

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«Национальный исследовательский университет ИТМО»**  
**(Университет ИТМО)**

Факультет инфокоммуникационных технологий

Отчет по дисциплине: **«Объектно-ориентированное программирование»**  
Лабораторная работа 5. “Создание и использование массивов”

Выполнил: Абдулов  
Илья Александрович

Группа: К3221

Проверил: Васильев  
Сергей Юрьевич

Санкт-Петербург  
2023

## Содержание

Введение . . . . .	3
1 Упражнение 1. Работа с массивом размерного типа данных . .	4
2 Упражнение 2. Перемножение матриц . . . . .	6
3 Упражнение 3. Обработка данных массива . . . . .	9
Заключение . . . . .	12
Список использованных источников . . . . .	13

## Введение

Целью работы является изучение массивов и приобретение навыков работы с ними. Для достижения поставленной цели необходимо решить следующие упражнения:

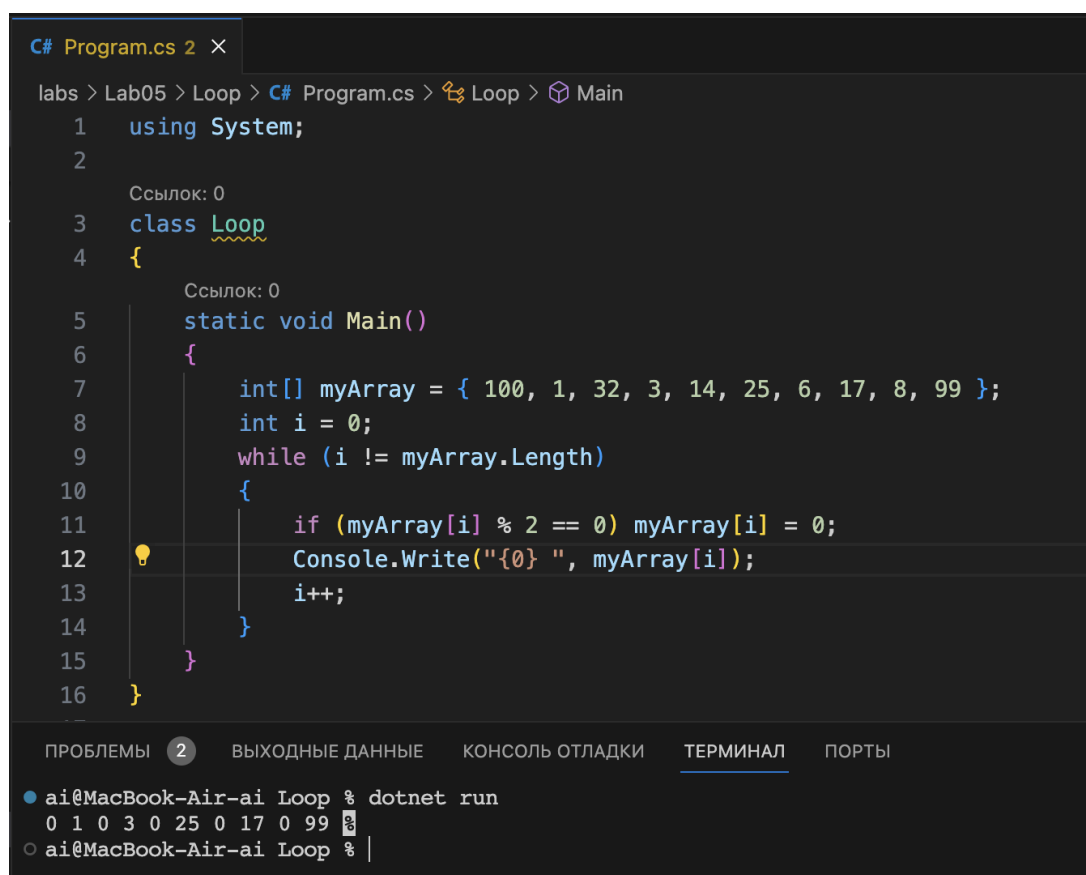
1. Работа с массивом размерного типа данных;
2. Перемножение матриц;
3. Обработка данных массива.

Лабораторная выполняется на компьютере под macOS, поэтому некоторые этапы и инструменты выполнения работы будут отличаться от стандартных в Windows.

## 1 Упражнение 1. Работа с массивом размерного типа данных

Реализуем в проекте Loop массив для хранения данных.

В методе Main создадим массив, заполненный числами. Реализуем вывод элементов массива в строку через пробел с помощью оператора цикла for. Вместо четных чисел будут выводиться нули. Код метода Main и вывод приложения представлены ниже (Рисунок 1.1).



```
C# Program.cs 2 x
labs > Lab05 > Loop > C# Program.cs > Loop > Main
1  using System;
2
3  class Loop
4  {
5      static void Main()
6      {
7          int[] myArray = { 100, 1, 32, 3, 14, 25, 6, 17, 8, 99 };
8          int i = 0;
9          while (i != myArray.Length)
10         {
11             if (myArray[i] % 2 == 0) myArray[i] = 0;
12             Console.Write("{0} ", myArray[i]);
13             i++;
14         }
15     }
16 }

ПРОБЛЕМЫ 2 Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
ai@MacBook-Air-ai Loop % dotnet run
0 1 0 3 0 25 0 17 0 99
ai@MacBook-Air-ai Loop %
```

Рисунок 1.1 — Метод Main и вывод

Заменяем массив с явной инициализацией на массив, размер которого вводит с клавиатуры пользователь.

Сначала просто объявим массив, запросим у пользователя его размер. Реализуем инициализацию элементов массива в цикле и выведем на экран элементы массива циклом foreach (Рисунок 1.2).

```
C# Program.cs 2 X
labs > Lab05 > Loop > C# Program.cs > Loop > Main
5 static void Main()
6 {
7     int[] MyArray;
8     int n = int.Parse(Console.ReadLine());
9     MyArray = new int[n];
10    for (int i = 0; i < MyArray.Length; ++i)
11    {
12        Console.Write("a[{0}]=", i);
13        MyArray[i] = int.Parse(Console.ReadLine());
14    }
15    foreach (int x in MyArray) Console.Write("{0} ", x);
16    Console.WriteLine();
17    int j = 0;
18    while (j != MyArray.Length)
19    {
20        if (MyArray[j] % 2 == 0) MyArray[j] = 0;
21        Console.Write("{0} ", MyArray[j]);
22        j++;
23    }

ПРОБЛЕМЫ 2 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

● ai@MacBook-Air-ai Loop % dotnet run
5
a[0]=32
a[1]=9
a[2]=4
a[3]=7
a[4]=16
32 9 4 7 16
0 9 0 7 0 %
○ ai@MacBook-Air-ai Loop %
```

Рисунок 1.2 — Инициализация массива пользователем

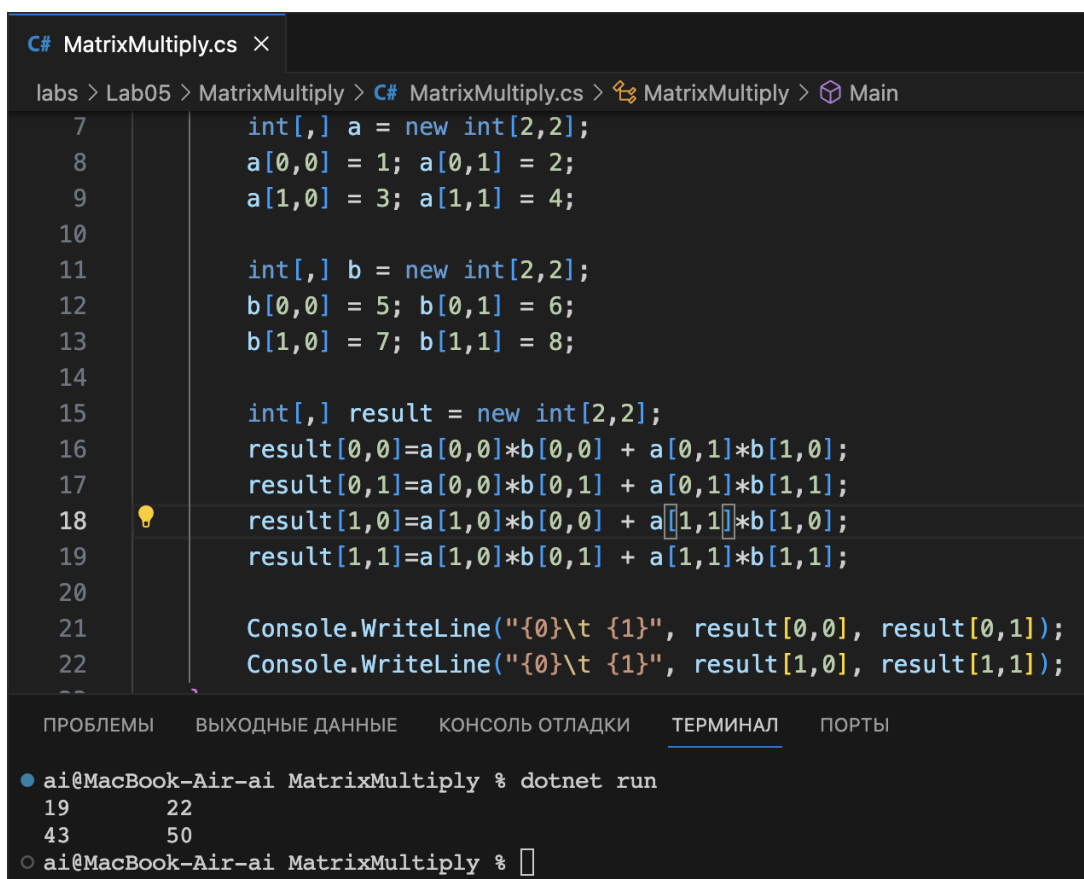
## 2 Упражнение 2. Перемножение матриц

Создадим проект MatrixMultiply, переименуем файл и класс Program на MatrixMultiply.

В методе Main класса MatrixMultiply объявим массив целых чисел a размером 2x2, пока заполним его числами 1, 2, 3, 4. Аналогичным образом добавьте в метод Main объявление еще одного массива целых чисел размером 2x2, назовите его b.

В методе Main объявите еще один массив целых чисел размером 2x2 и назовите его result. Заполните его значениями, рассчитанными по формулам перемножения матриц.

Выведем на консоль значения массива result. Убедимся в том, что перемножение матриц выполнено корректно (Рисунок 2.1).



```
C# MatrixMultiply.cs x
labs > Lab05 > MatrixMultiply > C# MatrixMultiply.cs > MatrixMultiply > Main
7      int[,] a = new int[2,2];
8      a[0,0] = 1; a[0,1] = 2;
9      a[1,0] = 3; a[1,1] = 4;
10
11     int[,] b = new int[2,2];
12     b[0,0] = 5; b[0,1] = 6;
13     b[1,0] = 7; b[1,1] = 8;
14
15     int[,] result = new int[2,2];
16     result[0,0]=a[0,0]*b[0,0] + a[0,1]*b[1,0];
17     result[0,1]=a[0,0]*b[0,1] + a[0,1]*b[1,1];
18     result[1,0]=a[1,0]*b[0,0] + a[1,1]*b[1,0];
19     result[1,1]=a[1,0]*b[0,1] + a[1,1]*b[1,1];
20
21     Console.WriteLine("{0}\t {1}", result[0,0], result[0,1]);
22     Console.WriteLine("{0}\t {1}", result[1,0], result[1,1]);
23
PROБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ
● ai@MacBook-Air-ai MatrixMultiply % dotnet run
19      22
43      50
○ ai@MacBook-Air-ai MatrixMultiply %
```

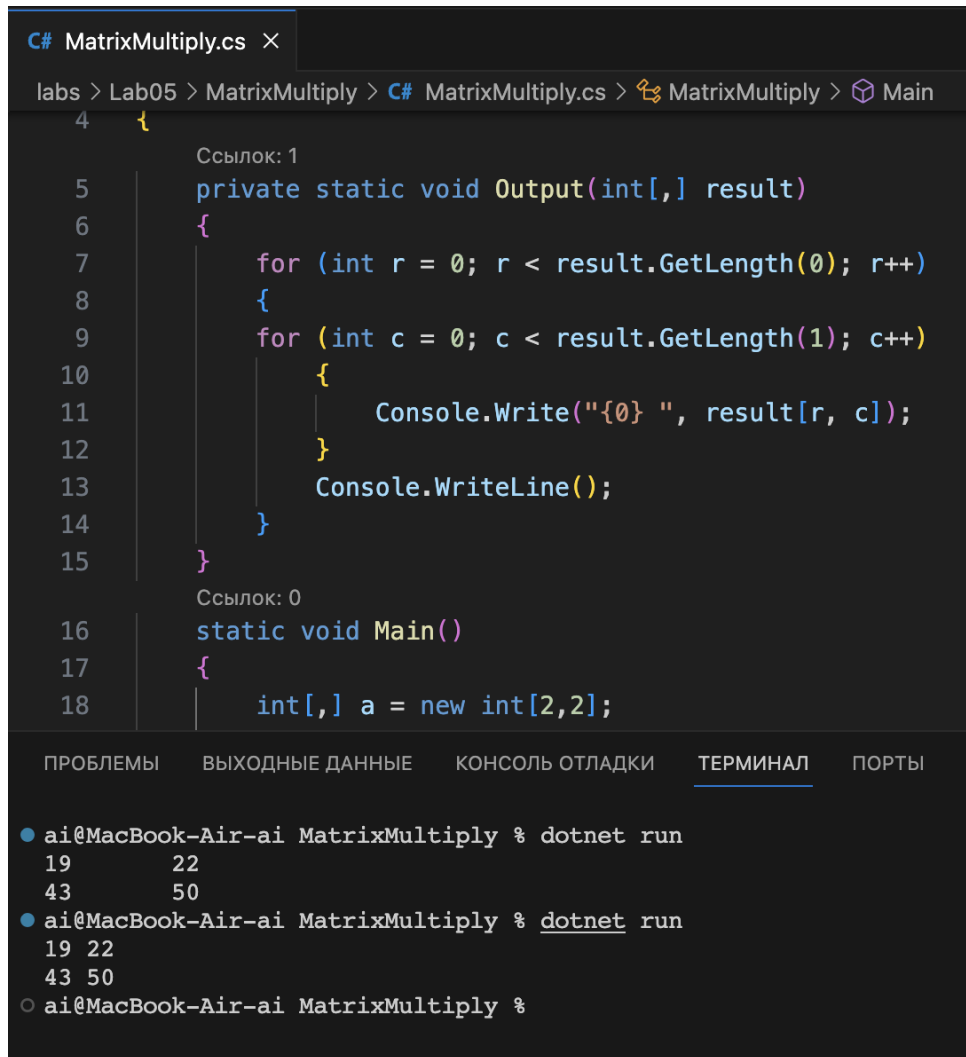
Рисунок 2.1 — Перемножение двух матриц

С помощью механизма рефакторинга извлечем метод для вывода результирующего массива. Появится новый статический метод Output.

Этот метод не возвращает значений и принимает в качестве параметра массив целых чисел с именем result.

Изменим метод Output, используя вместо четырех инструкций Console.WriteLine два вложенных цикла for.

Запустим и протестируем программу (Рисунок 2.2).



```
C# MatrixMultiply.cs ×
labs > Lab05 > MatrixMultiply > C# MatrixMultiply.cs > MatrixMultiply > Main
4 {
    Ссылка: 1
5     private static void Output(int[,] result)
6     {
7         for (int r = 0; r < result.GetLength(0); r++)
8         {
9             for (int c = 0; c < result.GetLength(1); c++)
10            {
11                Console.Write("{0} ", result[r, c]);
12            }
13            Console.WriteLine();
14        }
15    }
    Ссылка: 0
16    static void Main()
17    {
18        int[,] a = new int[2,2];

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

● ai@MacBook-Air-ai MatrixMultiply % dotnet run
19      22
43      50
● ai@MacBook-Air-ai MatrixMultiply % dotnet run
19 22
43 50
○ ai@MacBook-Air-ai MatrixMultiply %
```

Рисунок 2.2 — Тестирование метода Output

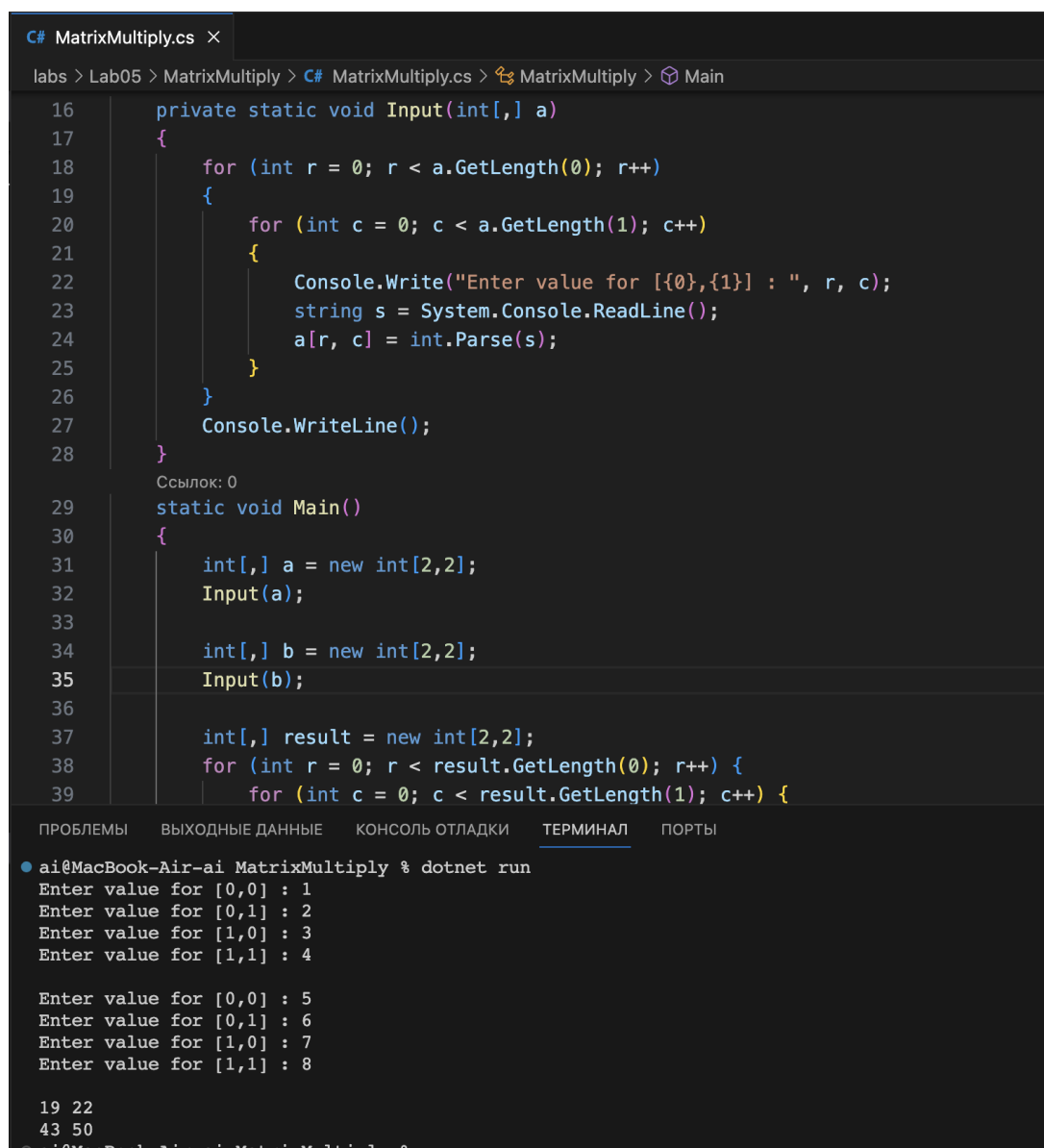
Создадим метод Multiply, рассчитывающий и возвращающий значения массива result. Этот метод возвращает массив целых чисел второго ранга и принимает в качестве параметров два массива целых чисел второго ранга с именами a и b.

В методе Multiply удалим инструкции инициализации массива result. Для этого добавим два вложенных цикла for. Во внешнем цикле для итерации по каждому индексу первого измерения массива result будем использовать переменную r. Во внутреннем цикле для итерации по

каждому индексу второго измерения массива `result` будем использовать переменную `c`.

Обеспечим возможность считывания значений первой и второй матрицы с консоли. Для этого объявим новый статический метод `Input`. В этот метод просто были перенесены инструкции инициализации матрицы. Заменяем инструкции инициализации элементов массива на два вложенных цикла `for`.

Откомпилируем программу и исправим ошибки. Убедимся в том, что перемножение матриц работает корректно (Рисунок 2.3).



```
C# MatrixMultiply.cs ×
labs > Lab05 > MatrixMultiply > C# MatrixMultiply.cs > MatrixMultiply > Main

16 private static void Input(int[,] a)
17 {
18     for (int r = 0; r < a.GetLength(0); r++)
19     {
20         for (int c = 0; c < a.GetLength(1); c++)
21         {
22             Console.WriteLine("Enter value for [{0},{1}] : ", r, c);
23             string s = System.Console.ReadLine();
24             a[r, c] = int.Parse(s);
25         }
26     }
27     Console.WriteLine();
28 }
    Ссылка: 0
29 static void Main()
30 {
31     int[,] a = new int[2,2];
32     Input(a);
33
34     int[,] b = new int[2,2];
35     Input(b);
36
37     int[,] result = new int[2,2];
38     for (int r = 0; r < result.GetLength(0); r++) {
39         for (int c = 0; c < result.GetLength(1); c++) {

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ
● ai@MacBook-Air-ai MatrixMultiply % dotnet run
Enter value for [0,0] : 1
Enter value for [0,1] : 2
Enter value for [1,0] : 3
Enter value for [1,1] : 4

Enter value for [0,0] : 5
Enter value for [0,1] : 6
Enter value for [1,0] : 7
Enter value for [1,1] : 8

19 22
43 50
ai@MacBook-Air-ai MatrixMultiply %
```

Рисунок 2.3 — Перемножение матриц 2x2 с пользовательскими данными



### 3 Упражнение 3. Обработка данных массива

В новом проекте DataProcess в методе Main инициализируем массив `a` размерностью 4 и реализуем заполнение с клавиатуры (Рисунок 3.1).

Добавим в программу методы определения суммы всех элементов, среднего значения, суммы отрицательных элементов, суммы элементов с четными номерами для обработки данных массива (Рисунок 3.2).

Запустим и протестируем приложение обработки данных (Рисунок 3.3).

```
static void Main()
{
    int[] a = new int[4];
    for (int ix = 0; ix < a.GetLength(0); ix++)
    {
        Console.Write("Enter value for [{0}] : ", ix);
        string val = System.Console.ReadLine();
        a[ix] = int.Parse(val);
    }
    Console.WriteLine();
    int s = Sum(a);
    Console.WriteLine("Сумма: {0}", s);
    double avg = Avg(a, s);
    Console.WriteLine("Среднее значение: {0}", avg);
    int negs = NegativeSum(a);
    Console.WriteLine("Сумма отрицательных: {0}", negs);
    int isum = IndexSum(a);
    Console.WriteLine("Сумма с четными i: {0}", isum);
}
```

Рисунок 3.1 — Метод Main обработки данных

```

C# MatrixMultiply.cs  C# DataProcess.cs 2 x
labs > Lab05 > DataProcess > C# DataProcess.cs > DataProcess > NegativeSum

Ссылка: 1
12 private static double Avg(int[] a, int s)
13 {
14     return (double)s / a.GetLength(0);
15 }

Ссылка: 1
16 private static int NegativeSum(int[] a)
17 {
18     int s = 0;
19     for (int i = 0; i < a.GetLength(0); i++)
20     {
21         if (a[i] < 0)
22             s += a[i];
23     }
24     return s;
25 }

Ссылка: 1
26 private static int IndexSum(int[] a)
27 {
28     int s = 0;
29     // Сумма элементов с четными индексами
30     for (int i = 0; i < a.GetLength(0); i+=2)
31     {
32         s += a[i];
33     }
34     return s;
35 }

Ссылка: 0
36 static void Main()
```

Рисунок 3.2 — Методы обработки данных

```
ПРОБЛЕМЫ 2 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

● ai@MacBook-Air-ai DataProcess % dotnet run
Enter value for [0] : 4
Enter value for [1] : 7
Enter value for [2] : -8
Enter value for [3] : -11

Сумма: -8
Среднее значение: -2
Сумма отрицательных: -19
Сумма с четными i: -4
● ai@MacBook-Air-ai DataProcess % dotnet run
Enter value for [0] : 3
Enter value for [1] : -6
Enter value for [2] : 2
Enter value for [3] : -2

Сумма: -3
Среднее значение: -0,75
Сумма отрицательных: -8
Сумма с четными i: 5
○ ai@MacBook-Air-ai DataProcess % |
```

Рисунок 3.3 — Тестирование приложения

## Заключение

В результате проделанной лабораторной работы были изучены и приобретены навыки работы с массивами. Для достижения поставленной цели были решены следующие упражнения:

1. Реализована работа с массивом размерного типа данных;
2. Реализовано перемножение матриц;
3. Реализована обработка данных массива.

Все программы и код выполненных упражнений доступны для просмотра и скачивания на платформе GitHub[1].

## **Список использованных источников**

1. Веб-сервис GitHub : [Электронный ресурс] : облачная платформа для хостинга IT-проектов и совместной разработки.  
— URL: <https://github.com/estle/itmo-uni/tree/main/sem3/OOP/labs/Lab05> (дата обращения: 21.12.2023).