**Лабораторная работа № 3**

**Тема:** Определение и вызов функций. Передача параметров и возврат результатов.

**Цель работы:** изучить правила определения и вызова функций в языке C#, получить практические навыки разработки спецификаций функций по обработке простых типов данных и функций по обработке структурированных типов данных, и создания программ, использующих функции программиста.

*Указания:*

Требуется разработать 2 проекта, каждая программа − консольное приложение на языке C#, в среде программирования Visual Studio, использующая дополнительные функции.

**Как начать разработку функции?** Разработку функции следует начинать с её заголовка. Определитесь с задачей, которую должна решать функция. Каждая функция должна быть направлена на решение только одной локальной проблемы. Придумайте название для подпрограммы. Название должно отражать основное назначение функции. Далее нужно тщательно продумать, что подается на вход функции, какие исходные данные ей нужно передать, чтобы функция смогла выполнить поставленную задачу, и каким образом будут передаваться результаты работы функции. Для обмена информацией между вызываемой и вызывающей функцией используется механизм параметров.

**Как функция получает исходные данные?** Функции должны получать исходные данные через параметры. Крайне нежелательно использование глобальных переменных в функции, так как это запутывает программу и затрудняет поиск ошибок в ней.

В языке С# для обмена информацией между вызываемой и вызывающей функцией предусмотрено **4 вида параметров:**

1. Параметры-значения – подставляются при вызове функции, но не изменяются. Обычно используются в качестве входных параметров функции;
2. Параметры-переменные – подставляются в виде ссылки со служебным словом **ref**, переданное значение переменной может быть изменено внутри функции;
3. Выходные параметры – описываются ключевым словом **out**, используются для передачи результатов;
4. Параметры-массивы (переменное число параметров), описываются с помощью ключевого слова **params**.

**Как функция передает результаты?** Подпрограмма должна вычислять результат и передавать его в место вызова (**передавать**, а не выводить на экран!) Результат должен возвращаться **именно в том типе**, в котором он предполагается по условию задачи.

Функция с результатом передает только одно вычисленное значение (результат) с помощью оператора return. В случае необходимости функции могут передавать несколько результатов через параметры-переменные и выходные параметры. Не стоит объединять несколько разнородных результатов в общий массив или структуру, если по смыслу эти данные не являются единым объектом.

**Как функция сигнализирует об ошибках?** Если подпрограмма не может сформировать результат по каким-то причинам (например, при ошибочных исходных данных), то сообщение об ошибке в теле функции лучше не печатать, а выполнить генерацию исключения.

*Пример описания и вызова функции:*

class Program

{

static bool Kvadrat\_Uravnenie(double a, double b, double c,   
out double x1, out double x2)

{

if (a == 0)

throw new Exception("Уравнение не квадратное.");

double D = b \* b - 4 \* a \* c; // дискриминант

if (D < 0)

return false;

else

{

x1 = (-b + Math.Sqrt(D)) / (2 \* a);

x2 = (-b - Math.Sqrt(D)) / (2 \* a);

return true;

}

}

static void Main(string[] args)

{

double a1, b1, c1, k1, k2;

try

{

Console.WriteLine("Введите коэффициенты кв. уравнения:");

a1 = Double.Parse(Console.ReadLine());

b1 = Double.Parse(Console.ReadLine());

c1 = Double.Parse(Console.ReadLine());

bool f = Kvadrat\_Uravnenie(a1, b1, c1, out k1, out k2);

if (f)

{

Console.WriteLine("Корни квадратного уравнения:");

Console.WriteLine("X1 = {0} X2 = {1}", k1, k2);

}

else

Console.WriteLine("Уравнение не имеет корней.");

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

*Пример алгоритма основной программы и подпрограммы:*



Рисунок 1 – Примеры алгоритмов

**Отчёт должен содержать:**

1. Название лабораторной работы, цель работы;
2. Условие каждой задачи;
3. Математическую формулировку каждой задачи (описание входных и выходных данных, их смысла, типов, ограничений, связь);
4. Алгоритмы каждой подпрограммы и алгоритм основной программы для каждого задания;
5. Код программ с комментариями;
6. Результаты испытаний каждой программы (скриншоты экраны вывода).

**ВАРИАНТЫ для задания 1:**

1. Разработать подпрограмму для вычисления суммы банковского вклада с капитализацией процентов, если известна первоначальная сумма вклада, процент годовых и количество месяцев, на которое делается вклад. Проценты начисляются каждый месяц.
2. Даны два основания и высота равнобедренной трапеции. Разработать подпрограммы для вычисления её периметра и площади.
3. Прямоугольную комнату необходимо оклеить обоями. Составьте подпрограмму для определения площади стен, если известна длина, ширина и высота комнаты. Учесть, что в комнате имеется заданное количество дверей размером 2 м на 80 см и окон размером 1,5 м на 1,5 м. Вычислить также, сколько рулонов обоев потребуется, если стандартная длина рулона 10 м, ширина 60 см (добавить 30% для подгона по рисунку, если обои с рисунком).
4. Треугольник задан координатами своих вершин. Разработать подпрограмму для вычисления периметра треугольника и его площади.
5. Даны стороны параллелограмма и угол при основании (в градусах). Разработать подпрограммы для вычисления его периметра, площади и длины диагоналей.
6. Разработать подпрограмму для вычисления площади треугольника по трем сторонам и подпрограмму определения вида треугольника (остроугольный, тупоугольный, прямоугольный). Учесть, что при некоторых значениях сторон треугольник нельзя построить.
7. Заданы радиус основания и высота цилиндра. Разработать подпрограммы для вычисления площади полной поверхности и объёма цилиндра.
8. Разработать подпрограмму для вычисления расстояния между двумя точками на плоскости. Используя подпрограмму, определить расстояние между каждой парой точек (имеется N точек) и наибольшее расстояние.
9. Разработать подпрограмму для определения, пересекаются ли две окружности на плоскости. Окружности задаются координатами центра и радиусом.
10. Разработать подпрограмму для решения системы двух линейных уравнений по правилу Крамара. Учесть, что система может не иметь решения. Формулы для справки:



1. Разработать подпрограмму для определения, пересекаются ли два отрезка на плоскости. Отрезки задаются координатами своих концов. Найти также точку пересечения двух отрезков.
2. Разработать подпрограмму, которая находит самую длинную последовательность цифр в строке, а также подпрограмму, которая удаляет самую длинную последовательность цифр в строке.
3. Разработать подпрограммы для вычисления площади полной поверхности и объёма конуса, если известны его высота и радиус основания.
4. Разработать подпрограмму для вычисления объёма усеченного конуса, если известны его высота и радиусы обоих оснований.
5. Разработать подпрограмму, которая определяет, есть ли в строке повторяющиеся символы, а также подпрограмму, которая возвращает строку без повторяющихся символов.
6. Разработать подпрограмму, которая определяет, является ли переданное ей натуральное число совершенным. Совершенным называется число, равное сумме всех своих делителей (от 1, но не включая само число).

*Например*, 28 - совершенное число, так как 1+2+4+7+14 = 28.

1. Разработать подпрограммы для вычисления площади и внутреннего угла правильного многоугольника (в градусах), если известна сторона многоугольника и количество его вершин.
2. Разработать подпрограмму для вычисления площади треугольника по стороне и двум прилежащим к ней углам, и подпрограмму для вычисления его высот. Углы задаются в градусах.
3. Разработать подпрограмму для вычисления площади треугольника по двум сторонам и углу между ними, и подпрограмму для вычисления его высот. Угол задается в градусах.
4. Разработать подпрограмму, которая определяет, является ли переданное ей целое многозначное число «счастливым», то есть сумма цифр левой половины числа должна совпадать с суммой цифр правой половины числа (количество разрядов числа произвольно).

**ВАРИАНТЫ для задания 2:**

ВАРИАНТ 1

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти максимальный элемент в каждом массиве, используя функцию программиста;
* В том массиве, где его значение больше, удвоить элементы до середины массива при помощи функции.

ВАРИАНТ 2

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти первый положительный элемент в каждом массиве, используя функцию программиста;
* В том массиве, где значение первого положительного элемента больше, обнулить положительные элементы до максимального элемента при помощи функции.

ВАРИАНТ 3

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти первый нечетный элемент в каждом массиве, используя функцию программиста;
* В том массиве, где это значение меньше, раскрасить нечетные элементы красным цветом при помощи функции.

ВАРИАНТ 4

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти сумму элементов до максимального в каждом массиве, используя функцию программиста;
* В том массиве, где сумма больше, поменять местами первый элемент с максимальным. Обмен осуществить при помощи функции.

ВАРИАНТ 5

* Заданы три одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти количество максимальных элементов каждого массива, используя функцию программиста;
* В том массиве, где максимальных элементов более одного, раскрасить их при выводе желтым цветом при помощи функции.

ВАРИАНТ 6

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти сумму не кратных 5 элементов каждого массива, используя функцию программиста;
* В том массиве, где это значение меньше всего, поменять местами первую и вторую половину массива. Обмен осуществить при помощи функции.

ВАРИАНТ 7

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти сумму кратных 7 элементов каждого массива, используя функцию программиста;
* В том массиве, где это значение больше всего, поменять местами минимальный и последний элемент. Обмен осуществить при помощи функции.

ВАРИАНТ 8

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти максимальный элемент каждого массива, используя функцию программиста;
* В тех массивах, где максимум четный, удвоить значения на четных местах. Удвоение осуществить при помощи функции.

ВАРИАНТ 9

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти среднее геометрическое элементов каждого массива, используя функцию программиста;
* В том массиве, где среднее геометрическое больше всего, поменять местами первый чётный и последний чётный элемент массива. Обмен осуществить при помощи функции.

ВАРИАНТ 10

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти первый отрицательный элемент каждого массива, используя функцию программиста;
* В тех массивах, где первый отрицательный элемент по модулю больше 10, поменять его местами с последним элементом массива. Обмен осуществить при помощи функции.

ВАРИАНТ 11

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти сумму кратных 5 элементов каждого массива, используя функцию программиста;
* В том массиве, где это значение больше всего, поменять местами первый и последний элемент. Обмен осуществить при помощи функции.

ВАРИАНТ 12

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти максимальный элемент каждого массива, используя функцию программиста;
* В тех массивах, где эта максимум четный, обнулить четные по значению элементы. Обнуление осуществить при помощи функции.

ВАРИАНТ 13

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти сумму нечетных по значению элементов каждого массива, используя функцию программиста;
* В том массиве, где эта сумма больше всего, поменять местами максимальный и минимальный элемент массива. Обмен осуществить при помощи функции.

ВАРИАНТ 14

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти среднее арифметическое положительных элементов каждого массива, используя функцию программиста;
* В том массиве, где это значение меньше всего, увеличить положительные элементы на 2. Увеличение осуществить при помощи функции.

ВАРИАНТ 15

* Заданы три одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти произведение нечётных элементов каждого массива, используя функцию программиста;
* В том массиве, где это произведение больше всего, раскрасить при выводе красным цветом положительные элементы после минимального элемента. Раскрашивание осуществить при помощи функции.

ВАРИАНТ 16

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти произведение отрицательных элементов каждого массива, используя функцию программиста;
* В том массиве, где это произведение больше всего, обнулить отрицательные элементы. Обнуление осуществить при помощи функции.

ВАРИАНТ 17

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти позицию первого максимума в массиве, используя функцию программиста;
* Определить сумму элементов, стоящих после первого максимума при помощи функции программиста;
* В том массиве, где полученная сумма больше, исключить из массива максимальный элемент при помощи функции.

ВАРИАНТ 18

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти произведение четных по значению элементов каждого массива, используя функцию программиста;
* В том массиве, где это произведение больше всего, найти сумму и количество четных элементов. Вычисление суммы и количества осуществить при помощи функции.

ВАРИАНТ 19

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти произведение нечетных по значению элементов каждого массива, используя функцию программиста;
* В том массиве, где это произведение больше всего, увеличить отрицательные элементы на 10. Увеличение осуществить при помощи функции.

ВАРИАНТ 20

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти количество нечетных элементов каждого массива, используя функцию.
* В том массиве, где это количество больше всего, раскрасить при выводе нечетные элементы зеленым цветом. Окрашивание элементов осуществить при помощи функции.