

C# 事件

2019年9月18日 23:19

事件是对委托的一种封装。我们直接使用委托，要在外部向类的委托中添加方法，必须让委托实例公开，这个时候如果外部直接使用‘=’赋值，会覆盖之前添加的所有方法，只剩当前方法，影响到其他程序，委托也可以直接在类外部执行，影响其他部分。事件的优点就在与它只能使用‘+=’或‘-=’向委托中添加或删除方法，不能直接赋值，且事件只能在类的内部调用，外部只能订阅此事件，不能执行。

1. 事件的定义

在定义一个事件前，我们要定义一个相应的委托类型，一般以EventHandler结尾。也可以使用程序集中预先定义好的委托，事件最常见的委托类型是EventHandler类型的委托，在System命名空间下，还有其他在控件中经常使用的SelectionChanged、TextChanged等。假定现在有一个自定义的颜色选择控件ColorSelector，选择改变后外部有关联的控件颜色要发生改变，我们可以自定义一个ColorChanged委托。

```
public delegate void ColorChangedEventHandler(object sender, SelectionChangedEventArgs e);
public delegate void EventHandler(object sender, EventArgs e);
public delegate void SelectionChangedEventHandler(object sender, SelectionChangedEventArgs e);
public delegate void TextChangedEventHandler(object sender, TextChangedEventArgs e);
```

根据委托定义事件

```
public event ColorChangedEventHandler ColorChanged;
public event SelectionChangedEventHandler SelectionChanged;
```

2. 内部事件定义及使用

在颜色选择控件内部有一个ComboBox，用来选择指定的几种颜色。定义一个颜色改变事件供关联的外部控件订阅。其实这种情况用SelectionChangedEventHandler就可以了，不过为了示例这里还是自定义一个委托。

```
public delegate void ColorChangedEventHandler(object sender, SelectionChangedEventArgs e);
public event ColorChangedEventHandler ColorChanged; //颜色改变事件，供外部订阅
private void CbColor_SelectionChanged(object sender, SelectionChangedEventArgs e)
//ComboBox选中项改变事件
{
    ColorChanged?.Invoke(sender, e); //选中项改变时执行外部订阅事件
}
```

3. 外部订阅事件

外部其他控件在初始化或加载时订阅颜色改变事件。

```
public Map()
{
```

```
        InitializeComponent();
        colorSelector.ColorChanged += Color_ColorChanged;
    }
private void Color_ColorChanged(object sender, SelectionChangedEventArgs e)
{
    this.Background = colorSelector.SelectedColor;//颜色选择控件颜色改变后处理
}
代码
```