

Параметры методов

№ 37 **Курс:** Процедурное программирование на языке C#
урока:

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

На этом уроке будет более детально рассмотрена тема параметров методов и того, какие виды параметров бывают. Будут рассмотрены опциональные параметры, именованные параметры, параметры, передаваемые по ссылке и другие особенности.

Изучив материал данного занятия, учащийся сможет:

- Понимать, какие бывают параметры методов в C#.
- Уметь применять на практике методы с параметрами, передаваемыми по ссылке.
- Использовать ключевое слово `params` для передачи в метод переменного числа параметров.

Содержание урока

1. Позиционные и именованные параметры
2. Опциональные параметры
3. Ref-In-Out параметры
4. Params параметры

Резюме

- **Позиционные параметры (аргументы)**, это самая простая и традиционная разновидность фактических параметров. Используя такие параметры при вызове метода фактические параметры передаются в том же порядке, в каком указаны формальные параметры. Количество передаваемых в метод фактических параметров, должно совпадать с количеством, типом и позицией формальных параметров.
- **Именованные параметры (аргументы)** – это фактические параметры, которые передаются в метод в соответствии с именем и типом формальных параметров. Позиция фактических параметров может не совпадать с позицией формальных, но количество фактических параметров должно совпадать с количеством формальных параметров. Для указания
- Именованные параметры освобождают разработчика от необходимости запоминать или уточнять правильный порядок параметров при вызове методов. Фактический параметр можно указать, используя имя формального параметра.
`Method(sellerName: "Gift Shop", 31, productName: "Red Mug");`
`sellerName` и `productName` – имена формальных параметров метода `Method`.

- Стив Макконнелл, советует использовать именованные параметры.
- Использование именованных параметров может быть полезно при работе с методами, у которых длинный список параметров одинакового типа, потому что в этом случае вероятность неправильного сопоставления параметров более высока, а компилятор эту ошибку определить не может.
- При создании метода, можно указать, что параметры являются обязательными или необязательными. При каждом вызове необходимо указывать фактические параметры для всех обязательных формальных параметров, но можно опустить фактические параметры для необязательных формальных параметров.
- **Опциональные параметры (аргументы).** При создании метода можно для параметров указать значения по умолчанию, такие параметры называются опциональными. В этом случае при вызове метода можно как передать фактический параметр, так и не передавать его. Если фактический параметр в метод не передается - используется значение по умолчанию.

```
public void ExampleMethod(int required, string optionalstr = "default string", int optionalint = 10)
```
- При работе с методом, в котором имеются опциональные параметры, для передачи фактических параметров можно использовать как технику именования параметров, так и передать фактические параметры, без их именования. При этом, можно передать только некоторые параметры, а часть параметров упустить. Можно, вообще не передавать никаких фактических параметров.
- **ref** - это сокращение от слова reference (ссылка). Ссылка — это адрес переменной в памяти. Переменная помеченная ключевым словом **ref**, будет хранить в себе адрес переменной, но не значение переменной.
- Если параметр метода помечен ключевым словом **ref**, то в качестве фактического параметра этот метод будет принимать не значение переменной, а адрес памяти, по которому размещается эта переменная.
- Для работы с методом, формальные параметры которого помечены ключевым словом **ref**, при вызове такого метода нужно **явно** использовать ключевое слово **ref**, перед фактическим параметром в методе.
- Переменную, передаваемую в качестве фактического **ref** параметра, требуется предварительно инициализировать некоторым значением.
- Ключевое слово **out**, схоже по функциональности с ключевым словом **ref**, за исключением того, что переменную, которая будет использоваться в качестве **out** параметра предварительно не требуется инициализировать. Также объявление переменной, передаваемой как **out** параметр метода, можно встроить прямо в вызов метода.

- Для организации работы с out параметром, фактические и формальные параметры должны быть помечены ключевым словом out.
- В теле метода требуется обязательно присваивать некоторое значение формальным out параметрам.
- Ключевое слово **in** аналогично ключевым словам ref и out, за исключением того, что формальные in параметры не могут быть изменены в теле метода.
- Ключевые слова ref и out не считаются частью сигнатуры метода.
- Все три ключевых слова: ref, in и out, - делают так, что параметры, которыми они помечены, хранят в себе ссылки, а не значения.
- Ключевое слово **ref** – это универсальное ключевое слово, которое, не накладывает на работу с ссылочными параметрами никаких ограничений.
- Ключевое слово **in** – позволяет передавать параметр по ссылке, также, как и ref, за исключением того, что in-параметры невозможно изменять в теле метода.
- Ключевое слово **out** – позволяет передавать параметр по ссылке, как и ref, за исключением того, что out-параметры обязательно требуется инициализировать в теле метода. Эта обязательная инициализация позволяет воспринимать out-параметры, как своеобразные возвращаемые значения из метода.
- Формальные параметры метода, помеченные ключевым словом **params**, позволяют при вызове методе принимать переменное число фактических параметров.
- В списке формальных параметров может быть только один параметр, помеченный ключевым словом params, и этот params-параметр должен располагаться последним в списке формальных параметров. Иначе будет ошибка.
- Параметр, помеченный ключевым словом params обязательно должен быть одномерным массивом.

Закрепление материала

- Какие виды параметров бывают?
- Чем отличаются позиционные, именованные и опциональные параметры?
- Чем отличаются ключевые слова ref, in и out?
- Что означает ключевое слово params?

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Напишите программу, в которой создайте три метода.

Первый метод - принимает три целых числа и возвращает сумму этих чисел.

Второй метод принимает три целочисленных ref параметра, увеличивает каждый из полученных параметров на единицу, а затем возвращает сумму этих увеличенных значений.

Третий метод в качестве параметров принимает три целых числа и булево значение, и возвращает целочисленное значение. Если булевый параметр содержит значение true – третий метод вызовет метод №1 и передаст ему значения своих числовых параметров, иначе - если булевый параметр содержит значение false - третий метод в своем теле вызовет метод №2 и передаст ему свои числовые параметры.

Выведите на экран результат выполнения метода №3 и значение его параметров.

Рекомендуемые ресурсы

<https://docs.microsoft.com/ru-ru/dotnet/csharp/methods>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/ref>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/out-parameter-modifier>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/in-parameter-modifier>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/params>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/named-and-optional-arguments>