委托的完整声明

```
using System;
using System.Threading;
using System.Timers;
namespace LeetCode.Properties
{
    public class 事件
    {
        static void Main(string[] args)
        {
            Customer customer=new Customer();
            Waiter waiter=new Waiter();

            customer.Order += waiter.Action;
            customer.Action();
            customer.PayTheBill();
        }
    }



    class OderEventArgs:EventArgs
    {
        public string DishName { get; set; }
        public string Size { get; set; }
    }

    delegate void OderEventHandle(Customer customer,OderEventArgs e);

    /// <summary>
    /// 顾客 事件的拥有者
    /// </summary>
    class Customer
    {
```

```
public OderEventHandle OderEventHandle;

public event OderEventHandle Order
{
    add { this.OderEventHandle += value; }

    remove { this.OderEventHandle -= value; }
}
public double Bill { get; set; }

public void PayTheBill()
{
    Console.WriteLine("I will pay${0}",this.Bill);
}

public void WalkIn()
{
    Console.WriteLine("Walk into the restaurant");
}

public void SitDown()
{
    Console.WriteLine("Sit Down");
}

public void Think()
{
    for (int i = 0; i <5; i++)
    {
        Console.WriteLine("Let me think.....");
        Thread.Sleep(1000);
    }

    if (this.OderEventHandle!=null)
    {
      OderEventArgs e=new OderEventArgs();
      e.DishName = "GongBaoJiDing";
```

```
          e.Size = "small";

          this.OderEventHandle.Invoke(this,e);
        }
    }

    public void Action()
    {
        Console.ReadLine();
        this.WalkIn();
        this.SitDown();
        this.Think();
    }

}

/// <summary>
/// 事件的响应者
/// </summary>
class Waiter
{
    public void Action(Customer customer, OderEventArgs e)
    {
        Console.WriteLine("I will serve you the dish-{0}",e.DishName);
        double price = 10;

        switch (e.Size)
        {
            case "small":
                price = price * 0.5;
                break;
            case "large":
                price = price * 1.5;
                break;
                default:
                    break;
        }
```

```
            customer.Bill += price;
        }
    }
}
```

事件的简化声明

```
namespace LeetCode.Properties
{
    public class 事件
    {
        static void Main(string[] args)
        {
            Customer customer=new Customer();
            Waiter waiter=new Waiter();

            customer.Order += waiter.Action;
            customer.Action();
            customer.PayTheBill();
        }
    }
    class OderEventArgs:EventArgs
    {
        public string DishName { get; set; }
        public string Size { get; set; }
    }

    delegate void OderEventHandle(Customer customer,OderEventArgs e);

    /// <summary>
    /// 顾客 事件的拥有者
    /// </summary>
    class Customer
    {
        //事件的简化声明
        public event OderEventHandle Order;
public double Bill { get; set; }

public void PayTheBill()
{
    Console.WriteLine("I will pay${0}",this.Bill);
}

public void WalkIn()
{
    Console.WriteLine("Walk into the restaurant");
}

public void SitDown()
{
    Console.WriteLine("Sit Down");
}

public void Think()
{
    for (int i = 0; i <5; i++)
    {
        Console.WriteLine("Let me think.....");
        Thread.Sleep(1000);
    }

    if (this.Order!=null)
    {
      OderEventArgs e=new OderEventArgs();
      e.DishName = "GongBaoJiDing";
      e.Size = "small";

      this.Order.Invoke(this,e);
    }
}

public void Action()
{
        Console.ReadLine();
        this.WalkIn();
        this.SitDown();
        this.Think();
    }

}

/// <summary>
/// 事件的响应者
/// </summary>
class Waiter
{
    public void Action(Customer customer, OderEventArgs e)
    {
        Console.WriteLine("I will serve you the dish-{0}",e.DishName);
        double price = 10;

        switch (e.Size)
        {
            case "small":
                price = price * 0.5;
                break;
            case "large":
                price = price * 1.5;
                break;
                default:
                    break;
        }

        customer.Bill += price;
    }
}
```