

Лабораторная работа №6

Цель работы:

Познакомиться с использованием делегатов и событий в языке программирования C#, научиться создавать собственные события для избранных классов.

- 1) Для иерархии классов, построенной в лабораторной работе №2-3, придумать и реализовать не менее двух оригинальных событий для каждого класса иерархии.
- 2) Защитить лабораторную работу, ответив на вопросы преподавателя.

Необходимые теоретические сведения

Делегаты

Делегат — это объект, имеющий ссылку на метод. Делегат позволяет выбрать вызываемый метод во время выполнения программы. Фактически значение делегата — это адрес области памяти, где находится точка входа метода.

Важным свойством делегата является то, что он позволяет указать в коде программы вызов метода, но фактически вызываемый метод определяется во время работы программы, а не во время компилирования.

Делегат объявляется с помощью ключевого слова `delegate`, за которым указывается тип возвращаемого значения, имя делегата и список параметров вызываемых методов.

Синтаксис:

```
delegate тип_возвращаемого_значения имя_делегата(список_параметров);
```

Характерной особенностью делегата является возможность его использования для вызова любого метода, который соответствует подписи делегата. Это дает возможность определить во время выполнения программы, какой из методов должен быть вызван. Вызываемый метод может быть методом экземпляра, ассоциированным с объектом, либо статическим методом, ассоциированным с классом. Метод можно вызвать только тогда, когда его подпись соответствует подписи делегата.

Многоадресность делегатов

Многоадресность — это способность делегата хранить несколько ссылок на различные методы, что позволяет при вызове делегата инициировать эту цепочку методов.

Для создания цепочки методов необходимо создать экземпляр делегата, и пользуясь операторами `+` или `+=` добавлять методы к цепочке. Для удаления метода из цепочки используется оператор `-` или `-=`. Делегаты, хранящие несколько ссылок, должны иметь тип возвращаемого значения `void`.

Обработка событий

Событие — это автоматическое извещение о каком-либо произошедшем действии. События являются членами класса и объявляются с использованием ключевого слова `event`. Механизм событий основан на использовании делегатов.

Синтаксис:

```
event имя_делегата имя_объекта;
```

Широковещательные события

События могут активизировать несколько обработчиков, в том числе те, что определены в других объектах. Такие события называются широковещательными. Широковещательные события создаются на основе многоадресных делегатов.

Пример:

```
//Объявление делегата, на основе которого будет определено событие.
delegate void MyEventHandler();

//Объявление класса, в котором иницируется событие.
class MyEvent
{
    public event MyEventHandler activate;
    // В этом методе иницируется событие.
    public void fire()
    {
        if (activate != null) activate();
    }
}

class X
{
    public void Xhandler()
    {
        Console.WriteLine("Событие получено объектом класса X.");
    }
}

class Y
{
    public void Yhandler()
    {
        Console.WriteLine("Событие получено объектом класса Y.");
    }
}

class EventDemo
{
    static void handler()
    {
        Console.WriteLine("Событие получено объектом класса EventDemo.")
    }

    public static void Main()
    {
        MyEvent evt = new MyEvent();
        X xOb = new X();
        Y yOb = new Y();

        //Добавление методов handler (), Xhandler()
        //и Yhandler() в цепочку обработчиков события
        evt.activate += new MyEventHandler(handler);
        evt.activate += new MyEventHandler(xOb.Xhandler);
        evt.activate += new MyEventHandler(yOb.Yhandler);

        evt.fire();
        Console.WriteLine();

        evt.activate -= new MyEventHandler(xOb.Xhandler);
    }
}
```

```
    evt.fire();  
  }  
}
```