**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.

**Задача 1**

**Алгоритм функции** Main:

**Описание входных данных:**

1. **initialAmount** — первоначальная сумма вклада (в рублях или другой валюте). Тип: **double**. Ограничение: **initialAmount > 0**.
2. **annualRate** — годовая процентная ставка (в процентах). Тип: **double**. Ограничение: **annualRate >= 0**.
3. **months** — количество месяцев, на которое делается вклад. Тип: **int**. Ограничение: **months > 0**.

**Описание выходных данных:**

* **finalAmount** — итоговая сумма вклада после указанного количества месяцев с учетом капитализации процентов. Тип: **double**.

**Формула расчета:**

Капитализация процентов производится ежемесячно. Месячная процентная ставка рассчитывается как:

Итоговая сумма вклада вычисляется по формуле:

**Код программы с комментариями:**

using System;

namespace BankDeposit

{

    class Program

    {

        static double CalculateDeposit(double initialAmount, double annualRate, int months)

        {

            // Проверка входных данных

            if (initialAmount <= 0)

                throw new ArgumentException("Первоначальная сумма вклада должна быть положительной.");

            if (annualRate < 0)

                throw new ArgumentException("Годовая процентная ставка не может быть отрицательной.");

            if (months <= 0)

                throw new ArgumentException("Количество месяцев должно быть больше нуля.");

            // Вычисление месячной процентной ставки

            double monthlyRate = annualRate / (12 \* 100);

            // Вычисление итоговой суммы вклада

            double finalAmount = initialAmount \* Math.Pow(1 + monthlyRate, months);

            return finalAmount;

        }

        static void Main(string[] args)

        {

            try

            {

                // Ввод данных

                Console.Write("Введите первоначальную сумму вклада: ");

                double initialAmount = double.Parse(Console.ReadLine());

                Console.Write("Введите годовую процентную ставку (%): ");

                double annualRate = double.Parse(Console.ReadLine());

                Console.Write("Введите количество месяцев: ");

                int months = int.Parse(Console.ReadLine());

                // Вызов функции

                double finalAmount = CalculateDeposit(initialAmount, annualRate, months);

                // Вывод результата

                Console.WriteLine($"Итоговая сумма вклада через {months} месяцев: {finalAmount:F2}");

            }

            catch (Exception ex)

            {

                // Обработка ошибок

                Console.WriteLine($"Ошибка: {ex.Message}");

            }

        }

    }

}

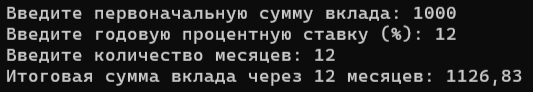
**Тестовые примеры и скриншоты выполнения программ:**

**Тестовый пример 1:**

* **Входные данные:**
  + Первоначальная сумма вклада: 1000
  + Годовая процентная ставка: 12
  + Количество месяцев: 12
* **Ожидаемый результат:**

Итоговая сумма вклада через 12 месяцев: 1126.83

* **Фактический результат:**

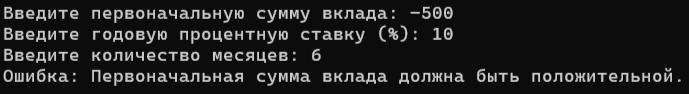


Тестовый пример 2:

* **Входные данные:**
  + Первоначальная сумма вклада: -500
  + Годовая процентная ставка: 10
  + Количество месяцев: 6
* **Ожидаемый результат:**

Ошибка: Первоначальная сумма вклада должна быть положительной.

* **Фактический результат:**

****

**Задача 2**

**Условие задачи :**

* 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.

**Математическая формулировка**

**Описание входных данных:**

1. Два одномерных массива *A* и *B*, размерности которых *n* и *m* соответственно:
   * *A*={*a*0​,*a*1​,...,*an*−1​}
   * *B*={*b*0​,*b*1​,...,*bm*−1​}
   * Размерности массивов *n* и *m* задаются пользователем.
   * Элементы массивов заполняются случайными числами в диапазоне [−100,100].

Описание выходных данных:

1. Максимальные элементы массивов *A* и *B*:
   * *maxA*=max(*A*)
   * *maxB*=max(*B*)
2. Если *maxA*>*maxB*, то элементы массива *A* до середины (*i*<*n*/2) удваиваются:
   * *ai*​=2⋅*ai*​ для всех *i* < *n*/2
3. Если *maxB* > *maxA*, то элементы массива *B* до середины (*j* <*m*/2) удваиваются:
   * *bj*​= 2⋅ *bj*​для всех *j* < *m*/2

**Алгоритм, построенный MS Visio:**



**Код программы с комментариями:**

using System;

namespace ArrayProcessing

{

    class Program

    {

        // Функция для заполнения массива случайными числами

        static void FillArray(int[] array)

        {

            Random rnd = new Random();

            for (int i = 0; i < array.Length; i++)

            {

                array[i] = rnd.Next(-100, 101); // [-100, 100]

            }

        }

        // Функция для вывода массива на экран

        static void PrintArray(int[] array)

        {

            foreach (int num in array)

            {

                Console.Write($"{num,8}");

            }

            Console.WriteLine();

        }

        // Функция для нахождения максимального элемента в массиве

        static int FindMax(int[] array)

        {

            int max = array[0];

            for (int i = 1; i < array.Length; i++)

            {

                if (array[i] > max)

                {

                    max = array[i];

                }

            }

            return max;

        }

        // Функция для удвоения элементов массива до середины

        static void DoubleHalf(int[] array)

        {

            int mid = array.Length / 2;

            for (int i = 0; i < mid; i++)

            {

                array[i] \*= 2;

            }

        }

        static void Main(string[] args)

        {

            try

            {

                // Ввод размерностей массивов

                Console.Write("Введите размер первого массива: ");

                int n = int.Parse(Console.ReadLine());

                Console.Write("Введите размер второго массива: ");

                int m = int.Parse(Console.ReadLine());

                // Проверка размерностей

                if (n <= 0 || m <= 0)

                {

                    throw new ArgumentException("Размерности массивов должны быть положительными.");

                }

                // Создание массивов

                int[] arrayA = new int[n];

                int[] arrayB = new int[m];

                // Заполнение массивов

                FillArray(arrayA);

                FillArray(arrayB);

                // Вывод массивов

                Console.WriteLine("Первый массив:");

                PrintArray(arrayA);

                Console.WriteLine("Второй массив:");

                PrintArray(arrayB);

                // Поиск максимальных элементов

                int maxA = FindMax(arrayA);

                int maxB = FindMax(arrayB);

                Console.WriteLine($"Максимальный элемент первого массива: {maxA}");

                Console.WriteLine($"Максимальный элемент второго массива: {maxB}");

                // Сравнение максимумов и удвоение элементов

                if (maxA > maxB)

                {

                    Console.WriteLine("Удваиваем элементы первого массива до середины.");

                    DoubleHalf(arrayA);

                }

                else if (maxB > maxA)

                {

                    Console.WriteLine("Удваиваем элементы второго массива до середины.");

                    DoubleHalf(arrayB);

                }

                else

                {

                    Console.WriteLine("Максимальные элементы равны. Изменений нет.");

                }

                // Вывод измененных массивов

                Console.WriteLine("Измененный первый массив:");

                PrintArray(arrayA);

                Console.WriteLine("Измененный второй массив:");

                PrintArray(arrayB);

            }

            catch (Exception ex)

            {

                Console.WriteLine($"Ошибка: {ex.Message}");

            }

        }

    }

}

**Тестовые примеры и скриншоты выполнения программ:**

**Тестовый пример 1:**

* **Входные данные:**
  + Размер первого массива: **5**
  + Размер второго массива: **4**
  + Первый массив: **[ -93 57 2 -40 -48]**
  + Второй массив: **[ 62 33 -81 87 ]**
* **Ожидаемый результат:**

Первый массив:

-**93 57 2 -40 -48**

Второй массив:

**62 33 -81 87**

Максимальный элемент первого массива: 57

Максимальный элемент второго массива: 87

Удваиваем элементы первого массива до середины.

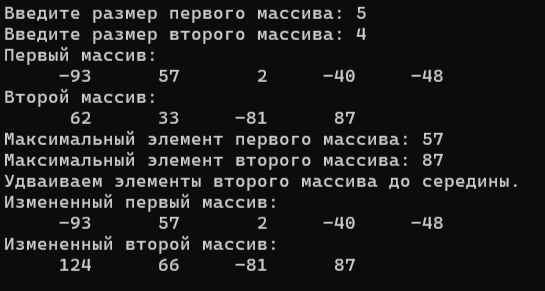
Измененный первый массив:

-**93 57 2 -40 -48**

Измененный второй массив:

**124 66 -81 87**

**Фактический результат:**



**Тестовый пример 2:**

* **Входные данные:**
  + Размер первого массива: **3**
  + Размер второго массива: **6**
  + Первый массив: **[-17 -57 45 ]**
  + Второй массив: **[-21 42 3 19 99 35]**
* **Ожидаемый результат:**

Первый массив:

**-17 -57 45**

Второй массив:

**-21 42 3 19 99 35**

Максимальный элемент первого массива: 45

Максимальный элемент второго массива: 99

Удваиваем элементы первого массива до середины.

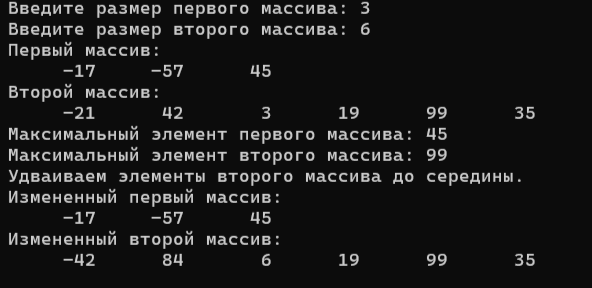
Измененный первый массив:

**-17 -57 45**

Измененный второй массив:

**-21 84 6 19 99 35**

* **Фактический результат:**



**Тестовый пример 3:**

**Входные данные:**

* Размер первого массива: **0**
* Размер второго массива: **5**

**Ожидаемый результат:**

Ошибка: Размерности массивов должны быть положительными.

* **Фактический результат:**

