事件与委托的区别

1.事件与委托的关系

　　委托是一种类型  
　　事件是委托的一个实例

事件中涉及两种角色——事件发布者和事件订阅者。

　　事件发布者： 触发事件的对象称为事件发布者。

     事件订阅者： 捕获事件并对其做出处理的对象称为事件订阅者。

2.事件的实例说明

[复制代码](javascript:void(0);)

namespace MyEvent

{

//委托的声明

public delegate void DoSometing();

/// <summary>

/// 委托与事件的关系

/// 委托是一种类型

/// 事件是委托的一个实例

/// </summary>

public class MyEvent

{

public DoSometing doSomethingDelegate;//委托的一个变量

public event DoSometing doSomethingEvent;//一个事件，就是委托的一个变量，加上了event关键字

//事件的使用

public void Invoke()

{

if (doSomethingEvent != null)

{

doSomethingEvent();

}

}

}

}

namespace MyEvent

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("欢迎来到流星小子博客学习");

MyEvent myEvent = new MyEvent();

// myEvent.doSomethingEvent += new DoSometing(ShowSomething);

myEvent.doSomethingEvent += ShowSomething;//为事件实例化

myEvent.Invoke();//触发事件

Console.Read();

}

private static void ShowSomething()

{

Console.WriteLine("ShowSomething");

}

}

}

[复制代码](javascript:void(0);)

 3. 事件的发布者与订阅者实例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | //新郎官，充当事件发布者角色      public class Bridegroom      {          //自定义委托          public delegate void MarryHandler(string msg);          //定义事件          public event MarryHandler MarryEvent;            //发布事件          public void OnMarriageComing(string msg)          {              if(MarryEvent != null)              {                  MarryEvent(msg);              }          }      }    public class Friend      {          //属性          public string Name { get; set; }          //构造函数          public Friend(string name)          {              Name = name;          }          //事件处理函数，该函数需要符合 MarryHandler委托的定义          public void SendMessage(string message)          {              //输出通知信息              Console.WriteLine(message);              //对事件做出处理              Console.WriteLine(this.Name + "收到了，到时候准时参加");          }      }   static void Main(string[] args)          {              Bridegroom bridegroom = new Bridegroom();              Friend friend1 = new Friend("张三");              Friend friend2 = new Friend("李四");              Friend friend3 = new Friend("王五");                //使用 “+=” 来订阅事件              bridegroom.MarryEvent += new Bridegroom.MarryHandler(friend1.SendMessage);              bridegroom.MarryEvent += friend2.SendMessage;                //发出通知，此时只有订阅了事件的对象才能收到通知              bridegroom.OnMarriageComing("朋友门，我要结婚了，到时候准时参加婚礼！");              Console.WriteLine("---------------------------------------------");                //使用 “-=” 来取消事件订阅，此时李四将收不到通知              bridegroom.MarryEvent -= friend2.SendMessage;                //使用 “+=” 来订阅事件，此时王五可以收到通知              bridegroom.MarryEvent += friend3.SendMessage;              //发出通知              bridegroom.OnMarriageComing("朋友门，我要结婚了，到时候准时参加婚礼！");              Console.Read();          } |

　4.　输出结果

