



# Программирование на C#

## Семинар №7

Модуль №1

Тема:

Методы. Статические переменные и статические  
методы

Размещение статических методов в отдельном статическом классе.

Простые статические методы, вызовы.



# Задания преподавателя к семинару

- Во всех задачах статические методы разместите в статическом классе **Methods**. Исходный код **Methods** сохраните в отдельном файле проекта консольного приложения
- Класс может иметь и другое, более подходящее, название



# Полезные материалы к семинару

- static (Справочник по C#) <https://msdn.microsoft.com/ru-ru/library/98f28cdx.aspx>
- Создавайте проекты и решения Visual Studio, работайте с ними и удаляйте их(<http://learn.microsoft.com/ru-ru/visualstudio/ide/creating-solutions-and-projects?view=vs-2022#add-files-to-a-solution> )

# Добавление Файлов с Исходным Кодом на C# в Visual Studio



На практике размещать все методы и классы в одном файле оказывается неудобно, а в случае изменений приходится перекомпилировать весь код.

- Чтобы создать отдельный класс в Visual Studio нужно выполнить несколько шагов:
  - 1) Открыть обозреватель решений (Solution Explorer, *Ctrl+Alt+L*);
  - 2) Нажать правой кнопкой мыши по имени проекта (не решения!);
  - 3) Выбрать пункт «Добавить» (Add) в выпадающем меню;
  - 4) Выбрать «Класс...» (Class...) во вложенном меню;
  - 5) Ввести в появившемся поле имя класса и нажать кнопку «Добавить» (Add).

# Демо 01. Проект с Двумя Файлами Кода

```
// Файл Separate.cs
public class Separate {
    public static double Average(double x, double y) {
        return (x + y) / 2;
    }
}
```

Без public  
возникнет  
ошибка  
доступа.

```
using System;
// Файл Program.cs
class Program {
    static void Main() {
        double average = Separate.Average(3, 6);
        Console.WriteLine($"Average = {average}");
    } //end of Main
} // end of Program
```

# Demo 02. Статические Методы.



[модификаторы] тип\_возврата ИмяМетода([параметры])

```
{  
    // Тело метода  
}
```

Что произойдет, если заменить  
r = Sum1(5, 10) на Sum2(5, 10)?



```
static void Main(string[] args)  
{  
    int p1 = 10, p2 = 30;  
    Sum1(p1, p2);  
    int r;  
    r = Sum2(5, 10);  
    Console.WriteLine(r);  
    Console.ReadKey();  
}
```

```
public static void Sum1(int a, int b)  
{  
    int c;  
    c = a + b;  
    Console.WriteLine(c);  
}
```

```
public static int Sum2(int a, int b)  
{  
    int c;  
    c = a + b;  
    return c;  
}
```

# Demo 03. Возврат Значения из Метода



```
public static double sqrCircle(int r)
{
    return Math.PI * r * r;
}
```

```
public static double longCircle(int r)
{
    return 2 * Math.PI * r;
}
```

```
public static void writeResult(int r)
{
    Console.WriteLine("Вычислить площадь или длину? s/l:");
    string s = Console.ReadLine();
    s = s.ToLower();
    if (s == "s")
    {
        Console.WriteLine("Площадь круга равна {0:###.###}", sqrCircle(r));
        return;
    }
    else
    {
        Console.WriteLine("Длина окружности равна {0:###.###}", longCircle(r));
        return;
    }
}
```

```
static void Main(string[] args)
{
    Console.WriteLine("Введите радиус: ");
    string radius = Console.ReadLine();
    int rad = int.Parse(radius);

    writeResult(rad);

    Console.ReadLine();
}
```

# ToDo 01. Вывод таблицы целых чисел



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

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

```
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);
```

```
        }
```

```
    }
```



# Self 01: Рисуем звёздочками



Разработайте метод `Triangle()`, который позволяет рисовать на экране звёздочками треугольник. Треугольник высотой в  $N$  строк,  $N$  – целочисленный параметр метода. Для  $N \leq 0$  метод завершается, ничего не выводя на экран.

`N = 5`

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

`N = 7`

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

Подумайте заранее, что возвращает метод? Какие у него параметры и каких типов?

# Self 02: Ёлочка



Напишите метод `Ornament(int N, int M)`, который вызывает метод `Triangle()` и рисует ёлочку с  $M > 0$  секциями. Для  $M \leq 0$  метод `Ornament()` ничего на экран не выводит и завершается

```
N = 5 M = 3
```

```
*
**
***
****
*****
*
**
***
****
*****
*
**
***
****
*****
```

- Подумайте заранее, что возвращает метод?
- Какие у него параметры и каких типов?

## Self 03: «Пила»



Усовершенствуйте методы Ornament() и Triangle() так, чтобы они отрисовывали «пилу»

`N = 4, M = 3`

`* * *`

`** ** **`

`*** ***`

`**** ****`

Какой метод будем  
модифицировать и как?

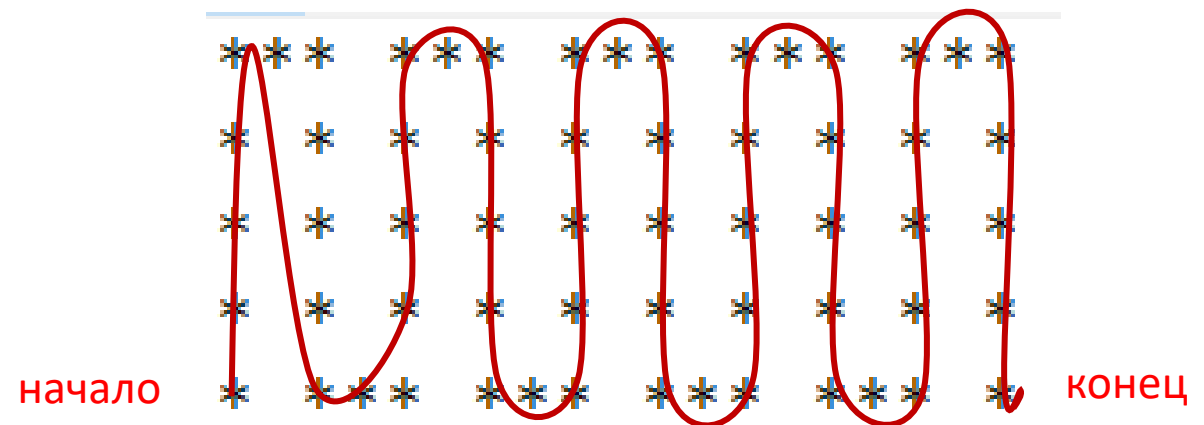
# Self 04\*: Визуализируем «Питона»



Напишите метод `Python(int n)`, который выводит на экран «змейку» из звёздочек.

Например, для  $n = 5$  появляется змейка с пятью изломами сверху и высотой в пять звёздочек.

\* Поищите самостоятельно, каким образом организовать задержку хода программы при выводе, модифицируйте код так, чтобы последовательность вывода звёздочек на экран была заметна (как в анимации)





## Self 05\*: Сокращаем дробь

Написать метод, вычисляющий НОК двух целых чисел (любого знака).

Используя метод, вывести результат сокращения правильной дроби. Числитель и знаменатель – два целых числа с любым знаком, полученные от пользователя.

\* Дополнительно подумайте о других методах, которые требуются для решения задачи. Реализуйте их.

Пример:

Введите числитель: 55

Введите знаменатель: 33

Результат: 5/3



## Self 06\*: Рекурсивные вычисления

Написать рекурсивный метод вычисления чисел Стирлинга второго рода, заданных рекуррентной формулой:

$$S(n, n) = 1, n \geq 0$$

$$S(n, 0) = 0, n > 0$$

$$S(0, k) = 0, k > 0$$

$$S(n, k) = S(n - 1, k - 1) + k \cdot S(n - 1, k), 0 < k < n$$

Написать рекурсивный метод вычисления чисел Стирлинга первого рода, заданных рекуррентной формулой:

$$S(0, 0) = 1$$

$$S(n, 0) = 0, n > 0$$

$$S(0, k) = 0, k > 0$$

для чисел со знаком  $S(n, k) = S(n - 1, k - 1) - (n - 1)S(n - 1, k), 0 < k < n$

для чисел без знака  $S(n, k) = S(n - 1, k - 1) + (n - 1)S(n - 1, k), 0 < k < n$