ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА (САМАРСКИЙ УНИВЕРСИТЕТ)»

Институт Информатики и кибернетики
Кафедра Программных систем

ОТЧЁТ

по лабораторной работе

№1 «Основы языка C#: Классы. Повторение»

по дисциплине «Языки программирования и структуры данных»

Выполнил Фадеев А.М. 6101

Проверил Котенёва С.Э.

Самара

2024

ЗАДАНИЕ

Задание 1.

Прочитать теоретический материал.

Задание 2. «Сортировка»

Описать класс с именем «ArrayVector». Класс описывает вектор в n-мерном пространстве. Координаты конца вектора задаются массивом, количество элементов которого равно n – размерности пространства. Класс должен иметь следующую структуру:

– поле – массив элементов целого типа (координаты вектора в пространстве);

– конструктор с параметром – длиной массива;

– конструктор без параметров, создающий массив из 5 элементов;

– метод установки элемента массива по индексу SetElement() (параметры метода – индекс элемента и устанавливаемое значение);

– метод чтения элемента массива по индексу GetElement() (параметр метода – индекс элемента);

– метод GetNorm() получения модуля (длины, нормы) вектора;

– метод SumPositivesFromChetIndex() подсчета суммы всех положительных элементов массива с четными номерами;

– метод SumLessFromNechetIndex() подсчета суммы тех элементов массива, которые имеют нечетные номера и одновременно меньше среднего значения всех модулей элементов массива;

– метод MultChet() подсчета произведения всех четных положительных элементов (по значению);

– метод MultNechet() подсчета произведения всех нечетных элементов (по значению), не делящихся на три. Примечание: во всех четырех методах производящих действия с элементами массива нумерация элементов массива для конечного пользователя должна начинаться с единицы. То есть, в массиве [2, 3, 4, 5] элемент со значением «3» это второй элемент по индексу с точки зрения пользователя;

– метод SortUp() сортировки массива по возрастанию;

– метод SortDown() сортировки массива по убыванию.

Где необходимо выбрасывать разные типы исключений.

Задание 3.

Добавить класс с именем «Vectors», содержащий публичные статические методы:

– сложения двух векторов Sum(), который принимает в качестве параметра 2 объекта типа ArrayVector и возвращает новый объект ArrayVector;

– скалярного произведения двух векторов Scalar(), который принимает в качестве параметра 2 объекта типа ArrayVector и возвращает вещественное число;

– умножения вектора на число MultNumber(), который принимает в качестве параметра объект типа ArrayVector и вещественное число и возвращает новый объект ArrayVector;

– получения модуля (длины) вектора GetNormSt(), который принимает в качестве параметра объект типа ArrayVector и возвращает вещественное число.

Выбрасывать исключения в методах Sum() и Scalar() в случае невозможности проведения указанных действий над векторами (возможно, FormatException). В классе Program в методе Main() реализовать всю функциональность описанного класса – написать программу, проверяющую все (!) разработанные элементы класса.

В классе Program проверить работоспособность всех элементов описанного класса.

Задание 4.

Подготовить отчет о работе.

КОД ПРОГРАММЫ

```
namespace Lab01;

public class ArrayVector
{
    private int[] _vector;

    public int this[int idx]
    {
        get
        {
            if (idx < 0 || idx >= _vector.Length)
            {
                throw new IndexOutOfRangeException("Vector index out of
range");
            }
            return _vector[idx];
        }
        set
        {
            if (idx < 0 || idx >= _vector.Length)
            {
                throw new IndexOutOfRangeException("Vector index out of
range");
            }
            _vector[idx] = value;
        }
    }

    public int Length => _vector.Length;

    public ArrayVector(int length)
    {
        _vector = new int[length];
    }

    public ArrayVector()
    {
        _vector = new int[5];
    }

    public double GetNorm()
    {
        double acc = 0;
        for (int i = 0; i < _vector.Length; i++)
        {
            acc += Math.Pow(_vector[i], 2);
        }

        return Math.Sqrt(acc);
    }

    public int SumPositivesWithEvenIndex()
    {
        int acc = 0;
        for (int i = 0; i < _vector.Length; i += 2)
        {
            if (_vector[i] > 0)
            {
                acc += _vector[i];
            }
        }
```

```
    return acc;
}

public int SumLessAverageAbsoluteWithOddIndex()
{
    if (_vector.Length == 0)
    {
        return 0;
    }

    int average = 0;
    for (int i = 0; i < _vector.Length; i++)
    {
        average += Math.Abs(_vector[i]);
    }

    average /= _vector.Length;


    int acc = 0;
    for (int i = 1; i < _vector.Length; i += 2)
    {
        if (_vector[i] < average)
        {
            acc += _vector[i];
        }
    }

    return acc;
}

public int MultiplyEven()
{
    int result = 1;
    for (int i = 0; i < _vector.Length; i++)
    {
        if (_vector[i] > 0 && _vector[i] % 2 == 0)
        {
            result *= _vector[i];
        }
    }

    return result;
}

public int MultiplyOdd()
{
    int result = 1;
    for (int i = 0; i < _vector.Length; i++)
    {
        if (_vector[i] % 2 != 0 && _vector[i] % 3 != 0)
        {
            result *= _vector[i];
        }
    }

    return result;
}

public void SortUp()
{
    int n = _vector.Length;
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
```

```
                if (_vector[j] > _vector[j + 1])
                {
                    (_vector[j], _vector[j + 1]) = (_vector[j + 1],
_vector[j]);
                }
            }
        }
    }

    public void SortDown()
    {
        int n = _vector.Length;
        for (int i = 0; i < n - 1; i++)
        {
            for (int j = 0; j < n - i - 1; j++)
            {
                if (_vector[j] < _vector[j + 1])
                {
                    (_vector[j], _vector[j + 1]) = (_vector[j + 1],
_vector[j]);
                }
            }
        }
    }

    public void Log(string message = "")
    {
        if (message != "")
        {
            Console.Write(message + ": ");
        }
        Console.Write("{");
        for (int i = 0; i < _vector.Length; i++)
        {
            if (i == _vector.Length - 1)
            {
                Console.Write(_vector[i]);
            }
            else
            {
                Console.Write(_vector[i] + ", ");
            }
        }
        Console.WriteLine("}");
    }

    public static ArrayVector GetFromUserInput()
    {
        string? inp;
        do
        {
            Console.WriteLine("Выберете как хотите заполнить вектор:\n" +
                              "1 - Случайно\n" +
                              "2 - Ручной ввод");
            inp = Console.ReadLine();
        } while (inp != "1" && inp != "2");

        int length;
        do
        {
            Console.Write("Введите длину вектора: ");
        } while (!int.TryParse(Console.ReadLine(), out length) || length <=
0);

        var vec = new ArrayVector(length);
```

```
        if (inp == "1")
        {
            var r = new Random();
            for (int i = 0; i < length; i++)
            {
                vec[i] = r.Next(100);
            }

            return vec;
        }
        else
        {
            for (int i = 0; i < length; i++)
            {
                int value;
                do
                {
                    Console.Write($"Введите значение координаты {{{i}}}: ");
                    inp = Console.ReadLine();
                } while (!int.TryParse(inp, out value));

                vec[i] = value;
            }

            return vec;
        }
    }
}
namespace Lab01;

public class Vectors
{
    public static ArrayVector Sum(ArrayVector a, ArrayVector b)
    {
        if (a.Length != b.Length)
        {
            throw new Exception("Vectors norms are not equals");
        }

        var vec = new ArrayVector(a.Length);
        for (int i = 0; i < vec.Length; i++)
        {
            vec[i] = a[i] + b[i];
        }

        return vec;
    }

    public static double ScalarMultiply(ArrayVector a, ArrayVector b)
    {
        if (a.Length != b.Length)
        {
            throw new Exception("Vectors norms are not equal");
        }

        var result = 0;
        for (int i = 0; i < a.Length; i++)
        {
            result += a[i] * b[i];
        }

        return result;
    }

    public static ArrayVector MultiplyByNumber(ArrayVector vector, int
number)
```

```
    {
        for (int i = 0; i < vector.Length; i++)
        {
            vector[i] *= number;
        }

        return vector;
    }

    public static double GetNormSt(ArrayVector vector)
    {
        return vector.GetNorm();
    }
}

namespace Lab01;

public static class Program
{
    public static void Main(string[] args)
    {
        Greeting();
        while (true)
        {
            Console.WriteLine("Выберете действие:\n\n" +
                              "\t1 - Посчитать сумму всех положительных
элементов массива с четными номерами\n" +
                              "\t2 - Посчитать сумму элементов массива с
нечетным индексом и одновременно меньше среднего значения всех модулей
элементов массива\n" +
                              "\t3 - Сортировка по возрастанию\n" +
                              "\t4 - Сортировка по убыванию\n" +
                              "\t5 - Сумма векторов\n" +
                              "\t6 - Скалярное умножение\n" +
                              "\t7 - Умножить вектор на число\n" +
                              "\t8 - Посчитать модуль вектора\n" +
                              "\t0 - Выход из программы\n");

            string? inp = Console.ReadLine();

            switch (inp)
            {
                case "1":
                    CountSumPositiveNumbersWithEvenIndex();
                    break;
                case "2":
                    CountSumLessAverageAbsWithOddIndex();
                    break;
                case "3":
                    SortAscending();
                    break;
                case "4":
                    SortDescending();
                    break;
                case "5":
                    SumVectors();
                    break;
                case "6":
                    ScalarMultiply();
                    break;
                case "7":
                    MultiplyVectorByNumber();
                    break;
                case "8":
                    GetVectorNorm();
                    break;
```

```
                case "0":
                    Console.WriteLine("До встречи!");
                    return;
                default:
                    Console.WriteLine("Нет такого пункта в меню");
                    break;
            }

            Console.WriteLine("Нажмите любую клавишу для продолжения...");
            Console.ReadKey();
        }
    }

    public static void CountSumPositiveNumbersWithEvenIndex()
    {
        var vector = ArrayVector.GetFromUserInput();

        vector.Log("Созданный вектор");

        int sum = vector.SumPositivesWithEvenIndex();

        Console.WriteLine("Сумма положительных элементов с четными индексами:
" + sum);
    }

    public static void CountSumLessAverageAbsWithOddIndex()
    {
        int length;
        Console.Write("Введите длину вектора: ");
        string inp = Console.ReadLine();

        while (!int.TryParse(inp, out length))
        {
            Console.Write("Введите длину вектора: ");
            inp = Console.ReadLine();
        }

        var vector = ArrayVector.GetFromUserInput();

        vector.Log("Созданный вектор");

        int sum = vector.SumLessAverageAbsoluteWithOddIndex();

        Console.WriteLine("Сумма элементов с нечетными индексами, которые
меньше среднего значения всех модулей: " + sum);
    }

    public static void SortAscending()
    {
        var vector = ArrayVector.GetFromUserInput();

        vector.Log("Исходный вектор");

        vector.SortUp();

        vector.Log("Отсортированный вектор по возрастанию");
    }

    public static void SortDescending()
    {
        var vector = ArrayVector.GetFromUserInput();

        vector.Log("Исходный вектор");

        vector.SortDown();
```

```
        vector.Log("Отсортированный вектор по убыванию");
    }

    public static void SumVectors()
    {
        var vec1 = ArrayVector.GetFromUserInput();
        var vec2 = ArrayVector.GetFromUserInput();

        vec1.Log("Первый вектор");
        vec2.Log("Второй вектор");

        var result = Vectors.Sum(vec1, vec2);

        result.Log("Результирующий векторсложения");
    }

    public static void ScalarMultiply()
    {
        var vec1 = ArrayVector.GetFromUserInput();
        var vec2 = ArrayVector.GetFromUserInput();

        vec1.Log("Первый вектор");
        vec2.Log("Второй вектор");

        var result = Vectors.ScalarMultiply(vec1, vec2);

        Console.WriteLine("Результат скалярного произведения: " + result);
    }

    public static void MultiplyVectorByNumber()
    {
        var vector = ArrayVector.GetFromUserInput();

        vector.Log("Созданный вектор");

        int number;
        string inp;
        do
        {
            Console.Write("Введите число на которое умножить вектор: ");
            inp = Console.ReadLine();
        } while (!int.TryParse(inp, out number));

        var result = Vectors.MultiplyByNumber(vector, number);

        result.Log("Результат умножения вектора на число");
    }

    public static void GetVectorNorm()
    {
        var vector = ArrayVector.GetFromUserInput();

        vector.Log("Созданный вектор");

        Console.WriteLine($"Модуль созданного вектора:
{vector.GetNorm():0.00}");
    }

    public static void Greeting()
    {
        Console.WriteLine("Языки Программирования и Структуры Данных\n" +
                          "Лабораторная работа #1\n" +
                          "Выполнил студент группы 6101-020302D - Фадеев
Артем");
    }
}
```

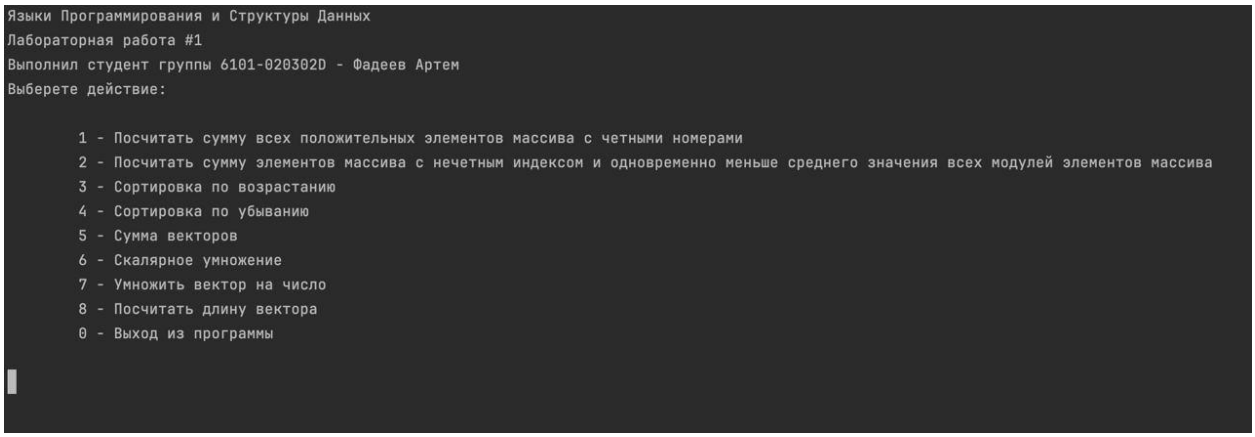

Рисунок 1 – Главное меню программы

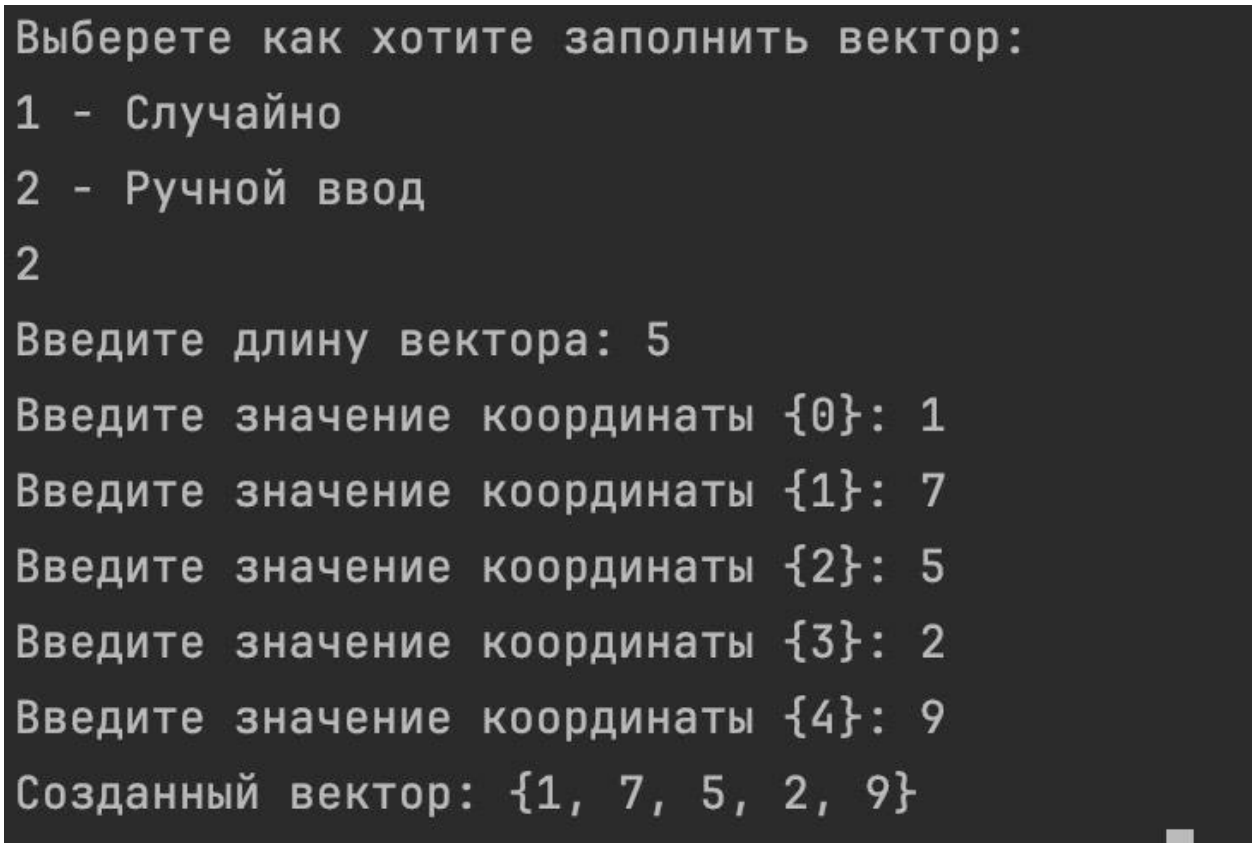

Рисунок 2 – Создание нового вектора

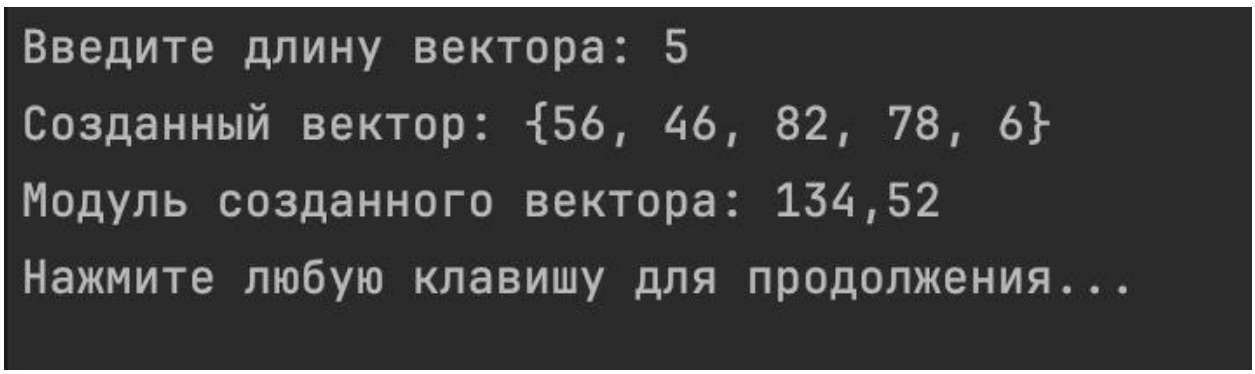

Рисунок 3 – Нахождение модуля вектора

```
Выберете как хотите заполнить вектор:
1 - Случайно
2 - Ручной ввод
1
Введите длину вектора: 5
Выберете как хотите заполнить вектор:
1 - Случайно
2 - Ручной ввод
1
Введите длину вектора: 5
Первый вектор: {72, 84, 58, 25, 62}
Второй вектор: {55, 85, 92, 3, 90}
Результирующий векторсложения: {127, 169, 150, 28, 152}
Нажмите любую клавишу для продолжения...
```

Рисунок 4 – Сумма векторов

```
Введите длину вектора: 12
Исходный вектор: {26, 38, 31, 17, 75, 43, 59, 15, 91, 78, 80, 97}
Отсортированный вектор по возрастанию: {15, 17, 26, 31, 38, 43, 59, 75, 78, 80, 91, 97}
Нажмите любую клавишу для продолжения...
```

Рисунок 5 – Сортировка по возрастанию

ВЫВОДЫ

В лабораторной работе были использованы конструкции языка:

- форматированный вывод информации на консоль;
- оператор switch;
- условные операторы if/else/else if;
- классы;
- конструкторы класса;
- статические и динамические методы класса;
- поля класса;
- конструкция try-catch.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов [Текст]/Т.А. Павловская. – СПб.: Питер, 2007. – 432 с.