

C# Essential

Делегаты

C# Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал
на [TestProvider.com](http://testprovider.com)

C# Essential

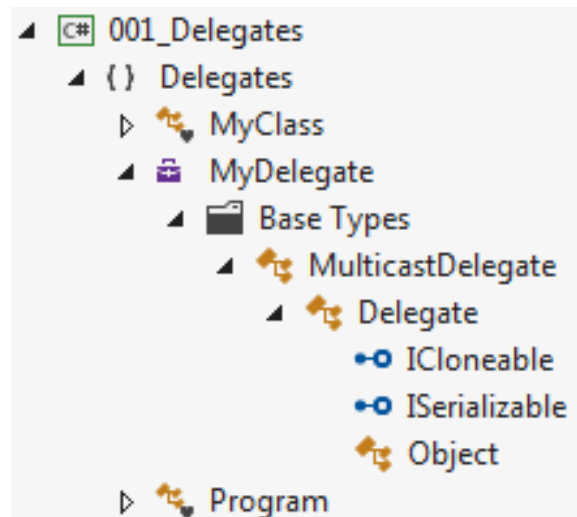
Тема

Делегаты

Делегаты

Delegates

Делегат (delegate) – это разновидность объектов которые содержат в себе указатели на методы.



Все делегаты, являются производными от абстрактного класса `System.MulticastDelegate`, который в свою очередь наследуется от абстрактного класса `Delegate`.

Делегаты

Сигнатура

Экземпляр делегата может ссылаться на любой статический метод или метод экземпляра – при условии, что сигнатура метода совпадает с сигнатурой делегата.

```
public delegate void MyDelegate();

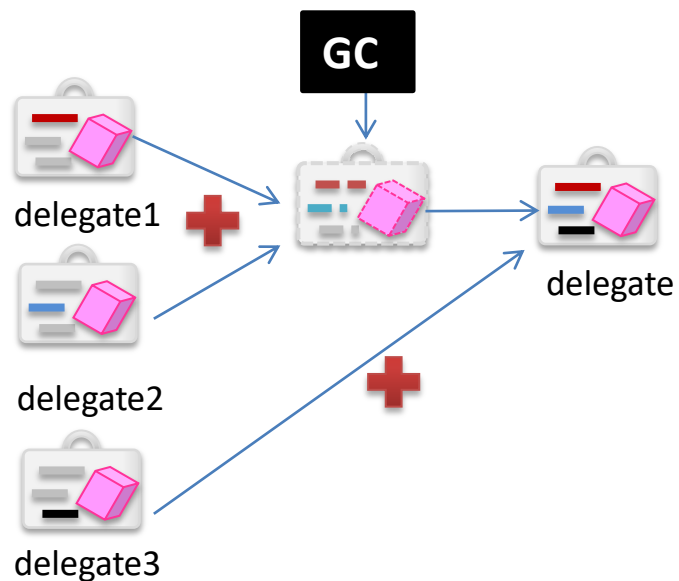
class Program
{
    static void Main()
    {
        MyDelegate myDelegate = new MyDelegate(MyClass.Method);
        myDelegate.Invoke();
        myDelegate();
    }
}
```

```
static class MyClass
{
    public static void Method()
    {
    }
}
```

Делегаты

Комбинированные делегаты

```
static void Main()  
{  
    MyDelegate delegate = null;  
    MyDelegate delegate1 = new MyDelegate(Method1);  
    MyDelegate delegate2 = new MyDelegate(Method2);  
    MyDelegate delegate3 = new MyDelegate(Method3);  
  
    delegate = delegate1 + delegate2 + delegate3;  
}
```



Анонимные функции

Anonymous Functions

Анонимные методы – это оператор или выражение "**inline**", которое можно использовать каждый раз, когда ожидается тип делегата. Ее можно использовать для инициализации именованного делегата или подставить вместо типа именованного делегата в качестве параметра метода.



Существует два типа анонимных методов (лямбда-методов) – это **Лямбда-операторы** и **Лямбда-выражения**.

Лямбда-операторы

Lambda-operators

Во всех лямбда-выражениях используется лямбда-оператор \Rightarrow , который читается как "переходит в". Левая часть лямбда-оператора определяет параметры ввода (если таковые имеются), а правая часть содержит выражение или блок оператора. Лямбда-выражение $x \Rightarrow x * x$ читается как "x переходит в x, x раз".

Лямбда-выражение

Lambda expression

Лямбда-выражение – это анонимная функция, которая содержит выражения и операторы и может использоваться для создания делегатов.

```
static void Main()
{
    MyDelegate myDelegate;

    myDelegate = delegate(int x) { return x * 2; };
    myDelegate = (x) => { return x * 2; };
    myDelegate = x => x * 2;

    int result = myDelegate(4);
}
```

Лямбда-Оператор – это многооператорное лямбда выражение.
Лямбда-Выражение – это однооператорный лямбда оператор.

Анонимные методы

Anonymous Methods

Создание анонимных методов является, по существу, способом передачи блока кода в качестве параметра делегата.

```
static void Main()
{
    MyDelegate myDelegate = delegate { Console.WriteLine("Hello world!"); };
    myDelegate();
}
```

Делегаты

Техника предположения делегатов

```
class Program
{
    static void Main()
    {
        // MyDelegate myDelegate = new MyDelegate(MyClass.Method);

        MyDelegate myDelegate = MyClass.Method;
        myDelegate();
    }
}
```



Для упрощения записи, можно использовать технику предположения делегатов.

Делегаты

Правила использования

Следующие правила применимы к области действия переменной в лямбда-выражениях.

- Захваченная переменная не будет уничтожена сборщиком мусора до тех пор, пока делегат, который на нее ссылается, не выйдет за границы области.
- Переменная, введенная в лямбда-выражение, невидима во внешнем методе.
- Лямбда-выражение не может непосредственно захватывать параметры `ref` или `out` из включающего их метода.
- Лямбда-выражение не может содержать оператор `goto`, оператор `break` или оператор `continue`, для которых, метка перехода находится вне тела либо в теле содержащейся анонимной функции.

C# Essential

Q&A

Информационный видеосервис для разработчиков программного обеспечения

