

## ЛАБОРАТОРНАЯ РАБОТА № 4

### Разработка методов, использование в программах встроенных функций

**Цель работы:** научиться разрабатывать и применять алгоритмы и программы с применением пользовательских методов, функций.

**Оборудование:** персональный компьютер, практикум, тетради для лабораторных работ.

**Программное обеспечение:** MS Office, MSVisualStudio

**Правила по технике безопасности:** общие (приложение).

**Литература:**

1. Г. Шилдт, Полный справочник по C#.

**Время выполнения:** 2 часа.

**ВОПРОСЫ ВХОДНОГО КОНТРОЛЯ:**

1. Дайте определение строки.
2. Перечислите функции для работы со строками.

### ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Функция являет собой небольшую подпрограмму. Если просто программа - это решение какой-то прикладной задачи, то функция – это тоже решение, только уже в рамках программы и, соответственно, она выполняет задачу «попроще». Функции позволяют уменьшить размер программы за счет того, что не нужно повторно писать какой-то фрагмент кода - мы просто вызываем сколько угодно и где нужно объявленную функцию.

Функции в Си-шарп также называют методами. Между этими двумя понятиями разница небольшая, и тут мы будем использовать термин функция.

До этого, мы весь код писали в функции main. Функция main является главной функцией приложения и точкой входа программы. Любая функция в Си-шарп может быть объявлена только в рамках класса, так как C# - полностью объектно-ориентированный язык программирования (ООП). Объявление пользовательской функции внутри другой функции (например main) недопустимо. Объявление функции имеет следующую структуру:

```
[модификатор доступа] [тип возвращаемого значения] [имя функции]
([аргументы])
{
    // тело функции
}
```

Модификатор доступа (области видимости) может быть public, private, protected, internal. Для чего это нужно будем говорить в отдельном уроке, а пока будем везде использовать public.

Между модификатором и типом может стоять ключевое слово static, что означает, что функция будет статичной. Подробно говорить о статичных функциях и переменных мы будем в отдельном уроке. Скажу только, что из статичной функции можно вызывать другие функции, если они тоже статичные. Главная функция main – всегда static, поэтому все функции в этом уроке мы будем объявлять также статичными.

Функция может возвращать значение или не возвращать. Если функция, например, возвращает целое число, то нужно указать тип int. Если функция не возвращает никакого значения, то для этого используется ключевое слово void. Функции, которые не возвращают значение, еще называют процедурами.

Называть функции стоит так, чтобы имя отображало суть функции. Используйте глаголы или словосочетания с глаголами. Примеры: GetAge(), Sort(), SetVisibility().

Аргументы – это те данные, которые необходимы для выполнения функции. Аргументы записываются в формате **[тип] [идентификатор]**. Если аргументов несколько, они отделяются запятой. Аргументы могут отсутствовать.

Первая строка функции, где указываются тип, имя, аргументы и т.д. называется заголовком функции.

Итак, хватит теории, переходим к практике.

### **Пример функции, которая не возвращает значение**

*Напишем простую функцию, которая будет заменять в массиве строк указанное имя на другое. Данная функция будет принимать три аргумента: массив строк, имя, которое необходимо заменить и новое имя. Так как функция не будет возвращать значение, указываем тип void перед именем функции.*

```
class Program
{
    public static void ReplaceName(string[] names, string name, string
newName)
    {
        for (int i=0; i<names.Length; i++)
        {
            if (names[i] == name)
                names[i] = newName;
        }
    }
    static void Main(string[] args)
    {
        string[] names = { "Sergey", "Maxim", "Andrey", "Oleg", "Andrey",
"Ivan", "Sergey" };
        Console.WriteLine("Исходный массив");
        foreach (string s in names)
            Console.Write(s+" ");
        Console.WriteLine();
        Console.WriteLine("Введите Имя которое хотите заменить");
        string name = Console.ReadLine();
```

```

        Console.WriteLine("Введите Имя которое хотите вставить");
        string newName = Console.ReadLine();
        ReplaceName(names, name, newName);
        Console.WriteLine();
        Console.WriteLine("Полученный массив");
        foreach (string s in names)
            Console.Write(s + " ");
        Console.ReadKey();
    }
}

```

## Пример функции, которая возвращает значения

Напишем функцию, которая будет находить максимальное число в массиве. Аргумент у этой функции будет один – массив целых чисел. Тип возвращаемого значения – целое число *int*.

```

public static int GetMax(int[] array)
{
    int max = array[0];
    for (int i = 1; i < array.Length; i++)
    {
        if (array[i] > max)
            max = array[i];
    }
    return max;
}

static void Main(string[] args)
{
    int[] numbers = { 3, 32, 16, 27, 55, 43, 2, 34 };
    int max;
    max = GetMax(numbers);
    Console.WriteLine("Максимальный элемент массива: " + max);
    Console.ReadKey();
}
}

```

формальные параметры указываются при объявлении функции. Фактические - передаются при вызове.

## Рекурсия

```

class Program
{
    private static int factorial(int i)
    {
        int result;
        if (i == 0 || i == 1)
            return 1;
        result = factorial(i - 1) * i;
        return result;
    }

    private static int noRecfactorial(int n)
    {
        int result = 1;
        if (n == 0 || i == 1)
            return 1;
        for (int i = 1; i <= n; i++)
            result *= i;
        return result;
    }

    static void Main(string[] args)
    {

```

```

        Console.WriteLine("Введите число: ");
        int i = int.Parse(Console.ReadLine());
        Console.WriteLine("{0}! Через рекурсию = {1}", i, factorial(i));
        Console.WriteLine("{0}! Без рекурсии = {1}", i,
noRecfactorial(i));
        Console.ReadKey();
    }
}

```

### Модификатор params

Модификатор **params** может быть указан для последнего параметра метода, что позволяет передать в метод любое число параметров одного и того же типа. Тип параметра при этом должен быть объявлен как массив.

```

class Program
{
    public static int Sum(params int[] values)
    {
        int sum = 0;
        for (int i = 0; i < values.Length; i++)
        {
            sum += values[i];
        }
        return sum;
    }

    static void Main(string[] args)
    {
        Console.WriteLine(Sum(1, 2, 3));
        Console.WriteLine(Sum(1, 2, 3, 5, 6));
        Console.WriteLine(Sum(1, 2, 3, 7));
        Console.WriteLine(Sum(1, 2, 3, 9, 10, 15));
        Console.WriteLine(Sum(new int[] { 1, 2, 3, 4 }));
    }
}

```

## ПРАКТИЧЕСКАЯ ЧАСТЬ

### Варианты индивидуальных заданий

**Общее требование. функции Main должна содержать максимум 3 строчки: вызов функции Task1() Task2() и задержку консоли**

1	1)Написать функцию Min для нахождения минимального с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров. 2)Написать метод SpecSum для нахождения суммы четных минус сумму нечетных аргументов с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров
2	1)Написать метод Average для нахождения среднего арифметического с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров 2)Напишите рекурсивную функцию расчета суммы всех двузначных чисел.

3	<p>1)Даны действительные числа <math>x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}</math>. Найти периметр десятиугольника, вершины которого имеют соответственно координаты <math>(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})</math>. (Определить метод вычисления расстояния между двумя точками, заданными своими координатами.</p> <p>2)Напишите функцию расчета суммы цифр любого произвольно вводимого числа, в качестве параметра.</p>
4	<p>1)Написать собственное представление функции <code>double pow(double a, int b)</code> для нахождения числа в степени. <math>b &gt; 0</math></p> <p>2)Напишите функцию умножения двух чисел, используя только операцию сложения.</p>
5	<p>1)Написать метод <code>Min</code> для нахождения минимального с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров</p> <p>2)Описать метод <code>int SumDigit(N)</code>, находящую сумму цифр целого числа <math>N</math> (<math>N</math> - входной). Запустить цикл, вызывающий данную функцию 5 раз, используя этот метод, найти суммы цифр пяти данных чисел.</p>
6	<p>1)Написать метод <code>Sum</code> для нахождения суммы аргументов с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров.</p> <p>2)Пусть процедура <code>double Socr(a,b)</code> от целых параметров (<math>b</math>) приводит дробь к несократимому виду</p>
7	<p>1)Написать метод <code>Average</code> для нахождения среднего арифметического с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров.</p> <p>2)Описать метод <code>Fact(N)</code> целого типа, вычисляющую значение факториала <math>N! = 1 \cdot 2 \cdot \dots \cdot N</math> (<math>N &gt; 0</math> - параметр целого типа)</p>
8	<p>1)Написать метод <code>double AreaCircle(r)</code> вычисляющий площадь круга. Показать его работу</p> <p>2)Написать функцию определения, является ли заданное натуральное число простым.</p>
9	<p>1)Написать метод <code>SpecSum</code> для нахождения суммы чисел делящихся на 3 минус сумму чисел делящихся на 2 для неограниченного количеством параметров. Показать его работу несколько раз с разным количеством параметров.</p> <p>2)Даны длины <math>a, b</math> и <math>c</math> сторон некоторого треугольника. Найти медианы треугольника, сторонами которого являются медианы исходного треугольника.</p> $m_a = \frac{1}{2} \sqrt{2b^2 + 2c^2 - a^2}$ <p>Длина медианы, проведенной к стороне <math>a</math>, равна</p>
10	<p>1)Даны вещественные положительные числа <math>a, b, c</math>. Написать функцию <code>bool CheckTriangle(double a, double b, double c)</code>, которая определяет, является ли треугольник прямоугольным</p> <p>2)Дано натуральное число <math>n</math>. Выяснить, является ли оно полным квадратом. Написать функцию, позволяющую распознавать полные квадраты. Целое неотрицательное число <math>n</math> называется <i>полным квадратом</i>, если найдется целое число <math>m</math> такое, что <math>n = m^2</math>.</p>

<b>11</b>	<p>1) Вывести таблицу умножения для чисел от 2х до 9. На числа от 0 до 10.</p> <p>2) Написать функцию IntegerPart(double x) возвращающий целую часть числа. Показать его работу. Round НЕ ИСПОЛЬЗОВАТЬ</p>
<b>12</b>	<p>1) Рассмотрим произвольное натуральное число и найдем сумму его цифр, затем сумму цифр полученного числа и так далее, пока не получим однозначное число. Назовем это число цифровым корнем. Требуется написать функцию, которая для заданного N находит его цифровой корень.</p> <p>2) Даны отрезки a,b,c и d. Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. Определить метод Plo(x,y,z), печатающую площадь треугольника со сторонами x,y и z, если такой треугольник существует.</p>
<b>13</b>	<p>1) Написать функцию bool CheckGoldbach(int number) Дано четное число <math>n &gt; 2</math>. Проверить для этого числа гипотезу Гольдбаха. Эта гипотеза (по сегодняшний день не опровергнутая и полностью не доказанная) заключается в том, что каждое четное n, большее двух, представляется в виде суммы двух простых чисел. Воспользоваться функцией распознавания простых чисел. Если гипотеза верна, то вывести эти числа.</p> <p>2) Написать функцию Procent(int number, int proc), которая возвращает число равное проценту proc от числа number.</p>
<b>14</b>	<p>1) Два натуральных числа называются "дружественными", если каждое из них равно сумме всех делителей другого, за исключением его самого (таковы, например, числа 220 и 284). Напечатать все пары "дружественных" чисел, не превосходящих заданного натурального числа</p> <p>2) Написать функцию, которая вычисляет сопротивление цепи, состоящей из двух резисторов. Параметрами функции являются величины сопротивлений и тип соединения (последовательное или параллельное).</p>
<b>15</b>	<p>1) Даны координаты вершин двух треугольников. Определить, какой из них имеет большую площадь.</p> <p>2) Написать функцию, которая вычисляет объем и площадь поверхности параллелепипеда.</p>
<b>16</b>	<p>1) Написать функцию Count для нахождения количество переданных аргументов с неограниченным количеством параметров. Показать его работу несколько раз с разным количеством параметров.</p> <p>2) Написать функцию, которая вычисляет объем цилиндра. Параметрами функции должны быть радиус и высота цилиндра double VolumeCylinder(int h, int r).</p>
<b>17</b>	<p>1) Написать функцию int CountNumberIn(a,b) для нахождения количества множителей b в числе a. Показать его работу. Пример CountNumberIn(12,2) ответ 2</p> <p>2) Написать функцию, которая выводит на экран строку, состоящую из звездочек. Длина строки (количество звездочек) является параметром функции.</p>

### ***III. Контрольные вопросы***

1. Что такое сигнатура метода?
2. Назовите уровни доступа ?
3. Модификаторы метод?

### ***Литература***

1. Полный справочник по С#. Г. Шилдт. Издательский дом «Вильямс», 2004.
2. С# в подлиннике. Наиболее полное руководство. Х.Дейтел.
3. С# в задачах и примерах. Культин. Н.Б.
4. С# учебный курс. Г.Шилдт. СПб.: Питер, 2002.
5. С# программирование на языке высокого уровня Павловская Т.А. СПб.: БХВ-Петербург.