Bob Liu

人生短暂，一旦过去，世上从此就不再有我们的任何痕迹了，得留下点什么。

刘振东，西洋名Bob，正宗燕赵人士，从事伟大光荣正确的软件开发工作已经不知道有多少年矣，擅长.Net外加SQL Server，喜欢阅读，围棋，羽毛球，阴阳八盘掌。

昵称：BobLiu
园龄：5年8个月
粉丝：113
关注：53
+加关注

委托, 匿名委托, Lambda表达式, 事件的本质, 以及Observer模式.

1.委托的本质

委托实现了面向对象的，类型安全的方法回调机制。委托看上去就一句话，很简单，但在本质上它是一个类，CLR和编译器在后台会将委托自动编译为一个类.
该类继承自System.MulticastDelegate类，该类维护一个委托列表，在调用多播委托时，将按照委托列表的委托顺序而调用的。该类包括一个接受两个参数的构造函数和3个重要方法：BeginInvoke、EndInvoke和Invoke。

下面是delegate委托的父类MulticastDelegate的部分代码:

```
public abstract class MulticastDelegate : Delegate
{
  public sealed override Delegate[] GetInvocationList();
  // Overloaded operators.
  public static bool operator ==(MulticastDelegate d1,
MulticastDelegate d2);
  public static bool operator !=(MulticastDelegate d1,
MulticastDelegate d2);
  // Used internally to manage the list of methods maintained
by the delegate.
  private IntPtr _invocationCount;
  private object _invocationList;
}
```

由此可以看出, 这个委托列表实际上就是一个Detegate类型的数组.

另外, 委托声明放置的位置还要注意, 因为C#中允许出现嵌套类，而委托本质上就是一个类, 所以下面namespace1, namespace2两个方案都正确, 但namespace2的委托就成为了Class A的嵌套类.

```
namespace namespace1

{
   public delegate void MyEventDelegate(int value);
   public class A
```

搜索
找找看
谷歌搜索
随笔分类
.Net, C#, Asp.net(83)
Active控件(2)
Appfabric(4)
Asp(1)
Asp.Net WebAPI(2)
Comuter Skill tips(10)
Daily Build
DeadLock
Design Patterns(2)
English(45)
Htc(Html Components)(3)
Html, Css, Javascript, jQuery, Xml,

Xsl, DOM(22)
IE兼容性(18)
IIS(1)
JQuery(3)
JSON(1)
Knockout.Js(7)
LINQ(3)
OData(2)
Office(2)
Open Xml(2)
Regex(1)
Require.Js(3)
SQL Server(16)
Sql Server全文检索(6)
VS.Net Trick(1)
WCF(2)
WebBrowser(4)
WF3(2)
WF4(3)
Windows Service(2)
WinForm(2)
WPF(3)
Xaml(1)
常用代码(8)
国际化(1)
文学诗歌(3)

随笔档案

2015年4月 (1)
2015年3月 (1)
2015年1月 (5)
2014年12月 (1)
2014年11月 (1)
2014年6月 (1)
2014年4月 (1)
2014年3月 (5)
2014年2月 (1)
2014年1月 (3)
2013年11月 (3)
2013年10月 (4)
2013年9月 (2)
2013年8月 (1)
2013年7月 (2)
2013年5月 (2)
2013年1月 (4)
2012年10月 (3)

```
  {

  }
}

namespace namespace2
{
  public class A
  {
      public delegate void MyEventDelegate(int value);
  }
}

public class A
{
    public class 委托
    {
    }
}
```

关于嵌套类的使用场合，看到有帖子说就是如果A类中需要有某些代码需要以面向对象的形式展现，而又不想被除调用类以外的类调用到就可以使用嵌套类，需要注意的是嵌套类最好不要使用public访问符，因为这样无任何意义。
如此一来，建议将委托放置在其他类外面声明.

一个完整的委托例子:

```
namespace LatestWebTestSite
{
  public delegate int DelegateAbc(int a, int b);
  public partial class _Default：System.Web.UI.Page
 {
  public int MethodAdd (int a, int b)
  {
   return a + b;
  }
  protected void Page_Load(object sender, EventArgs e)
  {
   DelegateAbc abc = MethodAdd;
   Response.Write(abc(100, 200));
  }
 }
}
```

2.匿名委托(方法)

其实就是将方法定义与委托变量赋值两步并作一步了，省了为方法起名了，写起来省事而已.

但编译后就知道了，实际上编译器又将一步分作两步，并给该方法起了个随机的名字，匿名方法也就是个语法糖而已.

2012年9月 (2)
2012年8月 (1)
2012年6月 (3)
2012年5月 (2)
2012年4月 (2)
2012年3月 (3)
2012年2月 (1)
2012年1月 (9)
2011年12月 (12)
2011年11月 (24)
2011年10月 (21)
2011年9月 (18)
2011年8月 (9)
2011年7月 (8)
2011年6月 (3)
2011年5月 (4)
2011年4月 (2)
2010年11月 (3)
2010年10月 (3)
2010年9月 (2)
2010年8月 (2)
2010年6月 (7)
2010年5月 (4)
2010年4月 (15)
2010年3月 (10)
2010年2月 (7)
2010年1月 (1)
2009年12月 (7)

tech

Application,Session,Cookie,ViewState和Cache区别

积分与排名

积分 - 141427
排名 - 1046

最新评论

1. Re:WebBrowser与IE的关系，如何设置WebBrowser工作在IE9模式下?
@BobLiu非常感谢你的回复，我按照你给的资料，进行了测试，但是还是没有解决这个问题。能否进一步帮助我一下呢？如果可以的话，麻烦加下我QQ：664113440，谢谢...
--lee_mosquito

2. Re:WebBrowser与IE的关系，如何设置WebBrowser工作在IE9模式

```
button1.Click += delegate
  {
     MessageBox.Show("Hello world.");
  };
```

3.Lambda表达式(=>)

第一眼看到觉得是个很古怪的一个名字，有点排斥它。

MSDN Lambda 表达式（http://msdn.microsoft.com/zh-cn/library/bb397687.aspx）上的定义：“Lambda 表达式”是一个匿名函数，它可以包含表达式和语句，并且可用于创建委托或表达式树类型。

所有 Lambda 表达式都使用 Lambda 运算符 =>，该运算符读为“goes to”。 该 Lambda 运算符的左边是输入参数（如果有），右边包含表达式或语句块。 Lambda 表达式 x => x * x 读作“x goes to x times x”。可以将此表达式分配给委托类型.

它是对匿名方法的进一步简化, 也是语法糖, 编译器会搞定将其一步步分开, 并起名的. Lambda表达式为LINQ提供了语法基础。

来个例子如下:

```
class Program
{
   delegate void delgateLzd(int iValue);

   static void Main(string[] args)
   {
      delgateLzd del = x => {

          Console.WriteLine(x * x);

      };

      del(5);

      Console.ReadKey();

   }
}
```

比较一下即可知道, 这里面用的是Lambda语句，看上去酷似匿名方法，实际上就是用=>这个符号取代了匿名方法里的delegate这个单词，可以说是对匿名方法的又一次简化。

原来想用委托要4步：1.定义委托，2.写一个符合委托的方法，3.建立委托实例并邦定刚才的方法, 4.调用委托。

到了匿名方法这个阶段就把2，3两步给合并了，不用写独立的方法了，大大方便了委托的使用。

而到了Lambda这里将匿名方法更进一步简化，连delegate都懒得写了，直接给个=>符号了事，如果需要参数就只把参数名列出来再接上=>，真是步步简化啊，到了这一步，如果是一个c#1的程序员穿越过来，看到这段代码，一定会大吃一惊，这里除了开始的委托类型以

下?
@lee_mosquito修改注册表不能彻底解决问题，最彻底是修改Htm页面，方法是，在Head->Title下添加，这样可确保HTM页面工作在IE9标准文档模式下。具体请看：...
--BobLiu
3. Re:WebBrowser与IE的关系，如何设置WebBrowser工作在IE9模式下?
楼主，请教你一个问题。在六十四位系统下面，500)
this.width=500;"/>
我这样修改成ie8，但是貌似没成功
--lee_mosquito
4. Re:创建,安装,调试 Windows Service
@勇哥哥不好意思回复晚了，兄弟的wcf如果部署在有安全证书保护的iis站点内，还需要IgnoreCertificateErrorHandler来规避证书问题。using System.Net;usin.....
--BobLiu
5. Re:创建,安装,调试 Windows Service
请教楼主个问题，兄弟最近也在弄windows服务里调用wcf服务，windows服务确定安装运行是没问题的,但是就是wcf服务好像没有数据出来，wcf服务也是没问题的,我在windows应用程序里已经.....
--勇哥哥

阅读排行榜

1. 再谈IE的浏览器模式和文档模式(31324)
2. WebBrowser与IE的关系，如何设置WebBrowser工作在IE9模式下?(21298)
3. IE6-IE9兼容性问题列表及解决办法总结(11031)
4. VS2010的智能提示没有了的可能原因(10411)
5. SQL Server 2008 创建索引视图(物化视图) 的一点总结(10083)

评论排行榜

1. WebBrowser与IE的关系，如何设置WebBrowser工作在IE9模式下?(29)
2. 再谈IE的浏览器模式和文档模式(20)
3. IE6-IE9兼容性问题列表及解决办法总结(11)
4. Sql Server 2005/2008 SqlCacheDependency查询通知的

外，已经几乎看不到委托的影子了，但实际上，背地里，编译器还是给编译成了委托，这个语法糖是越来越甜，封装的越来越抽象了。

4.事件的本质

事件建立在委托之上，只有了解了委托才能明白事件的原理。事件是对委托的封装，从委托的示例中可知，在客户端可以随意对委托进行操作，一定程度上破坏了面向的对象的封装机制，因此事件实现了对委托的封装。
事件其实就是委托类型的变量，也就是说如果想声明一个事件的话，你必须先声明一个委托类型的变量，然后将event关键字放在声明体的前部，例如:

```
//声明事件委托
public delegate void CalculateEventHandler(object sender,CalculateEventArgs e);
//定义事件成员，提供外部绑定
public event CalculateEventHandler MyCalculate;
```

一个完整的事件例子:

```
namespace UseEventExample
{
   public delegate void MyEventDelegate(int value, int value2);

   class Program
   {
      public static void MethodAdd(int a, int b)
      {
         Console.WriteLine(a + b);
      }

      public static event MyEventDelegate abc; //定义一个事件

      static void Main(string[] args)
      {
         abc = MethodAdd;
         abc(100, 200);
         Console.ReadKey();

      }
   }
}
```

比较一下委托与事件的两个例子就可以比较出来,

委托: **DelegateAbc abc = MethodAdd;**

事件: **Public static event MyEventDelegate abc;**
abc = MethodAdd;

事件实际上就是委托的变量而已，它的作用就是提高委托的封装性。

5.Observer模式

使用总结(11)
5. Lucene.Net, SQL Server 2008全文检索, Like模糊查询的一点心得(11)

使用委托和事件实现的观察者模式, 让观察者和被观察者不用任何耦合, 下面是猫叫鼠跑的例子代码:

5.1 Cat

```
namespace ConsoleApplication1
{
   public delegate void CatDelegate ();

   public class Cat
   {
      public event CatDelegate catEvent;

      public void CatCry()
      {
         Console.WriteLine(" Cat is crying");

          if (catEvent!=null)
          {
              catEvent();
          }
      }

   }
}
```

5.2 Mouse

```
namespace ConsoleApplication1
{
   public class Mouse
   {
      public string MouseName { set; get; }

      public Mouse(string mouseName)
      {
         this.MouseName = mouseName;
      }

      public void run()
      {
         Console.WriteLine(MouseName + " start to run");
      }
   }
}
```

5.3 Main

```
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         Mouse mouse = new Mouse("mouse1");

         Cat cat = new Cat();
         cat.catEvent += mouse.run;
         cat.CatCry();

         Console.ReadKey();
```

```
        }
    }
}
```

当然, 正式的代码要抽象出observer, subject的接口, 然后让鼠, 猫分别实现之.

作者：BobLiu
邮箱：lzd_ren@hotmail.com
出处：http://www.cnblogs.com/liuzhendong

分类：.Net, C#, Asp.net, Design Patterns

标签：委托, 匿名委托, Lambda表达式, 事件, Observer模式

« 上一篇：SQL Server 2008事务日志的[RowLog Contents 0] 字段没人能解析出来?
» 下一篇：漫谈面向对象基石之开闭原则（OCP）(转)

posted on 2011-08-31 17:03 BobLiu 阅读(229) 评论(0) 编辑 收藏