblyProduct("AssemblySmpl")] // Имя продукта.

[assembly: AssemblyCopyright("Copyright 2009")] // Копирайты.

[assembly: AssemblyTrademark("")] // Торговая марка.

[assembly: AssemblyCulture("")] // Какие культуры поддерживает сборка.

Применяются до пространства имен 

Получение сборки (Assembly assembly) код которой выполняется в данный момент.

Assembly assembly = Assembly.GetExecutingAssembly();

// Полное имя сборки.

Console.WriteLine("Assembly Full Name:\n{0}", assembly.FullName);

1 инвариантна ко всем культурам 2 не обладает цифровой подписью



object[] ac=assembly.GetCustomAttributes(typeof(AssemblyCompanyAttribute), false);

AssemblyCompanyAttribute[] arr = (AssemblyCompanyAttribute[])ac;

if (ac.GetLength(0)!=0)

{

foreach (var item in arr)

{

Console.WriteLine(item.Company+" "+item.TypeId);

}

}

assembly.CodeBase; - Место хранения сборки.

[assembly: AssemblyVersionAttribute("1.4.2000.0")] // Версия сборки.

Чтобы ее получить

AssemblyName объект, который позволяет разбить полное имя сборки на части.

AssemblyName assemblyName = assembly.GetName();

Console.WriteLine("\nName: {0}", assemblyName.Name); - Имя сборки

Console.WriteLine("Version: {0}.{1}", assemblyName.Version.Major, assemblyName.Version.Minor); - Версия сборки.



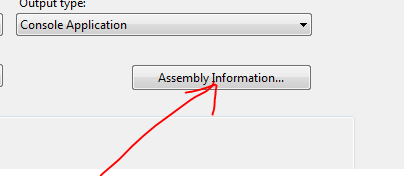


Получить первый элемент массива приведенного к нужному типу

var appVersion = attributes.OfType<AssemblyFileVersionAttribute>().Single();

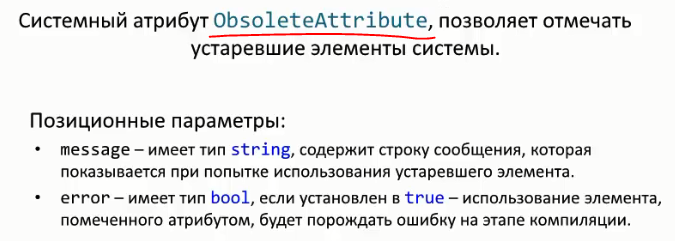
Console.WriteLine("Приложение версия {0}", appVersion.Version);

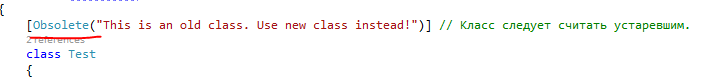
атрибуты



Системные атрибуты

Obsolete



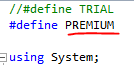


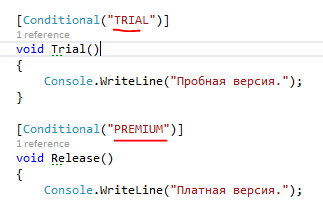
Conditional

* На условные методы накладываются следующие ограничения:
  + o Они должны возвращать значение типа void.
  + o Должны быть членами класса или структуры, но не интерфейса.
  + o Они не могут предшествовать ключевому слову override.

Метод вызовется в том случае если определена препроцессорная директива #define с указанным именем который передан в атрибут

(рефлексия не обнаружит метод который декорирован атрибутом и имя не определено в дерективе (void Trial() не виден))







[CallerMemberName] – имя метода из которого вызван данный метод

[CallerFilePath] – полный путь файла который вызвал метод 

[CallerLineNumber] – номер строки на которой находится метод во время вызова

