

Делегаты. Основные понятия

Основные понятия

Делегат - класс, который позволяет хранить в себе ссылку на метод с определённой сигнатурой произвольного класса

Экземпляр делегата — это такой объект, который может ссылаться на методы. При вызове объекта вызывается метод, с которым связан объект.

Описание делегата

Ключевое слово delegate

Имя делегата (по аналогии с именем класса)

delegate тип имя (аргументы);

Идентификатор типа результата метода,
на который ссылается делегат

Тип и формальное название аргументов,
которые должны быть у метода, для того,
чтобы на этот метод мог ссылаться делегат

```
delegate char MyDelegate(int k, string txt);
```

Создание объекта делегата

```
Делегат переменная = new делегат (метод);
```

```
MyDelegate meth = new MyDelegate(метод);
```

для выполнения ссылки на метод достаточно указать имя этого метода (без круглых скобок и аргументов).

- Ссылка на нестатический метод: объект.метод.
MyDelegate meth = new MyDelegate(obj.method);
- Ссылка на статический метод: класс.метод
MyDelegate meth=new MyDelegate(SomeClass.method);

Вызов экземпляра делегата

Производится вызов объекта делегата по его имени и с передачей ему необходимых аргументов

```
MyDelegate meth = new MyDelegate(obj.method);  
meth(number,text); // obj.method(number,text).
```

Количество и тип аргументов, которые следует передать экземпляру делегата при вызове, определяются тем, как описан соответствующий делегат.

2 способ создания объекта делегата

объявление переменной, тип которой является делегатом, и присвоение такой переменной ссылки на метод

делегат переменная = метод;

`MyDelegate meth=obj.method;`

`MyDelegate meth=SomeClassss.method;`

Листинг. Пример использования делегата

```
using System;

//объявление делегата
delegate char Mydelegate(int k, string txt);

namespace DelegateLesson
{
    class SomeClass
    {
        public int code;
        public SomeClass (int n)
        { code = n;}
        // нестатический метод с двумя переменными
        public char GetChar(int k, string txt)
        {
            return (char) (txt[k]+code);
        }
        //статический метод с двумя аргументами
        public static char GetFirst(int k, string txt)
        {
            return txt[k];
        }
    }
}
```

Продолжение листинга

```
//класс с главным методом
internal class Program
{
    //статический метод с двумя аргументами
    static char GetLast(int k, string txt)
    { return txt[txt.Length-1]; }

    static void Main(string[] args)
    {
        SomeClass obj = new SomeClass(5); //создание объекта
        Mydelegate meth = new Mydelegate(obj.GetChar); //создание экземпляра делегата
        Console.WriteLine($"СИМВОЛ {meth(4, "world")}"); //вызов экземпляра делегата
        obj.code = 1; //присваивание значения полю объекта
        Console.WriteLine($"СИМВОЛ {meth(4, "world")}"); //вызов экземпляра делегата
        meth = SomeClass.GetFirst; //присваивание нового значения переменной делегата
        Console.WriteLine($"СИМВОЛ {meth(2, "world")}"); //вызов экземпляра делегата
        meth = GetLast; //присваивание нового значения переменной делегата
        Console.WriteLine($"СИМВОЛ {meth(1, "world")}"); //вызов экземпляра делегата
    }
}
```

Консоль отладки Microsoft Visual Studio

```
СИМВОЛ i
СИМВОЛ e
СИМВОЛ r
СИМВОЛ d
```