

Методы

[**модификаторы**] тип_возвращаемого_значения название_метода ([параметры])

```
{  
    // тело метода  
}
```

```
static void Main(string[] args)  
{  
    Console.WriteLine("привет!");  
}
```

```
int GetFactorial()  
{  
    var x = 14;  
    return x;  
}
```

```
string Hello()  
{  
    return "Hello";  
}
```

Метод состоит из двух частей:

- Спецификация метода.
- Тело метода.

Перегрузка методов (method overloading)

Перегрузку методов следует использовать только для создания различных методов, которые имеют одну семантику.

Пример.

Вызов метода

```
static void Main(string[] args)
{
    var isLock Report = LockReport(string reportName, string userName)
    Console.WriteLine(isLockReport);
}
```

```
bool LockReport(string reportName, string userName)
{
    bool success = false;
    // Perform some processing here.
    return success;
}
```

Массив параметров

- Если существует перегрузка метода, соответствующая указанному типу и количеству параметров, она будет вызываться предпочтительнее, чем версия метода, принимающего массив параметров.
- Метод может принимать в качестве параметра только один массив параметров.
- Массив параметров можно использовать в сочетании с другими параметрами, при этом он должен быть последним параметром в списке параметров, указанных в методе.

```
int Add(params int[] data)
{
    int sum = 0;
    for (int i = 0; i < data.Length; i++)
    {
        sum += data[i];
    }
    return sum;
}
```

Необязательные параметры

Позволяют при определении метода предоставить для параметров в списке параметров значения по умолчанию

- Чтобы было легче включить COM библиотеки и компоненты в C# приложение
- когда не представляется возможным использовать перегрузку, поскольку типы параметров не меняются так, чтобы компилятор проводил различие между реализациями

```
voidMyMethod(int intData, float floatData, int moreIntData = 99)
```

```
{
```

```
...
```

```
}
```

Именованные аргументы

```
void MyMethod(int first, double second, string third)
{
}

...

// Method call using named arguments.
MyMethod(third: "Hello", first: 1234, second: 12.12);
```

Выходные параметры

Для использования выходных параметров необходимо указать переменную для соответствующего аргумента при вызове метода и добавить префикс out к аргументу.

```
void MyMethod(int first, double second, out int data)
{
...
data = 99;
}
```