C#允许开发人员以特性的形式为程序添加声明性信息。

特性定义了与类、结构、方法等相关的附加信息（元数据）。可以定义一个特性来确定类中将要显示的按钮的类型。

所有的特性类必须是Attribute的子类。

特性中的命名形参：

特性的命名形参与方法中的命名实参类似，但在细节上有所不同。

[attrib(positional-param-list, named-param1 = value, named-param2 = value)]

方法的命名实参：

Func(positional-param-list, named-param1 : value, named-param2 : value);

声明特性类时，前面应添加AttributeUsage特性，指明应该改自定义特性的项的类型。

例：程序attribute\_test1

// Copyright 2016.刘珅珅

// author：刘珅珅

// 特性：命名形参

using *System*;

using *System*.*Collections*.*Generic*;

using *System*.*Linq*;

using *System*.*Reflection*;

using *System*.*Text*;

using *System*.*Threading*.*Tasks*;

namespace attribute\_test1

{

// 特性类命名通常以Attribute结束

[AttributeUsage(*AttributeTargets*.*All*)]

public class RemarkAttribute : *Attribute*

{

string remark;

public string supplement;

public RemarkAttribute(string comment)

{

remark = comment;

supplement = "None";

Priority = 1;

}

public string Remark

{

get

{

return remark;

}

}

public int Priority { get; set; }

}

// 特性的命名形参

// 只有公有的、可读写的字段和属性才可以用作命名形参

[RemarkAttribute("This class uses an attribute.",

supplement = "This is additional info.",

Priority = 10)]

class UseAttrib

{

}

class AttributeTest

{

static void Main(string[] args)

{

*Type* t = typeof(UseAttrib);

// 获取类型的特性，false表示不获取继承的特性

object[] attribs = t.*GetCustomAttributes*(false);

foreach (object obj in attribs)

{

*Console*.*WriteLine*(obj);

}

*Type* remark\_att = typeof(RemarkAttribute);

RemarkAttribute ra = (RemarkAttribute)*Attribute*.*GetCustomAttribute*(t, remark\_att);

*Console*.*Write*("Remark: ");

*Console*.*WriteLine*(ra.Remark);

*Console*.*Write*("supplement: ");

*Console*.*WriteLine*(ra.supplement);

*Console*.*WriteLine*("Priority: " + ra.Priority);

}

}

}

输出结果为：

attribute\_test1.RemarkAttribute

Remark: This class uses an attribute.

supplement: This is additional info.

Priority: 10