

СОБЫТИЯ

В прошлой теме мы рассмотрели, как с помощью делегатов можно создавать механизм обратных вызовов в программе. Однако С# для той же цели предоставляет более удобные и простые конструкции под названием **события**, которые сигнализируют системе о том, что произошло определенное действие.

События объявляются в классе с помощью ключевого слова **event**, после которого идет название делегата:

```
// Объявляем делегат
public delegate void AccountStateHandler(string message);
// Событие, возникающее при выводе денег
public event AccountStateHandler Withdrawn;
```

Связь с делегатом означает, что метод, обрабатывающий данное событие, должен принимать те же параметры, что и делегат, и возвращать тот же тип, что и делегат.

Итак, посмотрим на примере. Для этого возьмем класс Account из прошлой темы и изменим его следующим образом:

```
class Account
{
    // Объявляем делегат
    public delegate void AccountStateHandler(string
message);

    // Событие, возникающее при выводе денег
    public event AccountStateHandler EventWithdrawn;

    // Событие, возникающее при добавление на счет
    public event AccountStateHandler EventAdded;

    // Переменная для хранения суммы
    int _sum;

    public Account(int sum)
    { _sum = sum; }

    public int CurrentSum
    { get { return _sum; }
}
}
```

```

public void PutMethod(int sum)
{
    _sum += sum;
    if (EventAdded != null)
        EventAdded($"На счет поступило {sum}");
}

public void WithdrawMethod(int sum)
{
    if (sum <= _sum)
    {
        _sum -= sum;
        if (EventWithdrawn != null)
            EventWithdrawn($"Сумма {sum} снята со
счета");
    }
    else
    {
        if (EventWithdrawn != null)
            EventWithdrawn("Недостаточно денег на
счете");
    }
}
}

```

Здесь мы определили два события:

- **EventWithdrawn** и
- **EventAdded**.

Оба события объявлены как экземпляры делегата **AccountStateHandler**, поэтому для обработки этих событий **потребуется метод, принимающий строку в качестве параметра**.

Затем в методах **PutMethod** и **WithdrawMethod** мы вызываем эти события.

Перед вызовом мы проверяем, закреплены ли за этими событиями обработчики (**if (WithdrawMethod != null)**). Так как эти события представляют делегат **AccountStateHandler**, принимающий в качестве параметра строку, то и при вызове событий мы передаем в них строку.

Теперь используем события в основной программе:

```
using System;
namespace EventConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Account account = new Account(200);

            // Добавляем обработчики события
            // и определяем метод для события MyShow_Message
            // MyShow_Message соответствует делегату
            account.EventAdded += MyShow_Message;
            account.EventWithdrawn += MyShow_Message;

            account.WithdrawMethod(100);

            // Удаляем обработчик события
            account.EventWithdrawn -= MyShow_Message;

            // Снятие денег со счета
            // вызывается событие EventWithdrawn
            account.WithdrawMethod(50);

            // Добавление денег
            // вызывается событие EventAdded
            account.PutMethod(150);

            Console.ReadLine();
        }

        // метод для события
        private static void MyShow_Message(string message)
        {
            Console.WriteLine(message);
        }
    }
}
```

Для прикрепления обработчика события к определенному событию используется операция += и соответственно для открепления - операция -=:

```
событие += метод_обработчика_события.
```

Опять же обращаю внимание, что метод обработчика должен иметь такие же параметры, как и делегат события, и возвращать тот же тип. В итоге мы получим следующий консольный вывод:

```
Сумма 100 снята со счета
```

```
На счет поступило 150
```