

## **ЛЕКЦИЯ №6 СОБЫТИЯ**

Москва, 2019

## События

Событие - это действие, выполняемое при выполнении указанного условия. Оно уведомляет всех своих подписчиков о действии, которое будет выполнено

События C# используется в классе для предоставления уведомлений клиентам этого класса, когда что-то происходит с его объектом.

События объявляются с использованием делегатов.

класс, содержащий определение  
событие и его делегат называется **издателем**.

С другой стороны, класс, который принимает событие и  
предоставляет

обработчик события называется **подписчиком**

```
event delegate_type OnEventName;
```

## События

Событие всегда является членом данных класса или структуры.  
Событие не может быть объявлено внутри метода

```
public event Action<object> Alert;
```

```
namespace eventDemo
{
    delegate void VisitEventHandler();

    ссылка: 0
    class Person
    {
        public event VisitEventHandler visit;
    }
}
```

## События

Событие всегда подписывается с помощью +=, например

```
myRoom.Alert += OnAlert
```

Событие отменяется с помощью объекта `object.EventName -= MethodName`

`myRoom.Temperature = 65;` Событие оповещения будет вызвано, потому что температура в помещении более 60. Следовательно, условие удовлетворяет и событие должно вызывать

Событие всегда вызывается внутри класса, где он определен. В отличие от делегатов, события не могут вызываться за пределами класса, где они определены

## События класса Account

```
class Account
{
    // Событие, возникающее при выводе денег
    public event Action<string> Withdrawn;
    // Событие, возникающее при добавление на счет
    public event Action<string> Added;
    int _sum; // Переменная для хранения суммы

    ссылка: 0
    public int CurrentSum
    {
        get { return _sum; }
    }

    ссылка: 0
    public void Put(int sum)
    {
        _sum += sum;
        if (Added != null)
            Added("На счет поступило "+sum);
    }
}
```

## Встроенные делегаты для реализации событий

### EventHandler

EventHandler - это делегат, определенный в пространстве имен System. Этот делегат определяет метод void для типа возврата.

Его первый параметр имеет тип System.Object, который ссылается на экземпляр (где событие было определено), что вызывает событие.

Его второй параметр имеет тип EventArgs, который содержит данные о событиях. Если событие нет данных для передачи, вторым параметром является просто значение поле EventArgs.Empty. Однако, если оно имеет значение для передачи, оно будет инкапсулированный в производный тип EventArgs

```
namespace System
```

```
{
```

```
public delegate void EventHandler(object sender, EventArgs e);
```

```
}
```

## Встроенные делегаты для реализации событий

```
class Pong
{
    public event EventHandler ItWasPong;
    // обработчик события ItWasPing
    public void OnWasPing(object sender, EventArgs e)
    {
        Console.WriteLine("Pong received Ping");
        Console.ReadLine();
        // отправляю событие ItWasPong всем подписчикам
        StartSendEvent();
    }
    public void StartSendEvent()
    {
        if (ItWasPong != null)
        {
            ItWasPong(this, EventArgs.Empty);
        }
    }
}
```

## Встроенные делегаты для реализации событий

```
class Ping
{
    public event EventHandler ItWasPing;

    // обработчик события ItWasPong
    public void OnWasPong(object sender, EventArgs e)
    {
        Console.WriteLine("Ping received Pong");
        Console.ReadLine();
        // отправляю событие ItWasPing всем подписчикам
        StartSendEvent();
    }
    public void StartSendEvent()
    {
        if (ItWasPing != null)
        {
            ItWasPing(this, EventArgs.Empty);
        }
    }
}
```