

# C# Starter

Варианты использования методов.

# C# Starter

Автор курса



Александр Шевчук  
MCT



MCID: 9230440

# C# Starter

После урока обязательно



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

Доступ можно получить через руководство вашего учебного центра



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

# Варианты использования методов

# Методы

## Перегрузка

```
void Method()  
{  
    Console.WriteLine("Hello!");  
}
```

```
void Method(string s)  
{  
    Console.WriteLine(s);  
}
```

```
void Method(int i)  
{  
    Console.WriteLine(i);  
}
```

Под перегрузкой методов понимается, создание нескольких методов (два или более) с одинаковым именем, но различным типом и/или количеством параметров.

Наборы параметров перегруженных методов могут отличаться порядком следования, количеством, типом. Таким образом перегрузка методов необходима для того, чтобы избежать создания нескольких разноименных методов, выполняющих сходные действия.

# Методы

## Аргументы (Параметры)

### Позиционные

---

При вызове метода, аргументы передаются в соответствии с заданной позицией и их типом.

```
void Method(int a, string b)
{
    Console.WriteLine(a + b);
}
```

Вызов метода:

---

```
Method(5, "Hello");
```

### Именованные

---

При вызове метода, аргументы передаются в соответствии с их именем и типом, при этом позиция не имеет значения, но количество аргументов должно совпадать.

```
void Method(int a, string b)
{
    Console.WriteLine(a + b);
}
```

Вызов метода:

---

```
Method(b: "Hello", a: 5);
```

### Опциональные

---

При вызове метода, аргументам можно передавать значения в соответствии с их позицией, типом и именем, при этом, те аргументы, которым не передано какое-то значение, уже имеют значение по умолчанию.

```
void Method(int a=1, string b="Ok")
{
    Console.WriteLine(a + b);
}
```

Вызов метода:

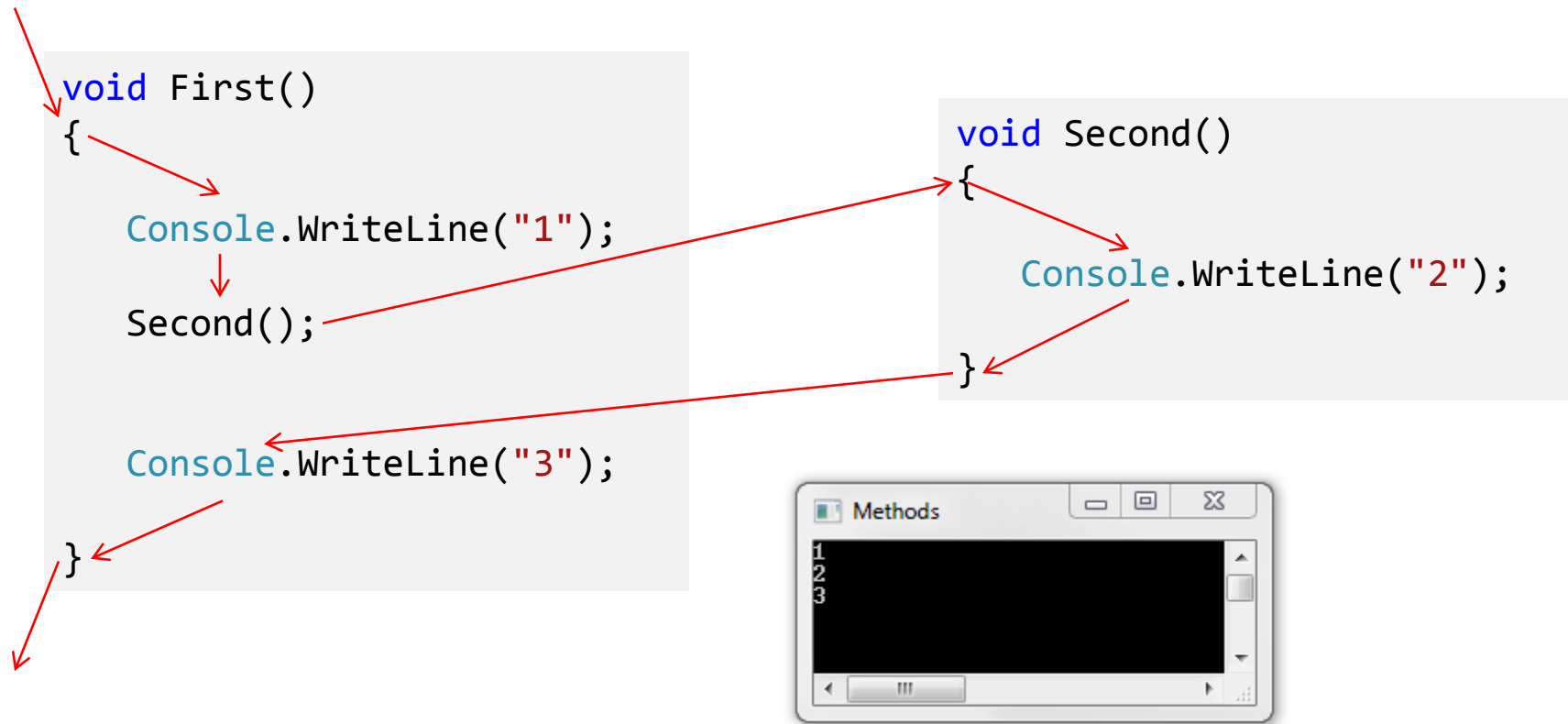
---

```
Method (b : "Hello1");
Method(5);
Method();
```

# Методы

## Вызов методом другого метода

Методы могут в своих телах вызывать другие методы. После того, как вызванный метод отработает, выполнение вызывающего метода продолжается.

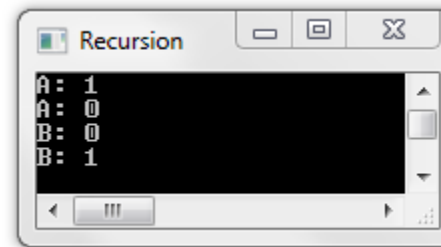


# Методы

## Рекурсия

**Рекурсия** – вызов методом самого себя, непосредственно (простая рекурсия) или через другой метод(сложная рекурсия)

```
void Main()
{
    Recursion(2);
}
```



```
void Recursion(int counter) // counter = 2
{
    counter--; // counter = 1

    Console.WriteLine("A: {0}", counter);

    if (counter != 0) // 1 != 0
        Recursion(counter);

    Console.WriteLine("B: {0}", counter);
}
```

### Копия метода

```
void Recursion(int counter) // counter = 1
{
    counter--; // counter = 0

    Console.WriteLine("A: {0}", counter);

    if (counter != 0) // 0 != 0
        Recursion(counter);

    Console.WriteLine("B: {0}", counter);
}
```



# Смотрите наши уроки в видео формате

ITVDN.com

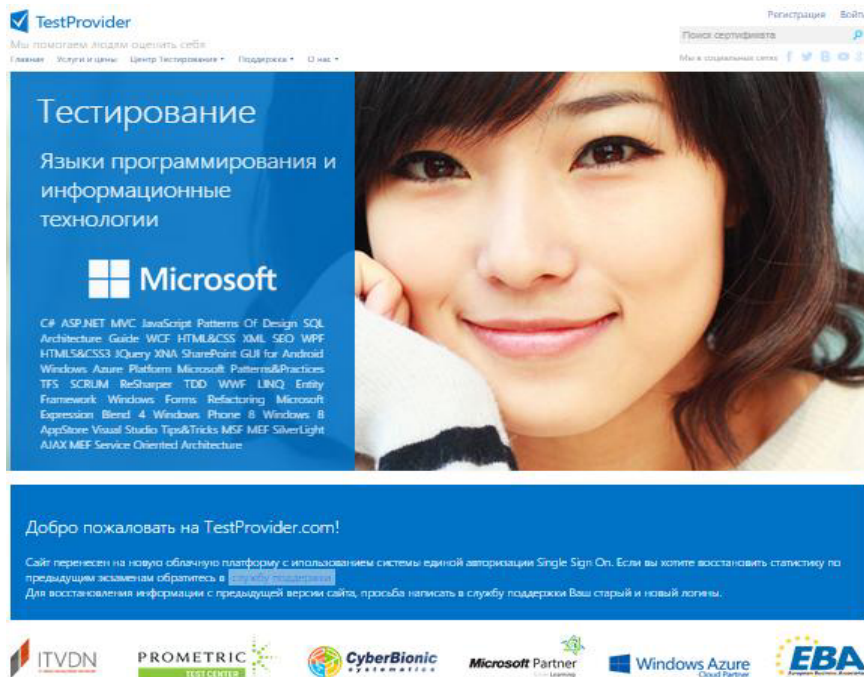
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://ITVDN.com) для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics



# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](http://TestProvider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# C# Starter

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

