

# Анонимные методы

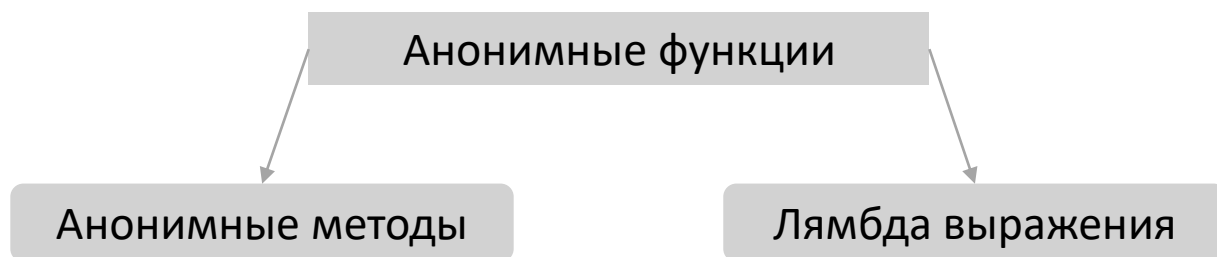
[Ссылка на примеры](#)

# Основные понятия

**Анонимный метод** — это оформленный особым образом блок программного кода, который можно присвоить в качестве значения переменной типа делегата.

## Особенности:

- описание анонимного метода начинается с ключевого слова `delegate`
- нет имени
- тип результата не указывается
- Не может быть вызван сам по себе



## Шаблон анонимного метода

```
delegate(аргументы)  
{  
    // тело метода  
}
```

# Пример 1. использование анонимных методов

```
using System;

//объявление первого делегата
delegate int Alpha(int n);

//объявление второго делегата
delegate void Bravo(string t);

namespace AnonymousMethods
{
    class MyClass
    {
        public int number;
        public MyClass(int n)
        {
            number = n;
        }
        public Alpha method; //поле класса - ссылка на экз делегата
    }
}
```

```
internal class Program
{
    static void Main(string[] args)
    {
        //создание объектов
        MyClass obj1 = new MyClass(100);
        MyClass obj2 = new MyClass(200);
        obj1.method = delegate (int n) //полю объекта значением присваивается анонимный метод
        { return obj1.number + n; };
        obj2.method = delegate (int n) //полю объекта значением присваивается анонимный метод
        { return obj2.number + n; };

        int x = 80;
        //вызов экземпляра делегата
        Console.WriteLine($"obj1.method: {obj1.method(x)}");
        Console.WriteLine($"obj2.method: {obj1.method(x)}");
    }
}
```

```
Bravo Show; //объявление переменной типа делегата

Show = delegate (string t) //присваивание переменной анонимного метода
{ Console.WriteLine($"текст: \"{t}\""); };

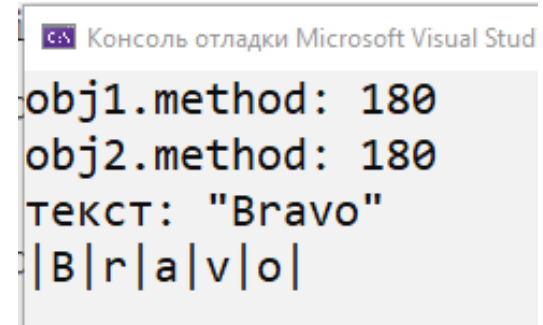
Show("Bravo"); //вызов экземпляра делегата

Show = delegate (string t) //присваивание переменной анонимного метода:
{
    for (int i = 0; i < t.Length; i++)
    { Console.Write($"|{t[i]}"); }
    Console.WriteLine("|");
};

//вызов экземпляра делегата
Show("Bravo");

}

}
```



Консоль отладки Microsoft Visual Stud

```
obj1.method: 180
obj2.method: 180
текст: "Bravo"
|B|r|a|v|o|
```

## Пример 2. анонимный метод как аргумент

```
using System;


// объявление делегата
delegate char MyDelegate(int n);

namespace AnonymMethAsArg
{
    class MyClass
    {
        public char symbol;
        public MyDelegate get; //экз делегата в качестве поля класса
        public MyClass(char s, MyDelegate md)
        {
            symbol=s;
            get=md;
        }
        public void Set(MyDelegate md) //экз делегата в качестве параметра метода
        {
            get=md;
        }
    }
}
```

```
internal class Program
{
    static void Main(string[] args)
    {
        //создание объекта. вторым аргументом является анонимный метод
        MyClass obj = new MyClass('K', //символьный аргумент
                                   delegate (int n) //анонимный метод
                                   { return (char)('A' + n); }
                                   );

        Console.WriteLine($"Символ: \"{obj.get(3)}\""); // вызов экземпляра делегата
        obj.Set( //вызов метода, аргументом которому передан анонимный метод
                delegate (int n)
                { return (char)(obj.symbol + n); }
                );

        Console.WriteLine($"Символ: \"{obj.get(3)}\"»); //вызов экз делегата
    }
}
```

 Консоль отладки Micro

Символ: 'D'

Символ: 'N'

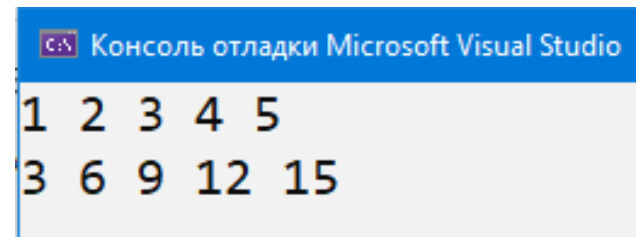
## Пример 3. анонимный метод как результат

```
using System;
delegate int MyDelegate();
namespace AnonymAsResult
{
    internal class Program
    {
        static MyDelegate calculate(int n) //метод возвр ссылку на экз делегата
        {
            int count = 0;
            return delegate () //результат через анонимный метод
            {
                count+=n;
                return count;
            };
        }
    }
}
```



```
static void Main(string[] args)
{
    MyDelegate next = calculate(1); //экземпляр делегата
    for (int i = 1; i<=5; i++)
    {
        Console.Write(next()+" "); //вызов экз делегата
    }
    Console.WriteLine();

    //новое значение переменной типа делегата
    next=calculate(3);
    for (int i = 1; i<=5; i++)
    {
        //вызов экз делегата
        Console.Write(next() + " ");
    }
    Console.WriteLine();
}
}
```



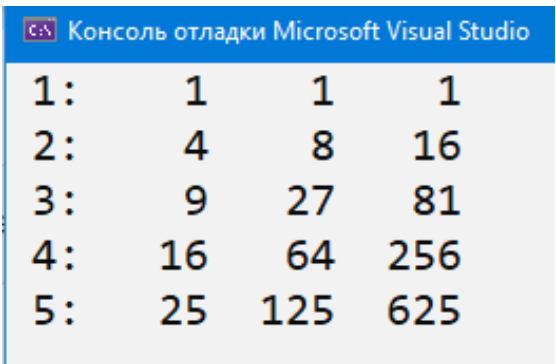
## Пример 4. реализация результата с помощью анонимного метода

```
using System;

delegate double Powers(double x);

namespace MoreAnonymAsResult
{
    internal class Program
    {
        static Powers maker(int n) //статич метод, возвр ссылку на экз делегата
        {
            Powers meth;
            meth=delegate (double x)
            {
                double s = 1;
                for (int i = 1; i<=n; i++)
                { s*=x; }
                return s;
            };
            return meth;
        }
    }
}
```

```
static void Main(string[] args)
{
    Powers sqr = maker(2);
    Powers cube = maker(3);
    for (int i = 1; i<=5; i++)
    {
        Console.WriteLine($"{i,2}:{sqr(i),5}{ cube(i),5}{ maker(4)(i),5}");
    }
}
}
```



Консоль отладки Microsoft Visual Studio

1:	1	1	1
2:	4	8	16
3:	9	27	81
4:	16	64	256
5:	25	125	625