Материалы для подготовки к семинару 4, модуль 1

# Передача параметров в методы по ссылке

Параметры, передаваемые в метод по ссылке, отмечаются модификаторами **ref** и **out**. С модификатором **ref** в метод передаётся уже проинициализированный параметр. Параметр с модификатором **out** может быть не проинициализирован до передачи в метод, но обязательно получает значение в коде метода.

# Примеры

## Пример 1. Передача параметров по ссылке.

**using System;**

**public class Program {**

**// Метод увеличивает высоту и ширину прямоугольника на 1**

**// Прямоугольник задан координатами верхнего левого угла**

**// высотой и шириной**

**public static bool RectangleInflate(int x, int y, ref int height, ref int width) {**

**// проверяем корректность переданных значений**

**if (height <= 0 || width <= 0) return false;**

**height++;**

**width++;**

**return true;**

**}**

**public static void Main() {**

**// тест метода для корректных данных**

**int x = 0, y = 0;**

**int height = 5, width = 7;**

**Console.WriteLine("({0},{1}) h={2} w={3}", x, y, height, width);**

**if (!RectangleInflate(x, y, ref height, ref width)) {**

**Console.WriteLine("Неверные параметры прямоугольника!");**

**return;**

**}**

**Console.WriteLine("Преобразование успешно произведено:");**

**Console.WriteLine("({0},{1}) h={2} w={3}", x, y, height, width);**

**int xWrong = 1, yWrong = 1;**

**Console.WriteLine();**

**int heightWrong = -5, widthWrong = 7;**

**// тест метода для неверных данных**

**Console.WriteLine("({0},{1}) h={2} w={3}", xWrong, yWrong, heightWrong, widthWrong);**

**if (!RectangleInflate(x, y, ref heightWrong, ref widthWrong)) {**

**Console.WriteLine("Неверные параметры прямоугольника!");**

**return;**

**}**

**Console.WriteLine("Преобразование успешно произведено:");**

**Console.WriteLine("({0},{1}) h={2} w={3}", xWrong, yWrong, heightWrong, widthWrong);**

**}**

**}**

## Задания к примеру 1

1. Тесты, предложенные в методе **Main()** – пример дублирующего кода. Напишите метод, в который можно выделить этот код и используйте его вызовы в **Main()**.
2. Напишите метод **RectangleDeflate(),** имеющий те же параметры, что и метод **RectangleInflate()**. Метод, возвращает **true**, если переданный в качестве параметра прямоугольник может быть уменьшен по обоим измерениям на **1** и уменьшает измерения на **1** и **false** в противном случае. Полагать, что длина стороны не может быть меньше **1**.
3. Напишите метод **RectangleNormalize()**, имеющий те же параметры, что и метод **RectangleInflate()**. Метод, возвращает **true**, если переданный в качестве параметров прямоугольник может быть преобразован в квадрат путём уменьшения длины одной из сторон и уменьшает одно из сторон и **false** в противном случае. Самостоятельно рассмотреть варианты некорректных параметров.

# Битовые операции

Поразрядное дополнение **~** (унарная операция) [<https://msdn.microsoft.com/ru-ru/library/d2bd4x66.aspx>]

Поразрядное логическое И **&** (бинарная операция) [<https://msdn.microsoft.com/ru-ru/library/sbf85k1c.aspx>]

Поразрядное логическое ИЛИ **I** (бинарная операция) [<https://msdn.microsoft.com/ru-ru/library/kxszd0kx.aspx>]

Поразрядные сдвиги влево (**<<**) и вправо (**>>**) (бинарные операции) [<https://msdn.microsoft.com/ru-ru/library/a1sway8w.aspx>], [<https://msdn.microsoft.com/ru-ru/library/xt18et0d.aspx>]

# Примеры

## Пример 1. Представление целых чисел

Вывести на экран таблицу чисел от 0 до 15 в десятичной и шестнадцатеричной системах счисления.

**using System;**

**class Program {**

**static void Main() {**

**Console.WriteLine("Dec\tHex");**

**Console.WriteLine("=================");**

**for (int i = 0; i < 16; i++) {**

**Console.WriteLine("{0}\t{0:X4}", i);**

**}**

**}**

**}**

## Пример 2. Поразрядное дополнение

**using System;**

**class Program {**

**static void Main() {**

**// значение целочисленной переменной**

**//можно задать шестнадцатеричным числом**

**int testValue = 0xffff;**

**// Вывод в десятичном представлении**

**Console.WriteLine("{0} || {1}", testValue, ~testValue);**

**// Вывод в шестнадцатеричном представлении**

**Console.WriteLine("{0:x8} || {1:x8}", testValue,~testValue);**

**}**

**}**

## Пример 3. Операция побитового сдвига

**// Циклы и поразрядные сдвиги.**

**// Ввести положительное целое число и вывести**

**// его битовое представление.**

**using System;**

**class Program {**

**static void Main() {**

**uint numb, // Введенное число**

**bit, // очередной разряд**

**temp; // рабочая переменная**

**do { // цикл для повторения решений задачи**

**// Конструкция для ввода значения:**

**do Console.Write("Введите положительное целое: ");**

**while (!uint.TryParse(Console.ReadLine(), out numb));**

**int expo = 0; // величина поразрядного сдвига (степень 2)**

**while (numb >= 1 << ++expo); // цикл с предусловием**

**temp = numb;**

**do { // цикл с постусловием**

**expo--;**

**bit = temp >> expo;**

**Console.Write(bit + " ");**

**temp -= bit << expo;**

**}**

**while (expo > 0);**

**Console.WriteLine("Для выхода нажмите клавишу ESC");**

**} while (Console.ReadKey(true).Key != ConsoleKey.Escape);**

**} //Конец определения метода Main()**

## **} //Конец объявления класса Program**

0000 0000 0000 000**1**

expo = 1

0000 0000 0000 00**1**0

expo = 2

expo = numb

0000 000**1** 0000 0000

numb

## Задание к примеру 1

Выделить метод, позволяющий для целого числа вывести в консольное окно его битовое представление. Код в методе **Main()** заменить вызовом этого метода.

# Задачи для подготовки к семинару 4

## Задача 1

Написать метод Props(), получающий в качестве параметров два целых числа N и K, метод имеет два out параметра GCD и LCM – в которые сохраняются вычисленные значения наибольшего общего делителя и наименьшего общего кратного. Метод возвращает значение типа bool: false, если НОД или НОК не может быть вычислено и true в противном случае.

В основной программе получить от пользователя два целых неотрицательных числа и, использовав метод Props(), вычислить их НОД и НОК.

## Задача 3

Используя операции побитового сдвига, написать методы «быстрого возведения» двойки в заданную степень целую степень.

## Задача 2

Используя операции побитового сдвига, написать метод «быстрого умножения» числа на 2.