Название говорит само за себя. Если надо ввести аргументы, их можно ввести вместе с названием этих самых аргументов.

Еще одним средством, связанным с передачей аргументов методу, является ***именованный***

***аргумент****.* Именованные аргументы были внедрены в версии C# 4.0. Как вам

должно быть уже известно, при передаче аргументов методу порядок их следования,

как правило, должен совпадать с тем порядком, в котором параметры определены в

самом методе. Иными словами, значение аргумента присваивается параметру по его

позиции в списке аргументов. Данное ограничение призваны преодолеть именованные

аргументы. Именованный аргумент позволяет указать имя того параметра, которому

присваивается его значение. И в этом случае порядок следования аргументов уже не

имеет никакого значения.

static bool IsFactor(int val, int divisor)

{

if ((val % divisor) == 0) return true;

return false;

}

// Вызов с использованием позиционных аргументов.

if(IsFactor(10, 2))

Console.WriteLine("2 - множитель 10.");

// Вызов с использованием именованных аргументов.

if(IsFactor(val; 10, divisor: 2))

Console.WriteLine("2 - множитель 10.");

// Для именованного аргумента порядок указания не имеет значения.

if(IsFactor(divisor: 2, val: 10))

Console.WriteLine("2 - множитель 10.");

// Применить как позиционный, так и именованный аргумент.

if(IsFactor(10, divisor: 2))

Console.WriteLine("2 - множитель 10.");

Независимость от порядка следования является главным преимуществом именованных

аргументов. Это означает, что запоминать (или даже знать) порядок следования

параметров в вызываемом методе совсем не обязательно. Для работы с СОМ-

интерфейсами это может быть очень удобно. И во-вторых, позиционные аргументы

можно указывать вместе с именованными в одном и том же вызове.

if(IsFactor(10, divisor: 2))

Синтаксис именованных аргументов более многословен, чем у обычных позиционных

аргументов, и поэтому для вызова методов чаще всего применяются позиционные

аргументы. Но в тех случаях, когда это уместно, именованные аргументы могут быть

использованы довольно эффективно.