

13 СОБЫТИЯ

Задание 1 и задание 2. Создать 3 метода для вывода трех различных сообщений. Создать делегат. В программе вызывать все три метода при помощи делегата. На основе своей программы создать метод, в котором делегат будет параметром.

Листинг программы:

```
class TestStrAndDelegate
{
    public void StrPrint()
    {
        Console.WriteLine("Тест делегата в качестве параметра");
    }
    public void StrPrintx2()
    {
        Console.WriteLine("Тест делегата в качестве параметра");
    }
    public void DelegatePrint(Message msg) //делегат в качестве параметра
    {
        msg(); //выполняем делегат (выполняем методы, которые мы занесли
в делегат)
    }
}
class MyEvent
{
    public event MyEventHandler activate; //public event имя_делегата
имя_объекта

    public void fire() //запуск события
    {
        if(activate != null)
        {
            activate();
        }
    }
}
```

					УП 2-40 01 01.31 ТП.2471.22.13			
Изм.	Лист	№ докум	Подп.	Дата				
Разраб.	Мушинский М.С				СОБЫТИЯ			
Пров.	Толочко П.С.							
Н.контр.								
Утв.								
						Лит	Лист	Листов
						Гродненский ГКТТид		

```

    }
}
}
class X
{
    public void Xhandler()
    {
        Console.WriteLine("Res = X");
    }
}
class Y
{
    public void Yhandler()
    {
        Console.WriteLine("Res = Y");
    }
}
class EventDemo
{
    static void Handler()
    {
        Console.WriteLine("Res = Event");
    }
    public static void Main()
    {
        try
        {

```

MyEvent evt = new MyEvent();
X xObj = new X();
Y yObj = new Y();

evt.activate += new MyEventHandler(xObj.Xhandler); //в событие передаем делегат MyEventHandler, в скобках которого указываем метод определенного объекта

evt.activate += new MyEventHandler(yObj.Yhandler);
evt.activate += new MyEventHandler(Handler);
evt.fire(); //активируем событие, тем самым запуская методы объектов, переданных в него (в событие)
Console.WriteLine();

					УП 2-40 01 01.31 ТП.2471.22.13	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

```

TestStrAndDelegate testStr = new TestStrAndDelegate();
Message msg = testStr.StrPrint;
msg += testStr.StrPrintx2;

testStr.DelegatePrint(msg);
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
}
}
delegate void MyEventHandler();
delegate void Message();

```

Таблица 13.1 – Входные и выходные данные

Входные данные	Выходные данные
	Res = X Res = Y Res = Event Тест делегата в качестве параметра Тест делегата в качестве параметра

Анализ результатов:

```

Res = X
Res = Y
Res = Event

Тест делегата в качестве параметра
Тест делегата в качестве параметра

```

Рисунок 13.1 – Результат работы программы
Источник: собственная разработка