

Министерство образования Московской области
Государственное бюджетное профессиональное образовательное учреждение
Московской области «Колледж «Коломна»

09.02.07

ОТЧЕТ

По лабораторным работам
ОП.04 Основы алгоритмизации и программирование
ККОО.ОАИП4211121.000

Студент

Самойлова К.А

Преподаватель

Тихонова А. А.

Дата защиты _____

Оценка _____

2023

Лабораторная работа №9. Методы работы с элементами одномерного массива

Часть 1. Массив как параметр

1. Заданы массивы A1(6), A2(7), A3(4), A4(6), состоящие из положительных и отрицательных чисел. Найти минимальное значение положительных чисел массива и определить максимальное среди них. Нахождение минимального значения положительных чисел массива, заполнение массивов и вывод на печать оформить в виде методов.

```
2. using System;
3. using System.Collections.Generic;
4. using System.Linq;
5. using System.Text;
6. using System.Threading.Tasks;
7.
8. namespace laba9
9. {
10.     internal class Program
11.     {
12.         static void Metod1(int LeghtMas)
13.         {
14.             Console.WriteLine($"Массив A ({LeghtMas})");
15.             int[] myArray = new int[LeghtMas];
16.             Random rnd = new Random();
17.             for (int i = 0; i < LeghtMas; i++)
18.             {
19.                 myArray[i] = rnd.Next(-100, 100);
20.                 Console.WriteLine($" {myArray[i]} ");
21.             }
```

Лис

ККОО.ОАИП4211121.000

3

Изм. Лис № докум. Подпись Дат

```

22.
23.     int Min = 999;
24.     int Max = 0;
25.     for (int i = 0; i < myArray.Length; i++)
26.     {
27.         if (myArray[i] < Min && myArray[i] >= 0) Min = myArray[i];
28.         if (myArray[i] > Max) Max = myArray[i];
29.     }
30.     Console.WriteLine($"мин полож число A({LeghtMas}) - {Min}, а
макс - {Max}");
31.
32.     }
33.
34.
35.     static void Main(string[] args)
36.     {
37.         Metod1(6);
38.         Metod1(7);
39.         Metod1(4);
40.         Metod1(6);
41.         Console.ReadLine();
42.
43.
44.
45.
46.     }
47. }
48. }

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		4

2. Заданы массивы A(7), B(4), C(12), состоящие из положительных и отрицательных чисел. Определить, в каком месте больше среднее арифметическое положительных чисел. Подсчет среднего арифметического положительных чисел массива оформить в виде метода. Заполнение массивов и вывод на печать оформить в виде методов.

```
namespace ConsoleApp2
{
    internal class Program
    {
        static void Print(int[] a)
        {
            for (int i = 0; i < a.Length; i++)
            {
                Console.Write(a[i] + " ");
            }
            Console.WriteLine();
        }
        static void Input(int[] a)
        {
            Random rnd = new Random();
            for (int i = 0; i < a.Length; i++)
            {
                a[i] = rnd.Next(-10, 10);
            }
        }
        static double Cnt(double[] a, double mid)
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		5

```

{
    double sum = 0;
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] > 0)
        {
            sum += a[i];
        }
    }

    mid = sum / a.Length;
    return mid;
}

static void Max3(int a, int b, int c)
{
    if (a < b)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное среднего арифметического
положительных чисел массива C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное среднего арифметического
положительных чисел массива B = {b}");
        }
    }
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		6

```

    }
    if (a < c)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное среднего арифметического
положительных чисел массива C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное среднего арифметического
положительных чисел массива B = {b}");
        }
    }
    else
    {
        Console.WriteLine($" Максимальное среднего арифметического
положительных чисел массива A = {a}");
    }
}
static void Main(string[] args)
{
    int[] A = new int[7];
    int[] B = new int[4];
    int[] C = new int[12];
    Input(A);
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		7

```

    Input(B);
    Input(C);
    Print(A);
    Print(B); ;
    Print(C);
    Max3(Cnt(A), Cnt(B), Cnt(C));
    Console.ReadKey();
}
}
}

```

3. Заданы массивы A(7), B(4),C(9), состоящие из положительных и отрицательных чисел. Определить, в каком массиве больше отрицательных элементов. Подсчет количества отрицательных элементов оформить в виде метода. Заполнение массивов и вывод на печать оформить в виде методов.

```

{
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(a[i] + " ");
    }
    Console.WriteLine();
}
static void Input(int[] a)
{
    Random rnd = new Random();
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-10, 10);
    }
}

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		8

```

}
static int NegativeCnt(int[] a)
{
    int cnt = 0;
    for(int i = 0; i < a.Length; i++)
    {
        if (a[i] < 0)
        {
            cnt ++;
        }
    }
    return cnt;
}
static void Max3(int a, int b, int c)
{
    if(a < b)
    {
        if(b < c)
        {
            Console.WriteLine($"Максимальное число отрицательных
элементов в массиве C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное число отрицательных
элементов в массиве B = {b}");
        }
    }
    if(a < c)

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		9

```

    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное число отрицательных
элементов в массиве C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное число отрицательных
элементов в массиве B = {b}");
        }
    }
else
{
    Console.WriteLine($"Максимальное число отрицательных
элементов в массиве A = {a}");
}
}
static void Main(string[] args)
{
    int[] A = new int[7];
    int[] B = new int[4];
    int[] C = new int[9];
    Input(A);
    Input(B);
    Input(C);
    Print(A);
    Print(B);
    Print(C);
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		10

```

Max3(NegativeCnt(A), NegativeCnt(B), NegativeCnt(C));
Console.ReadKey();
}

```

```

-1 -7 5 -8 9 -7 3
-1 -7 5 -8
-1 -7 5 -8 9 -7 3 2 -9
Максимальное число отрицательных элементов в массиве C = 5

```

4. Заданы массивы X(8),Y(7),Z(9), состоящие из положительных и отрицательных чисел. Найти максимальное значение отрицательных чисел в каждом массиве и определить минимальное среди них. Определение максимального значения отрицательных чисел в массиве оформить в виде метода. Заполнение массивов и вывод на печать оформить в виде методов.

```

{
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(a[i] + " ");
    }
    Console.WriteLine();
}
static void Input(int[] a)
{
    Random rnd = new Random();
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-10, 10);
    }
}
static int NegativeNum(int[] a)
{
    int sum = 0;
    for(int i = 0; i < a.Length; i++)

```

```

    {
        if (a[i] < 0)
        {
            sum += a[i];
        }
    }
    return sum;
}
static void Min3(int x, int y, int z)
{
    if(x > y)
    {
        if(y > z)
        {
            Console.WriteLine($"Минимальное число отрицательных
элементов в массиве Z = {z}");
        }
        else
        {
            Console.WriteLine($"Минимальное число отрицательных
элементов в массиве Y = {y}");
        }
    }
    else if(x > z)
    {
        if (y > z)
        {
            Console.WriteLine($"Минимальное число отрицательных
элементов в массиве Z = {z}");

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		12

```

    }
    else
    {
        Console.WriteLine($"Минимальное число отрицательных
элементов в массиве Y = {y}");
    }
}
else
{
    Console.WriteLine($"Минимальное число отрицательных
элементов в массиве X = {x}");
}
}
static void Main(string[] args)
{
    int[] X = new int[8];
    int[] Y = new int[7];
    int[] Z = new int[9];
    Input(X);
    Input(Y);
    Input(Z);
    Print(X);
    Print(Y);
    Print(Z);
    Min3(NegativeNum(X), NegativeNum(Y), NegativeNum(Z));
    Console.ReadKey();
}

```

```
-6 -5 -3 -6 4 -1 8 -8
-6 -5 -3 -6 4 -1 8
-6 -5 -3 -6 4 -1 8 -8 -2
Минимальное число отрицательных элементов в массиве Z = -31
```

5. Заданы массивы A(15), B(7),C(10), состоящие из положительных и отрицательных чисел. Определить в каком массиве, больше четных чисел. Подсчет количества четных чисел оформить в виде метода. Заполнение массивов и вывод на печать оформить в виде методов.

```
static void Print(int[] a)
{
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(a[i] + " ");
    }
    Console.WriteLine();
}
static void Input(int[] a)
{
    Random rnd = new Random();
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-10, 10);
    }
}
static int Even(int[] a)
{
    int cnt = 0;
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] % 2 == 0)
```

```

        {
            cnt++;
        }
    }
    return cnt;
}

static void Max3(int a, int b, int c)
{
    if (a < b)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное число чётных чисел в
массиве C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное число чётных чисел в
массиве B = {b}");
        }
    }
    if (a < c)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное число чётных чисел в
массиве C = {c}");
        }
        else

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		15

```

        {
            Console.WriteLine($"Максимальное число чётных чисел в
массиве B = {b}");
        }
    }
else
    {
        Console.WriteLine($"Максимальное число чётных чисел в массиве
A = {a}");
    }
}
static void Main(string[] args)
{
    int[] A = new int[15];
    int[] B = new int[7];
    int[] C = new int[10];
    Input(A);
    Input(B);
    Input(C);
    Print(A);
    Print(B); ;
    Print(C);
    Max3(Even(A), Even(B), Even(C));
    Console.ReadKey();
}

```

```

5 5 -1 7 8 -1 -1 5 -5 5 -6 -8 3 -9 7
5 5 -1 7 8 -1 -1
5 5 -1 7 8 -1 -1 5 -5 5
Максимальное число чётных чисел в массиве A = 3

```

6. Заданы массивы A(6),B(7),C(14). Найти какой из массивов имеет наибольшее число отрицательных элементов. Нахождение количества отрицательных элементов массива оформить в виде метода. Заполнение массивов и вывод на печать оформить в виде методов.

```
static void Print(int[] a)
{
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(a[i] + " ");
    }
    Console.WriteLine();
}
static void Input(int[] a)
{
    Random rnd = new Random();
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-10, 10);
    }
}
static int NegativeCnt(int[] a)
{
    int cnt = 0;
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] < 0)
        {
            cnt++;
        }
    }
}
```

```

    }
    return cnt;
}
static void Max3(int a, int b, int c)
{
    if (a < b)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное число отрицательных чисел
в массиве C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное число отрицательных чисел
в массиве B = {b}");
        }
    }
    if (a < c)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное число отрицательных чисел
в массиве C = {c}");
        }
        else
        {
            Console.WriteLine($"Максимальное число отрицательных чисел
в массиве B = {b}");
        }
    }
}

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		18

```

    }
}
else
{
    Console.WriteLine($"Максимальное число отрицательных чисел в
массиве A = {a}");
}
}
static void Main(string[] args)
{
    int[] A = new int[6];
    int[] B = new int[7];
    int[] C = new int[14];
    Input(A);
    Input(B);
    Input(C);
    Print(A);
    Print(B); ;
    Print(C);
    Max3(NegativeCnt(A), NegativeCnt(B), NegativeCnt(C));
    Console.ReadKey();
}

```

```

0 1 4 -5 -7 -4
0 1 4 -5 -7 -4 -4
0 1 4 -5 -7 -4 -4 -3 3 3 2 -5 2 2
Максимальное число отрицательных чисел в массиве C = 6

```

7. Заданы массивы A(15), B(7),C(10), состоящие из положительных и отрицательных чисел. Определить в каком массиве, больше нечетных чисел. Подсчет количества нечетных чисел оформить в виде метода. Заполнение массивов и вывод на печать оформить в виде методов.

```
static void Print(int[] a)
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		19

```

{
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(a[i] + " ");
    }
    Console.WriteLine();
}

static void Input(int[] a)
{
    Random rnd = new Random();
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(-10, 10);
    }
}

static int NoEven(int[] a)
{
    int cnt = 0;
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] % 2 != 0)
        {
            cnt++;
        }
    }
    return cnt;
}

static void Max3(int a, int b, int c)
{

```

```

    if (a < b)
    {
        if (b < c)
        {
            Console.WriteLine($"Максимальное число чётных чисел в
массиве C = {c}");
        }
    }
    else
    {
        Console.WriteLine($"Максимальное число чётных чисел в
массиве B = {b}");
    }
}
if (a < c)
{
    if (b < c)
    {
        Console.WriteLine($"Максимальное число чётных чисел в
массиве C = {c}");
    }
    else
    {
        Console.WriteLine($"Максимальное число чётных чисел в
массиве B = {b}");
    }
}
else
{

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		21

```

        Console.WriteLine($"Максимальное число чётных чисел в массиве
A = {a}");
    }
}

static void Main(string[] args)
{
    int[] A = new int[15];
    int[] B = new int[7];
    int[] C = new int[10];
    Input(A);
    Input(B);
    Input(C);
    Print(A);
    Print(B); ;
    Print(C);
    Max3(NoEven(A), NoEven(B), NoEven(C));
    Console.ReadKey();
}

```

```

4 9 5 -4 -6 4 8 -5 -6 5 -8 7 -7 6 -1
4 9 5 -4 -6 4 8
4 9 5 -4 -6 4 8 -5 -6 5
Максимальное число чётных чисел в массиве A = 7

```

Лабораторная работа №10. Двумерные массивы

Цель: использование двумерных массивов для выполнения консольных приложений расчетного характера.

Задание 1.

Разработать консольное приложение в соответствии с условием задания. Задания выполнить с применением целочисленного двумерного массива. Размерность массива вводить с клавиатуры. Способ заполнения массива выбрать самостоятельно.

1) подсчитать среднее арифметическое элементов;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp3

{

internal class Program

{

private static double mid;

//Заполнение массива

static void Input(out int[,] a)

{

Random rnd = new Random();

Console.Write("n = "); //Кол-во строк

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

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		23

```

a = new int[n, n];
for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
{
for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
{
a[i, j] = rnd.Next(-100, 100);
}
}
}
//ВЫВОД МАССИВА
static void Print(int[,] a)
{
for (int i = 0; i < a.GetLength(0); i++)
{
for (int j = 0; j < a.GetLength(1); j++)
{
Console.Write("{0,5} ", a[i, j]);
}
Console.WriteLine(); //переход на новую строку
}
}
static void Cnt(ref int[,] a)
{
double huh = 0;
double sum = 0;
for (int i = 0; i < a.GetLength(0); i++)
{
for (int j = 0; j < a.GetLength(1); j++)
{

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		24

```

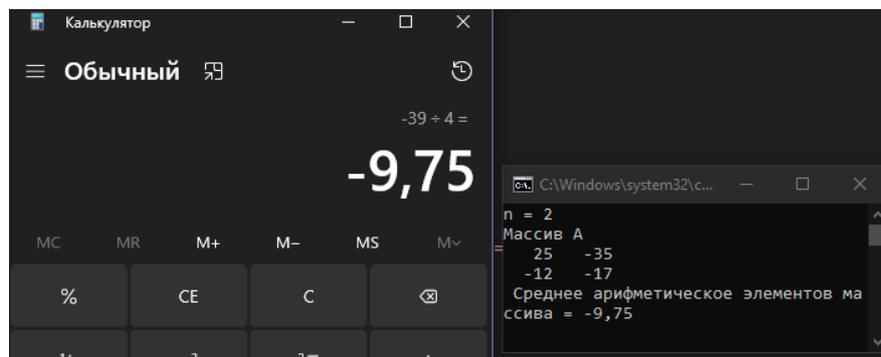
sum += a[i, j];
}
}
huh = sum / a.Length;

Console.WriteLine($" Среднее арифметическое элементов массива = {huh}");
}

static void Main()
{
int[,] A;

Input(out A);
Console.WriteLine("Массив A");
Print(A);
Cnt(ref A);
Console.ReadKey();
}
}
}

```



2) подсчитать среднее арифметическое отрицательных элементов;

```

namespace ConsoleApp3
{
    internal class Program
    {

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		25

```

private static double mid;

//Заполнение массива
static void Input(out int[,] a)
{
    Random rnd = new Random();
    Console.WriteLine("n = "); //Кол-во строк
    int n = int.Parse(Console.ReadLine());

    a = new int[n, n];
    for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
    {
        for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
        {
            a[i, j] = rnd.Next(-100, 100);
        }
    }
}

//Вывод массива
static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            Console.WriteLine("{0,5} ", a[i, j]);
        }
        Console.WriteLine(); //переход на новую строку
    }
}

static void Cnt(ref int[,] a)
{
    double huh = 0;
    double sum = 0;
    double jj = 0;
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] < 0)
            {
                sum += a[i, j];
                jj++;
            }
        }
    }
}

```

```

    }
}
huh = sum / jj;

Console.WriteLine($" Среднее арифметическое отрицательных
элементов массива = {huh}");
}

static void Main()
{
    int[,] A;

    Input(out A);
    Console.WriteLine("Массив A");
    Print(A);
    Cnt(ref A);
    Console.ReadKey();
}
}
}

```

```

n = 2
Массив A
-38    5
-90    34
Среднее арифметическое элементов массива = -64

```

3) подсчитать количество нечетных элементов;

```

namespace ConsoleApp3
{
    internal class Program
    {
        private static double mid;

        //Заполнение массива
        static void Input(out int[,] a)
        {
            Random rnd = new Random();
            Console.Write("n = "); //Кол-во строк

```

```

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

a = new int[n, n];
for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
{
    for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
    {
        a[i, j] = rnd.Next(-100, 100);
    }
}
}

//ВЫВОД МАССИВА
static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            Console.Write("{0,5} ", a[i, j]);
        }
        Console.WriteLine(); //переход на новую строку
    }
}

static void Cnt(ref int[,] a)
{
    double huh = 0;
    double sum = 0;
    int JJ = 0;
    for (int i = 0; i < a.GetLength(0); i++)

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		28

```

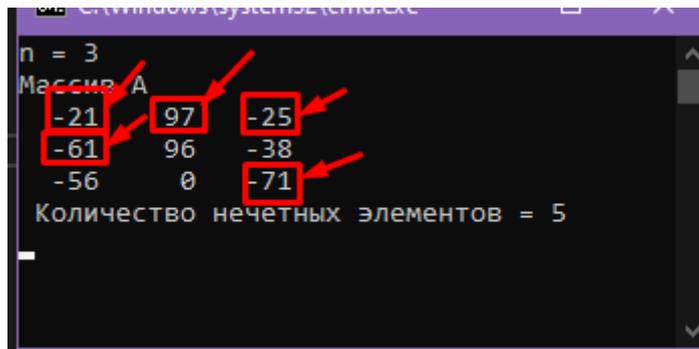
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] % 2 != 0)
            {
                sum +=a[i, j];
                JJ++;
            }
        }
    }
    // huh = sum / JJ;

    Console.WriteLine($" Количество нечетных элементов = {JJ}");
}

static void Main()
{
    int[,] A;

    Input(out A);
    Console.WriteLine("Массив A");
    Print(A);
    Cnt(ref A);
    Console.ReadKey();
}
}
}

```



4) подсчитать сумму элементов, кратных 9;

```
namespace ConsoleApp3
```

```
{
```

```
    internal class Program
```

```
    {
```

```
        private static double mid;
```

```
        //Заполнение массива
```

```
        static void Input(out int[,] a)
```

```
        {
```

```
            Random rnd = new Random();
```

```
            Console.Write("n = "); //Кол-во строк
```

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

```
            a = new int[n, n];
```

```
            for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
```

```
            {
```

```
                for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
```

```
                {
```

```
                    a[i, j] = rnd.Next(-100, 100);
```

```
                }
```

```
            }
```

```
        }
```

```
        //Вывод массива
```

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		30

```

static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            Console.Write("{0,5} ", a[i, j]);
        }
        Console.WriteLine(); //переход на новую строку
    }
}

static void Cnt(ref int[,] a)
{
    double huh = 0;
    double sum = 0;
    int JJ = 0;
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] % 9 == 0)
            {
                sum += a[i, j];
                JJ++;
            }
        }
    }
    // huh = sum / JJ;
}

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		31

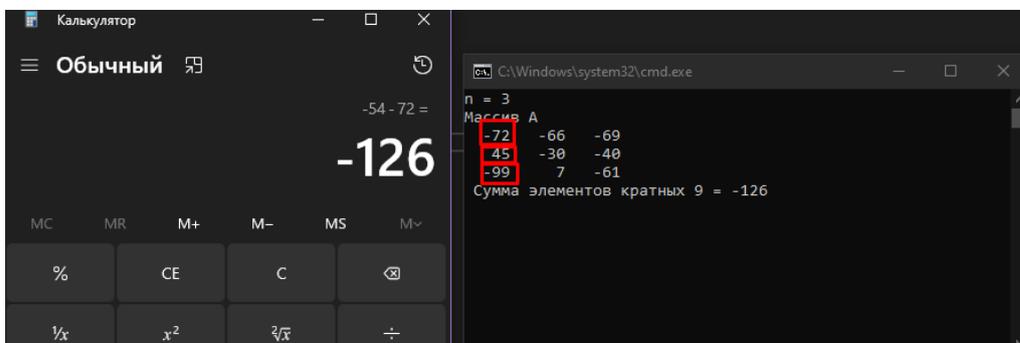
```

        Console.WriteLine($" Сумма квадратов четных элементов = {sum}");
    }

    static void Main()
    {
        int[,] A;

        Input(out A);
        Console.WriteLine("Массив A");
        Print(A);
        Cnt(ref A);
        Console.ReadKey();
    }
}
}
}

```



5) подсчитать сумму квадратов четных элементов;

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp3
{
    internal class Program

```

```

{
//Заполнение массива
static void Input(out int[,] a)
{
    Random rnd = new Random();
    Console.WriteLine("n = "); //Кол-во строк
    int n = int.Parse(Console.ReadLine());

    a = new int[n, n];
    for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
    {
        for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
        {
            a[i, j] = rnd.Next(-100, 100);
        }
    }
}
//Вывод массива
static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            Console.WriteLine("{0,5} ", a[i, j]);
        }
        Console.WriteLine(); //переход на новую строку
    }
}
static void Cnt(ref int[,] a)
{
    double sum = 0;
    int JJ = 0;
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] % 2 == 0)
            {
                sum += Math.Pow(a[i, j], 2);
                JJ++;
            }
        }
    }
}
}

```

```

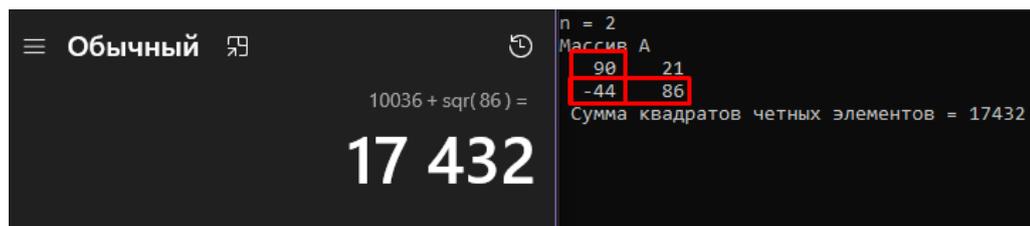
    }
    // huh = sum / JJ;

    Console.WriteLine($" Сумма квадратов четных элементов = {sum}");
}

static void Main()
{
    int[,] A;

    Input(out A);
    Console.WriteLine("Массив A");
    Print(A);
    Cnt(ref A);
    Console.ReadKey();
}
}
}

```



б) вывести на экран номера всех элементов, больших заданного числа;

```

namespace ConsoleApp3
{
    internal class Program
    {
        private static double mid;

        //Заполнение массива
        static void Input(out int[,] a)
        {
            Random rnd = new Random();
            Console.Write("n = "); //Кол-во строк

```

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		34

```

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

a = new int[n, n];
for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
{
    for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
    {
        a[i, j] = rnd.Next(-100, 100);
    }
}

//ВЫВОД МАССИВА
static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            Console.Write("{0,5} ", a[i, j]);
        }
        Console.WriteLine(); //переход на новую строку
    }
}

static void Cnt(ref int[,] a)

{
    Console.WriteLine("Введите число");
    double d = double.Parse(Console.ReadLine());
    double huh = 0;
}

```

```

double sum = 0;

int JJ = 0;

for (int i = 0; i < a.GetLength(0); i++)
{
    for (int j = 0; j < a.GetLength(1); j++)
    {
        if (a[i, j] > d)
        {
            Console.WriteLine($"Номера этих элементов {i}, {j}");
        }
    }
}

// huh = sum / JJ;

//Console.WriteLine($" Количество нечетных элементов = {JJ}");
}

static void Main()
{
    int[,] A;

    Input(out A);
    Console.WriteLine("Массив A");
    Print(A);
    Cnt(ref A);
    Console.ReadKey();
}
}

```

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		36


```

for (int i = 0; i < a.GetLength(0); i++)
{
    for (int j = 0; j < a.GetLength(1); j++)
    {
        Console.Write("{0,5} ", a[i, j]);
    }
    Console.WriteLine(); //переход на новую строку
}
}
static void Cnt(ref int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] % 2 != 0)
            {
                Console.WriteLine($"Номера этих элементов {i}, {j}");
            }
        }
    }
}
// huh = sum / JJ;

//Console.WriteLine($" Количество нечетных элементов = {JJ}");
}

static void Main()
{
    int[,] A;

    Input(out A);
    Console.WriteLine("Массив A");
    Print(A);
    Cnt(ref A);
    Console.ReadKey();
}
}
}

```

```

n = 3
Массив A
 97 -62 -81
-92 -92 47
-100 -97 -20
Номера этих элементов 0,0
Номера этих элементов 0,2
Номера этих элементов 1,2
Номера этих элементов 2,1

```

8) вывести на экран положительные элементы с нечетными индексами.

internal class Program

```

{
    private static double mid;

    //Заполнение массива
    static void Input(out int[,] a)
    {
        Random rnd = new Random();
        Console.WriteLine("n = "); //Кол-во строк
        int n = int.Parse(Console.ReadLine());

        a = new int[n, n];
        for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
        {
            for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
            {
                a[i, j] = rnd.Next(-100, 100);
            }
        }
    }

    //Вывод массива
    static void Print(int[,] a)
    {
        for (int i = 0; i < a.GetLength(0); i++)
        {
            for (int j = 0; j < a.GetLength(1); j++)
            {
                Console.WriteLine("{0,5} ", a[i, j]);
            }
            Console.WriteLine(); //переход на новую строку
        }
    }

    static void Cnt(ref int[,] a)

```


Для заданного массива A размером n*n:

- 1) подсчитать среднее арифметическое нечетных элементов расположенных выше главной диагонали;

```
static void Input(out int[,] a)
{
    Random rnd = new Random();
    Console.WriteLine("n = "); //Кол-во строк
    int n = int.Parse(Console.ReadLine());

    a = new int[n, n];
    for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
    {
        for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
        {
            a[i, j] = rnd.Next(-100, 100);
        }
    }
    double huh = 0;
    double sum = 0;
    int JJ = 0;
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] % 2 != 0 && j > i)
            {
                sum += a[i, j];
                JJ++;
            }
        }
    }
}
```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		41

```

        }
    }
    Console.WriteLine();
    huh = sum / JJ;

    Console.WriteLine($" Ср. арифмет. эл. выше главной диагонали
= {huh}");
}
//ВЫВОД МАССИВА
static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            Console.Write("{0,5} ", a[i, j]);
        }
        Console.WriteLine(); //переход на новую строку
    }
}
static void Cnt(string[] args)
{
    int sum = 0;

    Console.WriteLine("sum=" + sum);
}
static void Cnt(ref int[,] a)
{

```

```

    }

    static void Main()
    {
        int[,] A;

        Input(out A);
        Console.WriteLine("Массив A");
        Print(A);
        Cnt(ref A);
        Console.ReadKey();
    }
}

```

2) подсчитать среднее арифметическое четных элементов расположенных ниже главной диагонали;

```

internal class Program
{

    //Заполнение массива
    static void Input(out int[,] a)
    {
        Random rnd = new Random();
        Console.Write("n = "); //Кол-во строк
        int n = int.Parse(Console.ReadLine());
    }
}

```

```

a = new int[n, n];
for (int i = 0; i < a.GetLength(0); i++) //кол-во в строке
{
    for (int j = 0; j < a.GetLength(1); j++) //кол-во столбце
    {
        a[i, j] = rnd.Next(-100, 100);
    }
}
double huh = 0;
double sum = 0;
int JJ = 0;
for (int i = 0; i <a.GetLength(0); i++)
{
    for (int j = 0; j < a.GetLength(1); j++)
    {
        if (a[i, j] % 2 == 0 && j<i)
        {
            sum += a[i, j];
            JJ++;
        }
    }
}
Console.WriteLine();
huh = sum / JJ;

Console.WriteLine($" Ср. арифмет. эл. выше главной диагонали
= {huh}");

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		44

```

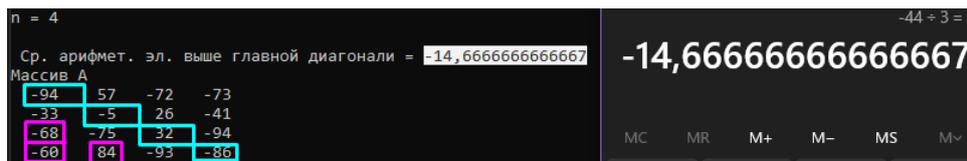
    }
    //ВЫВОД МАССИВА
    static void Print(int[,] a)
    {
        for (int i = 0; i < a.GetLength(0); i++)
        {
            for (int j = 0; j < a.GetLength(1); j++)
            {
                Console.Write("{0,5} ", a[i, j]);
            }
            Console.WriteLine(); //переход на новую строку
        }
    }
    static void Cnt(ref int[,] a)
    {
    }

    static void Main()
    {
        int[,] A;

        Input(out A);
        Console.WriteLine("Массив A");
        Print(A);
        Cnt(ref A);
        Console.ReadKey();
    }
}

```

}



- 3) подсчитать сумму элементов расположенных на побочной диагонали
- 4) подсчитать среднее арифметическое ненулевых элементов расположенных над побочной диагональю;
- 5) подсчитать среднее арифметическое элементов расположенных под побочной диагональю;
- 6) поменять местами столбцы по правилу первый с последним второй с предпоследним и т.д.;
- 7) поменять местами две средних строки если количество строк четное и первую со средней строкой, если количество строк нечетное;

Задание 3

Для заданного массива A размером $n \times n$, элементы которого целые числа,

- 1) подсчитать среднее арифметическое нечетных элементов, расположенных выше главной диагонали;
- 2) подсчитать среднее арифметическое четных элементов, расположенных ниже главной диагонали;
- 3) подсчитать сумму элементов, расположенных на побочной диагонали;
- 4) подсчитать среднее арифметическое ненулевых элементов, расположенных над побочной диагональю;
- 5) подсчитать среднее арифметическое элементов, расположенных под побочной диагональю;
- 6) поменять местами столбцы по правилу: первый с последним, второй с предпоследним и т.д.;
- 7) поменять местами две средних строки, если количество строк четное, и первую со средней строкой, если количество строк нечетное;
- 8) поменять местами два средних столбца, если количество столбцов четное, и первый со средним столбцом, если количество столбцов нечетное;
- 9) если количество строк в массиве четное, то поменять строки местами по правилу: первую строку со второй, третью – с четвертой и т.д.; если количество строк в массиве нечетное, то оставить массив без изменений.
- 10) если количество столбцов в массиве четное, то поменять столбцы местами по правилу: первый столбец со вторым, третий – с четвертым и т.д.; если количество столбцов в массиве нечетное, то оставить массив без изменений.

```

internal class Program
{
    static Random rand = new Random();

    static int[,] formirov2(int[,] mass, int n, int m)
    {
        {
            for (int i = 0; i < mass.GetLength(0); i++)
            {
                for (int j = 0; j < mass.GetLength(1); j++)
                {
                    mass[i, j] = rand.Next(100);
                    Console.WriteLine("{0,4}", mass[i, j]);
                }
                Console.WriteLine();
            }
            return mass;
        }
    }

    static void SRarf1(int[,] mass)
    {
        int sum = 0;
        int numb = 0;
        for (int i = 0; i < mass.GetLength(0); i++)
        {
            for (int j = 0; j < mass.GetLength(1); j++)
            {
                if (i > j)
                {
                    if (mass[i, j] % 2 != 0)
                    {
                        numb = numb + 1;
                        sum = sum + mass[i, j];
                    }
                }
            }
        }
    }
}

```

```

    }
    int sraf = sum / numb;
    Console.WriteLine($"1) среднее арифмитическое нечётных выше
главной диагонали = {sraf}");
}

```

```

static void SRarf2(int[,] mass)
{
    int sum = 0;
    int numb = 0;
    for (int i = 0; i < mass.GetLength(0); i++)
    {
        for (int j = 0; j < mass.GetLength(1); j++)
        {
            if (i < j)
            {
                if (mass[i, j] % 2 == 0)
                {
                    numb = numb + 1;
                    sum = sum + mass[i, j];
                }
            }
        }
    }
}

```

```

    }
    int sraf = sum / numb;
    Console.WriteLine($"2) среднее арифмитическое чётных ниже
главной диагонали = {sraf}");
}

```

```

static void SummOnAditional(int[,] mass)
{
    int sum = 0;
    int test = mass.GetLength(0) - 1;
    for (int i = 0; i < mass.GetLength(0); i++)
    {
        for (int j = 0; j < mass.GetLength(1); j++)
        {
            if (test == i + j)
            {

```

```

        sum = sum + mass[i, j];

    }

}

Console.WriteLine($"3) сумма на побочной диагонали = {sum}");
}

static void SummUpAditional(int[,] mass)
{
    int sum = 0;
    int numb = 0;
    int test = mass.GetLength(0) - 1;
    for (int i = 0; i < mass.GetLength(0); i++)
    {
        for (int j = 0; j < mass.GetLength(1); j++)
        {
            if (test > i + j)
            {
                if (mass[i, j] != 0)
                {
                    numb = numb + 1;
                    sum = sum + mass[i, j];
                }
            }
        }
    }

    int srarf = sum / numb;
    Console.WriteLine($"4) среднее арифметическое над побочной
диагональю = {sraf}");
}

static void SummDownAditional(int[,] mass)
{

```

```

int sum = 0;
int numb = 0;
int test = mass.GetLength(0) - 1;
for (int i = 0; i < mass.GetLength(0); i++)
{
    for (int j = 0; j < mass.GetLength(1); j++)
    {
        if (test < i + j)
        {
            numb = numb + 1;
            sum = sum + mass[i, j];
        }
    }
}
int srarf = sum / numb;
Console.WriteLine($"5) среднее арифметическое под побочной
диагональю = {srarf}");
}

```

```

static void Transform(ref int[,] a)
{
    int n = a.GetLength(0);
    int buff;
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j <= a.GetLength(1) / 2; j++)
        {
            buff = a[i, j];
            a[i, j] = a[i, n - j - 1];
            a[i, n - j - 1] = buff;
        }
    }
}
static void Print(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {

```

```

        Console.WriteLine("{0,5} ", a[i, j]);
    }
    Console.WriteLine();
}
}

static void Transform2 (int[,] a) // 7 number
{
    int row1;
    int row2;
    if (a.GetLength(0) % 2 == 0)
    {
        row1 = a.GetLength(0) / 2;
        row2 = row1 - 1;
    }
    else
    {
        row1 = a.GetLength(0) / 2;
        row2 = 0;
    }
    for (int i = 0; i < a.GetLength(1); i++)
    {
        int tmp = a[row1, i];
        a[row1, i] = a[row2, i];
        a[row2, i] = tmp;
    }
}

static void Transform3(int[,] a) // 8 number
{
    int col1;
    int col2;
    if (a.GetLength(0) % 2 == 0)
    {
        col1 = a.GetLength(1) / 2;
        col2 = col1 - 1;
    }
    else
    {
        col1 = a.GetLength(1) / 2;
        col2 = 0;
    }
}

```



```
Console.WriteLine("9) Массив А - transform");  
Print(array1);
```

```
Console.WriteLine("-----  
-----");
```

```
Console.ReadKey();
```

```
    }  
  }  
}
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		54

Введите высоту массива: 5

Введите длину массива: 5

```
24 88 46 82 94
99 27 22 40 0
5 46 40 16 3
8 94 76 37 37
27 45 5 52 67
```

1) среднее арифметическое нечётных выше главной диагонали = 36

2) среднее арифметическое чётных ниже главной диагонали = 48

3) сумма на побочной диагонали = 295

4) среднее арифметическое над побочной диагональю = 44

5) среднее арифметическое под побочной диагональю = 33

6) Массив A - transform

```
94 82 46 88 24
0 40 22 27 99
3 16 40 46 5
37 37 76 94 8
67 52 5 45 27
```

7) Массив A - transform

```
3 16 40 46 5
0 40 22 27 99
94 82 46 88 24
37 37 76 94 8
67 52 5 45 27
```

8) Массив A - transform

```
40 16 3 46 5
22 40 0 27 99
46 82 94 88 24
76 37 37 94 8
5 52 67 45 27
```

9) Массив A - transform

```
40 16 3 46 5
22 40 0 27 99
46 82 94 88 24
```

Изм.	Лис	№ докум.	Подпись	Дат

ККОО.ОАИП4211121.000

Лис

55

Задание 3. Для заданного массива размером $n \times n$, элементы которого являются целыми числами:

- 1) найти максимальный элемент в каждой строке и записать данные в новый массив;
- 2) найти минимальный элемент в каждом столбце и записать данные в новый массив;
- 3) четные столбцы таблицы заменить на вектор X;
- 4) нечетные строки таблицы заменить на вектор X;
- 5) поменять местами элементы главной и побочной диагонали;
- 6) для каждой строки подсчитать количество положительных элементов и записать данные в новый массив;
- 7) для каждого столбца подсчитать сумму отрицательных элементов и записать данные в новый массив;
- 8) для каждого столбца подсчитать сумму четных положительных элементов и записать данные в новый массив;
- 9) для каждой строки подсчитать количество элементов, больших заданного числа, и записать данные в новый массив;
- 10) для каждого столбца найти первый положительный элемент и записать данные в новый массив.

internal class Program

```
{
    static Random rand = new Random();

    static int[,] formirov2(int[,] mass, int n, int m)
    {
        {
            for (int i = 0; i < mass.GetLength(0); i++)
            {
                for (int j = 0; j < mass.GetLength(1); j++)
                {
                    mass[i, j] = rand.Next(100);
                    Console.WriteLine("{0,4}", mass[i, j]);
                }
                Console.WriteLine();
            }
            return mass;
        }
    }
}
```

```

static int[,] Max (int[,] mass)
{
    {
        int max = 0;

        for (int i = 0; i < mass.GetLength(0); i++)
        {
            for (int j = 0; j < mass.GetLength(1); j++)
            {
                if (max < mass[i, j])
                {
                    max = mass[i, j];
                }
            }
        }
        Console.WriteLine( max);
        max = 0;
    }
    return mass;
}
}

```

```

static int[,] Min(int[,] mass)
{
    {
        int min = 9999;

        for (int i = 0; i < mass.GetLength(0); i++)
        {
            for (int j = 0; j < mass.GetLength(1); j++)
            {
                if (min > mass[i, j])
                {
                    min = mass[i, j];
                }
            }
        }
        Console.WriteLine(min);
        min = 9999;
    }
}

```

```
        return mass;
    }
}
```

```
static void Main(string[] args)
{
    Console.Write("Введите размерность массива: ");
    int n = int.Parse(Console.ReadLine());
    int[,] array1 = new int[n, n];
    array1 = formirov2(array1, n, n);
    Console.WriteLine("");
    Max(array1);
    Console.WriteLine("");
    Min(array1);
    Console.WriteLine("");

    Console.ReadKey();
}
}
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		58

Контрольная работа. Двумерные массивы.

Критерии оценивания:

решено одно задание – «удовлетворительно»;

решено два задания – «хорошо»;

решено три задания – «отлично».

Задание 1.

Дан двумерный массив. Размерность массива вводится с клавиатуры. Элементами массива являются целые числа. Для заданной последовательности целых чисел:

1) заменить все положительные элементы противоположными им числами;

```
internal class Program
```

```
{
    static void Print(int[,] a)
    {
        for (int i = 0; i < a.GetLength(0); i++)
        {
            for (int j = 0; j < a.GetLength(1); j++)
            {
                Console.Write("{0,5} ", a[i, j]);
            }
            Console.WriteLine();
        }
    }
    static void Input(out int[,] a)
    {
        Console.Write("n = ");
        int n = int.Parse(Console.ReadLine());
        Console.Write("m = ");
```

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		59

```

int m = int.Parse(Console.ReadLine());
Random rnd = new Random((int)DateTime.Now.Ticks);
a = new int[n, m];
for (int i = 0; i < a.GetLength(0); i++)
{
    for (int j = 0; j < a.GetLength(1); j++)
    {
        a[i, j] = rnd.Next(-100, 100);
    }
}
static void Pol(int[,] a)
{
    for (int i = 0; i < a.GetLength(0); i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            if (a[i, j] > 0) a[i, j] *= -1;
        }
    }
    Console.WriteLine("Изменённый массив: ");
    Print(a);
}
static void Main(string[] args)
{
    int[,] A;
    Input(out A);
    Console.WriteLine("Массив A:");
    Print(A);
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		60

```

Console.WriteLine();
Pol(A);
Console.ReadKey();
}
}

```

```

n = 4
m = 5
Массив A:
91 81 -34 18 67
-85 94 -63 1 -11
96 -2 -66 -86 86
23 86 29 -89 -51

Изменённый массив:
-91 -81 -34 -18 -67
-85 -94 -63 -1 -11
-96 -2 -66 -86 -86
-23 -86 -29 -89 -51

```

Задание 2.

Дан двумерный массив. Размерность массива вводится с клавиатуры. Элементами массива являются целые числа. Для заданного массива $n \times m$:

7) поменять местами две средних строки если количество строк четное и первую со средней строкой, если количество строк четное и первую со средней строкой, если количество строк нечетное;

```

n = 6
Массив A:
9 -75 -94 1 59 41
-100 -66 -22 34 46 40
-40 76 98 -66 95 -71
29 -79 59 67 -93 30
27 29 91 -65 20 -56
-78 -93 57 -94 82 -62

Изменённый массив:
9 -75 -94 1 59 41
-100 -66 -22 34 46 40
29 -79 59 67 -93 30
-40 76 98 -66 95 -71
27 29 91 -65 20 -56
-78 -93 57 -94 82 -62

```

Лабораторная работа №11. Методы сортировки массива

Постановка задачи. Для решения многих задач необходимо упорядочить данные по определенному признаку. Процесс упорядочения заданного множества объектов по заданному признаку называется сортировкой. Для простоты изложения рассматривается одномерный массив из целых чисел. Суть большинства алгоритмов сортировки от такого упрощения не изменяется. Алгоритмы сортировки отличаются друг от друга степенью эффективности, под которой понимается количество сравнений и количество обменов, произведенных в процессе сортировки. В основном мы будем оценивать эффективность количеством операций сравнения (порядком этого значения). Заметим, что элементы массива можно сортировать:

по возрастанию – каждый следующий элемент больше предыдущего;

по неубыванию – каждый следующий элемент не меньше предыдущего, то есть больше или равен;

по убыванию – каждый следующий элемент меньше предыдущего;

по невозрастанию – каждый следующий элемент не больше предыдущего, то есть меньше или равен.

Научившись выполнять одну сортировку, изменить ее, чтобы получить другую, не составляет особого труда.

Задание 1.

Ознакомьтесь с алгоритмами сортировки простого выбора, простого обмена, простой вставки.

Задание 2.

Запрограммируйте эти методы, стараясь не использовать подсказки. Ниже приведены программные коды, реализующие эти методы.

Комментарий.

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		62

Класс предоставляет набор методов и свойств, которые можно использовать для точного измерения затраченного времени.

Метод Start() запускает таймер, который измеряет время выполнения

Задание 3.

Проведите испытание. Задайте количество элементов в массиве 100,1000,10000,100000 и сравните время выполнения алгоритма. Какой алгоритм эффективнее? Для ответа на этот вопрос заполните таблицу (* заменить на полученные данные)

n метод	100	1000	10000	100000
Метод простого выбора	0мс 0 сек 243 тик	1мс 0 сек 19971 тик	201мс 0сек 2018233тик	19256мс 19 сек 192566079тик
Метод простых обменов	0мс 0 сек 229 тик	2мс 0 сек 28525 тик	293мс 0 сек 2939699 тик	27185мс 27 сек 271852247 тик
Метод простых вставок	0мс 0 сек 110 тик	0мс 0 сек 7276 тик	74мс 0 сек 741018 тик	7561мс 7 сек 75613789 тик

Задание 4. Как изменятся алгоритмы, если сортировать массив надо по убыванию.

```
static void Main(string[] args)
{
    int n = 5;
    int[] a = new int[n];
    Random r = new Random();
```

```

for (int i = 0; i < n; i++)
{
    a[i] = r.Next(0, 10);
    Console.Write(a[i] + " ");
}
Stopwatch watch = new Stopwatch();
watch.Start();
for (int i = 1; i < n; i++)
{
    int w = a[i];
    int j = i - 1;
    while (j >= 0 && w > a[j])
    {
        a[j + 1] = a[j];
        j--;
    }
    a[j + 1] = w;
}
watch.Stop();
Console.WriteLine();
for (int i = 0; i < n; i++) Console.Write(a[i] + " ");
Console.WriteLine();
Console.WriteLine("Метод простого обмена");
Console.WriteLine("Сортировка массива из {0} элементов", n);
Console.WriteLine("Время выполнения программы в миллисекундах:"
+ watch.ElapsedMilliseconds + "мс");
Console.WriteLine("Время выполнения программы в секундах:" +
watch.Elapsed.Seconds + "сек");

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		64

```

        Console.WriteLine("Время выполнения программы в тиках:" +
watch.ElapsedTicks);
        Console.ReadLine();
    }

```

```

1 5 3 6 3
6 5 3 3 1
Метод простого обмена
Сортировка массива из 5 элементов
Время выполнения программы в миллисекундах:0мс
Время выполнения программы в секундах:0сек
Время выполнения программы в тиках:19

```

`internal class Program`

```

    {
    static void Main(string[] args)
    {
    int n = 4;
    int[] a = new int[n];
        Random r = new Random();
    for (int i = 0; i < n; i++)
    {
        a[i] = r.Next(0, 7);
    Console.WriteLine(a[i] + " ");
    }
        Stopwatch watch = new Stopwatch();
    watch.Start();
    for (int i = 0; i < n - 1; i++)
    {
    int w;
    for (int j = 0; j < n - i - 1; j++)
    {
    if (a[i] <= a[j+1])
    {

```

```

        w = a[j];
a[j] = a[j + 1]; a[j + 1] = w;
    }
}
}
watch.Stop();
Console.WriteLine();
for (inti = 0; i < n; i++) Console.Write(a[i] + " ");
Console.WriteLine();
Console.WriteLine("Метод простого выбора");
Console.WriteLine("Сортировка массива из {0} элементов", n);
Console.WriteLine("Время выполнения программы в миллесекундах: " +
watch.ElapsedMilliseconds + "мс");
Console.WriteLine("Время выполнения программы в секундах: " +
watch.Elapsed.Seconds + "сек");
Console.WriteLine("Время выполнения программы в тиках: " +
watch.ElapsedTicks);
Console.ReadLine();
}
}
}

```

```

1 1 0 2
2 1 1 0
Метод простого выбора
Сортировка массива из 4 элементов
Время выполнения программы в миллесекундах: 0мс
Время выполнения программы в секундах: 0сек
Время выполнения программы в тиках: 21

```

Задание 5.

Известны значения роста 25 учащихся класса, заданные в алфавитном порядке фамилий. Определить рост учащегося, который при построении учащихся по росту в порядке возрастания занимал бы 10-е место при счете от самого высокого ученика.

```
static void Print(int[] a)
{
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(" {0,3} см |", a[i]);
    }
}

static int[] Input()
{
    Random rnd = new Random();
    Console.Write("n=");
    int n = int.Parse(Console.ReadLine());
    int[] a = new int[n];
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(150, 190);
    }
    return a;
}

static void Sort(int[] a)
{
    int temp;
    for (int i = 0; i < a.Length - 1; i++)
    {
```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		67

```

        for (int j = a.Length - 1; j > i; j--)
        {
            if (a[j] < a[j - 1])
            {
                temp = a[j];
                a[j] = a[j - 1];
                a[j - 1] = temp;
            }
        }
    }
}

static void Vibor(int[] a)
{
    int temp1;
    for (int i = 0; i < a.Length; i++)
    {
        for (int j = 0; j < a.Length; j++)
        {
            temp1 = a[j];
        }
    }
    Console.Write(a[15] + "см");
}

static void Main(string[] args)
{
    int[] Mas = Input();
    Console.Write("Начальный:");
    Print(Mas);
    Console.WriteLine();
}

```

```

Sort(Mas);
Console.WriteLine("Отсортированный массив:");
Print(Mas);
Console.WriteLine();
Console.WriteLine("Высокий на 10-ом месте :");

Vibor(Mas);
Console.ReadKey();
}
}

```

```

n=25
Начальный: 155 см | 179 см | 181 см | 166 см | 179 см | 187 см | 160 см | 165 см | 168 см | 157 см |
160 см | 158 см | 183 см | 162 см | 173 см | 156 см | 181 см | 165 см | 181 см | 186 см | 179 см
| 189 см | 167 см | 165 см | 154 см
Отсортированный массив: 154 см | 155 см | 156 см | 157 см | 158 см | 160 см | 160 см | 162 см | 165 см
| 165 см | 165 см | 166 см | 167 см | 168 см | 173 см | 179 см | 179 см | 179 см | 181 см | 181 см
| 181 см | 183 см | 186 см | 187 см | 189 см
Высокий на 10-ом месте :179см

```

Задание 6.

Известны значения максимальной скорости 15 моделей автомобилей. Определить максимальную скорость автомобиля, являющегося "шестым самым быстрым автомобилем".

```

namespace ConsoleApp1
{
    internal class Program
    {
        static void Print(int[] a)
        {
            for (int i = 0; i < a.Length; i++)
            {
                Console.WriteLine("{0,4} см", a[i]);
            }
        }
        static int[] Input()

```

```

{
    Random rnd = new Random();
    Console.WriteLine("n=");
    int n = int.Parse(Console.ReadLine());
    int[] a = new int[n];
    for (int i = 0; i < a.Length; i++)
    {
        a[i] = rnd.Next(150, 190);
    }
    return a;
}

static void Sort(int[] a)
{
    int temp;
    for (int i = 0; i < a.Length - 1; i++)
    {
        for (int j = a.Length - 1; j > i; j--)
        {
            if (a[j] < a[j - 1])
            {
                temp = a[j];
                a[j] = a[j - 1];
                a[j - 1] = temp;
            }
        }
    }
}

static void Hot (int[] a)
{

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		70

```

int temp1;
for (int i = 0; i < a.Length; i++)
{
    for (int j = 0; j < a.Length; j++)
    {
        temp1 = a[j];
    }
}
Console.Write(a[14] + "см");
}
static void Main(string[] args)
{
    int[] Mas = Input();
    Console.Write("Начальный:");
    Print(Mas);
    Console.WriteLine();
    Sort(Mas);
    Console.Write("Отсортированный массив:");
    Print(Mas);
    Console.WriteLine();
    Console.Write("Рост :");

    Hot (Mas);
}
}
}
}

```

```

C:\Windows\system32\cmd.exe
n=25
Начальный: 189 см 167 см 177 см 172 см 153 см 176 см 156 см 175 см 179 см 179 см 151 см 163 см 164 см 161 см 152 см 167
см 177 см 186 см 165 см 189 см 163 см 182 см 152 см 181 см 155 см
Отсортированный массив: 151 см 152 см 152 см 153 см 155 см 156 см 161 см 163 см 163 см 164 см 165 см 167 см 167 см 172
см 175 см 176 см 177 см 177 см 179 см 179 см 181 см 182 см 186 см 189 см 189 см
Рост :175смДля продолжения нажмите любую клавишу . . .

```

Контрольная работа. Применение методов сортировки

Задание 1

```
internal class Program
{
    static void Print(int[] a)
    {
        for (int i = 0; i < a.Length; i++)
        {
            Console.Write("{0,5} кг", a[i]);
        }
    }
    static int[] Input()
    {
        Random rnd = new Random();
        Console.Write("n=");
        int n = int.Parse(Console.ReadLine());
        int[] a = new int[n];
        for (int i = 0; i < a.Length; i++)
        {
            a[i] = rnd.Next(110, 190);
        }
        return a;
    }
    static void Sort(int[] a)
    {
        int temp;
        for (int i = 0; i < a.Length - 1; i++)
        {
            for (int j = a.Length - 1; j > i; j--)
            {
                if (a[j] < a[j - 1])
                {
                    temp = a[j];
                    a[j] = a[j - 1];
                    a[j - 1] = temp;
                }
            }
        }
    }
    static void Hot(int[] a)
    {
        int temp1;
        for (int i = 0; i > a.Length; i++)
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		72

```

    {
        for (int j = 0; j < a.Length; j++)
        {
            temp1 = a[j];
        }
    }
    Console.WriteLine(a[4] + "кг");
}
static void Main(string[] args)
{
    int[] Mas = Input();
    Console.WriteLine("Начальный:");
    Print(Mas);
    Console.WriteLine();
    Sort(Mas);
    Console.WriteLine("Отсортированный массив:");
    Print(Mas);
    Console.WriteLine();
    Console.WriteLine("Самый легкий :");

    Hot(Mas);
}
}

```

```

n=20
Начальный: 133 кг 140 кг 139 кг 175 кг 122 кг 159 кг 183 кг
кг 138 кг 152 кг 116 кг 171 кг 111 кг 189 кг
Отсортированный массив: 111 кг 115 кг 116 кг 122 кг 124 кг 13
152 кг 159 кг 169 кг 169 кг 171 кг 175 кг 183 кг 189 кг
Самый легкий :124кгДля продолжения нажмите любую клавишу . . .

```

Лабораторная работа №12. Строки

Цель: работа со строками и методами для сравнения, поиска, сортировки и управления строковыми значениями

При решении задач использовать тип string.

Класс string обеспечивает множество встроенных методов для сравнения, поиска и

управления строковыми значениями. Вот неполный список всех возможностей этого

Empty — свойство, определяющее, пустая ли строка;

Compare() — функция сравнения двух строк;

CompareOrdinal() — сравнивает строки в независимости от региональных настроек;

Concat() — создает новую строку из двух и более исходных строк;

CopyO — создает дубликат исходной строки;

Equals() — определяет, содержат ли две строки одинаковые значения;

Format() — форматирует строку, используя строго заданный формат;

InternO — возвращает ссылку на существующий экземпляр строки;

Join() — добавляет новую строку в любое место уже существующей строки;

Chars — индекатор символов строки;

Length — количество символов в строке;

CloneO — возвращает ссылку на существующую строку;

CompareToO — сравнивает одну строку с другой;

CopyToQ — копирует определенное число символов строки в массив

Unicode

определяет,

заканчивается

последовательностью символов;

EqualsQ — определяет, имеют ли две строки одинаковые значения;

InsertO — вставляет новую строку в уже существующую;

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		74

3) выводит на экран символы, которые наиболее часто встречаются в строке;

4) выводит на экран символы, которые встречаются в строке только один раз;

5) определяет, имеются ли в строке два соседствующих одинаковых символа;

6) определяет, является ли строка палиндромом;

```
static void Main(string[] args)
```

```
{
```

```
    Console.WriteLine("Введите слово для проверки");
```

```
    string input = Console.ReadLine();
```

```
    char[] obrtext = input.ToCharArray();
```

```
    Array.Reverse(obrtext);
```

```
    string finaltext = new string(obrtext);
```

```
    if (finaltext == input)
```

```
    {
```

```
        Console.WriteLine("Слово {0} является палиндромом", input);
```

```
    }
```

```
    else
```

```
    {
```

```
        Console.WriteLine("Слово {0} не является палиндромом",
```

```
input);
```

```
    }
```

```
    Console.ReadKey();
```

```
}
```

```
Введите слово для проверки
```

```
шалаш
```

```
Слово шалаш является палиндромом
```

```
Введите слово для проверки
```

```
лоал
```

```
Слово лоал не является палиндромом
```

7) определяет, упорядочены ли по алфавиту символы строки;

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		<i>77</i>

```

static void Main(string[] args)
{
    Console.WriteLine("Введите буквы для проверки");
    string input = Console.ReadLine();
    bool FT = true;
    char[] nonesorttext = input.ToCharArray();
    char[] sorttext = input.ToCharArray();
    Array.Sort(sorttext);
    for (int i = 0; i < sorttext.Length; i++)
    {
        if(sorttext[i] == nonesorttext[i])
        {
            FT = true;
        }
        else
        {
            FT = false;
        }
    }
    if (FT == true)
    {
        Console.WriteLine("Строка упорядочена в алфавитном
порядке");
    }
    else
    {
        Console.WriteLine("Строка не упорядочена в алфавитном
порядке");
    }
}

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		78

```
Console.ReadKey();  
}
```

```
Введите буквы для проверки  
abc  
Строка упорядочена в алфавитном порядке
```

```
Введите буквы для проверки  
вба  
Строка не упорядочена в алфавитном порядке
```

8) подсчитывает количество букв в строке;

```
static void Main(string[] args)  
{  
    Console.WriteLine("Введите строку");  
    string input = Console.ReadLine();  
    int ind = 0;  
    for (int i = 0; i < input.Length; i++)  
    {  
        if (char.IsLetter(input[i]))  
            ind++;  
    }  
    Console.WriteLine("Количество букв равно {0}", ind);  
    Console.ReadKey();  
}
```

```
Введите строку  
asd1gkjhbvdskjhnnn234  
Количество букв равно 19
```

9) подсчитывает количество цифр в строке;

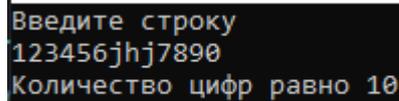
```
static void Main(string[] args)  
{  
    Console.WriteLine("Введите строку");  
    string input = Console.ReadLine();  
    int ind = 0;  
    for (int i = 0; i < input.Length; i++)  
    {
```

```

        if (char.IsNumber(input[i]))
            ind++;

    }
    Console.WriteLine("Количество цифр равно {0}", ind);
    Console.ReadKey();
}

```



```

Введите строку
123456jhj7890
Количество цифр равно 10

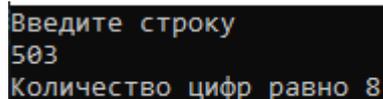
```

10) подсчитывает сумму всех содержащихся в строке цифр;
 static void Main(string[] args)

```

{
    Console.WriteLine("Введите строку");
    string input = Console.ReadLine();
    int sum = 0;
    for (int i = 0; i < input.Length; i++)
    {
        if (char.IsNumber(input[i]))
            sum += Convert.ToInt32(input[i].ToString());
    }
    Console.WriteLine("Количество цифр равно {0}", sum);
    Console.ReadKey();
}

```



```

Введите строку
503
Количество цифр равно 8

```

11) выводит на экран последовательность символов, расположенных до первого двоеточия;

```

static void Main(string[] args)
{
    Console.WriteLine("Введите строку");
    string input = Console.ReadLine();
    string[] words = input.Split(':');
    Console.WriteLine($" {words[0]}");
    Console.ReadKey();
}

```

```

Введите строку
123:sdf
123
_

```

12) выводит на экран последовательность символов, расположенных после последнего двоеточия;

```

static void Main(string[] args)
{
    Console.WriteLine("Введите строку");
    string input = Console.ReadLine();
    string[] words = input.Split(':');
    Console.WriteLine($" {words[1]}");
    Console.ReadKey();
}

```

```

Введите строку
123:asd
asd
_

```

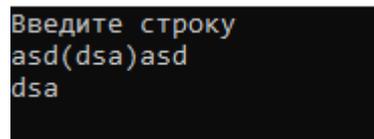
13) выводит на экран последовательность символов, расположенных между круглыми скобками (считается, что в строке ровно одна пара круглых скобок);

```

static void Main(string[] args)

```

```
{  
    Console.WriteLine("Введите строку");  
    string input = Console.ReadLine();  
    string[] words = input.Split(' ', ' ');  
    Console.WriteLine($" {words[1]}");  
    Console.ReadKey();  
}
```



```
Введите строку  
asd(dsa)asd  
dsa
```

14) находит самую длинную подстроку, состоящую только из цифр;

15) находит самую длинную подстроку, состоящую из повторяющегося символа.

Краткие теоретические сведения

Класс `StringBuilder` используется для создания и редактирования строк, обычно

строк из динамического набора данных, например из массива байтовых значений.

Создать объект класса `StringBuilder` возможно одним из следующих способов:

//создание пустой строки, размер которой по умолчанию 16 символов

```
StringBuilder a = new StringBuilder();
```

//инициализация строки и выделение памяти под 4 символа

```
StringBuilder b = new StringBuilder("abcd");
```

//создание пустой строки и выделение памяти под 100 символов

```
StringBuilder c = new StringBuilder(100);
```

//инициализация строки и выделение памяти под 100 символов

```
StringBuilder d = new StringBuilder("abcd", 100);
```

//инициализация подстрокой "bcd" и выделение памяти под 100 символов

```
StringBuilder d = new StringBuilder("abcdefg", 1, 3, 100);
```

Наиболее важными членами класса `StringBuilder` являются:

`Capacity` — определяет число символов, которые способен хранить и обрабатывать

`StringBuilder`;

`Chars` — индексатор класса;

`Length` — определяет длину объекта `StringBuilder`;

`MaxCapacity` — определяет максимальное число символов, которые способен

хранить и обрабатывать `StringBuilder`;

`Appendf()` — добавляет объект заданного типа в конец `StringBuilder`;

`AppendFormat()` — замещает или устанавливает новый формат `StringBuilder`;

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		83

EnsureCapacity() — гарантирует, что StringBuilder имеет емкость не менее

указанной в параметре;

Insert() — вставляет объект некоторого типа в указанную позицию;

Remove() — удаляет указанный символ;

Replace() — замещает все экземпляры указанных символов на новые символы.

Очень важной особенностью класса StringBuilder является то, что при изменении

значений в объекте StringBuilder происходит изменение значений в исходной строке, а не

в ее копии.

Задание 2 Разработать программу, которая позволяет ввести строку с экрана и выполняет следующие действия с введенной строкой:

- 1) вставляет в строку символ x после каждого вхождения символа y;
- 2) вставляет в строку подстроку x после каждого вхождения подстроки y;
- 3) удваивает каждое вхождение заданного символа x;
- 4) удваивает каждое вхождение заданной подстроки x;
- 5) удаляет среднюю букву, если длина строки нечетная, и две средних, если длина строки четная;
- 6) удаляет все символы x;
- 7) удаляет из строки все цифры;
- 8) удаляет все подстроки substr;
- 9) заменяет все вхождения подстроки str1 на подстроку str2 (при этом str1 может являться частью str2);
- 10) заменяет все группы стоящих рядом точек на многоточие;
- 11) меняет местами первую букву со второй, третью с четвертой и т.д.;
- 12) меняет местами первую букву с последней, вторую с

предпоследней и т.д.;

13) определяет, сколько различных символов встречается в строке;

14) удаляет из строки все подстроки, состоящие из цифр;

15) удаляет из строки самую длинную подстроку, состоящую из повторяющегося символа.

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		<i>85</i>

Контрольная работа. Строки

«удовлетворительно» – решена одна задача;

«хорошо» – решены две задачи, но допущены неточности;

«отлично» – верно решены две задачи.

Задание 1 Разработать программу, которая позволяет ввести строку с экрана и для введенной строки s:

4) удаляет все символы x;

```
static void Main(string[] args)
{
    Console.Write("Введите строку: ");
    string input = Console.ReadLine();
    string result = RemoveX(input);
    Console.WriteLine("Результат: " + result);
}

static string RemoveX(string input)
{
    string output = "";
    foreach (char c in input)
    {
        if (c != 'x')
        {
            output += c;
        }
    }
    return output;
}
```

```
Введите строку: хепон хепогаму hexagon xxxxy
Результат: епон епогаму heagon y
```

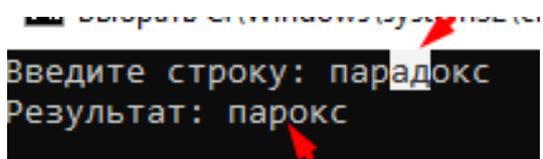
б) удаляет среднюю букву, если длина строки нечетная, и две средних, если длина строки четная;

```
static void Main(string[] args)
{
    Console.Write("Введите строку: ");
    string input = Console.ReadLine();

    StringBuilder sb = new StringBuilder(input);

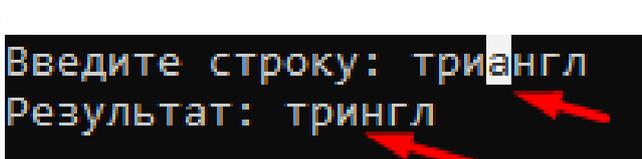
    int middleIndex = sb.Length / 2;
    if (sb.Length % 2 == 0)
    {
        sb.Remove(middleIndex - 1, 2);
    }
    else
    {
        sb.Remove(middleIndex, 1);
    }

    Console.WriteLine("Результат: " + sb.ToString());
    Console.ReadKey();
}
```



Введите строку: парадокс
Результат: парокс

This screenshot shows the program's execution with the input string "парадокс". The output is "Результат: парокс", where the two middle characters "а" and "д" have been removed. Red arrows point to the removed characters in the original image.



Введите строку: триангл
Результат: трингл

This screenshot shows the program's execution with the input string "триангл". The output is "Результат: трингл", where the middle character "а" has been removed. Red arrows point to the removed character in the original image.

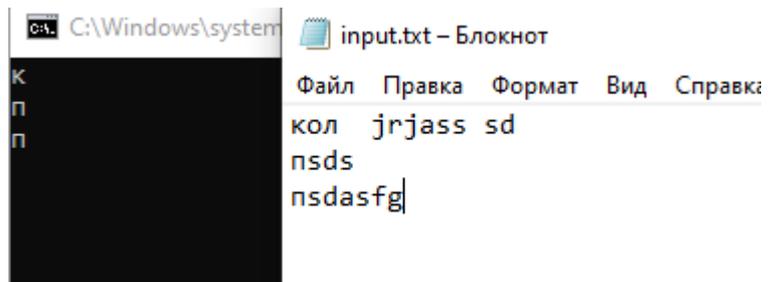
Практикум №9

Задание 1

8. Дан текстовый файл. Напечатать первый символ каждой строки.

```
static void Main(string[] args)
{
    using (StreamReader sr = new
StreamReader("C:/Users/Public/Documents/input.txt"))
    {
        while (sr.Peek() > -1)
        {
            if (sr.Peek() == ' ')
            {
                Console.WriteLine("пустота");
                break;
            }
            else
            {
                Console.WriteLine(sr.ReadLine()[0]);
            }
        }
    }
    Console.ReadKey();
}
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		88



9 Дан текстовый файл. Напечатать символы с k1 по k2 в каждой строке.

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		89

Лабораторная работа 13. Основы объектно-ориентированного программирования

```
internalclass Program
{
    staticvoid Main(string[] args)
    {
        Student std= new Student();
        std.name = "Кирилл";
        std.age = 17;
        std.Print();

        Student std2 = new Student();
        std2.name = "Максим";
        std2.age = 17;
        std2.Print();
    }
}

internalclass Student
{
    publicstring name;
    publicint age;

    publicvoid Print()
    {
        Console.WriteLine($"Моё имя{name}, возраст:{age}");
    }
}

Моё имя Кирилл, возраст:17
Моё имя Максим, возраст:17
}
```

Задание. Разработка класса

1) Создайте новый проект, выбрав: шаблон: Консольное приложение, сохраните проект в своей папке.

Напишите мини программку, которая складывает два числа введенных с клавиатуры без использования ООП:

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		90

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, c;
            string bufer;
            bufer = Console.ReadLine();
            a = Convert.ToInt16(bufer);
            bufer = Console.ReadLine();
            b = Convert.ToInt16(bufer);
            c = a + b;
            Console.WriteLine(c);
            Console.ReadLine();
        }
    }
}

```

Запустите программу на выполнение (F5).



```

static void Main(string[] args)
{
    int a, b, c;
    string bufer;
    bufer = Console.ReadLine();
    a = Convert.ToInt16(bufer);
    bufer = Console.ReadLine();
    b = Convert.ToInt16(bufer);
    c = a + b;
    Console.WriteLine(c);
    Console.ReadLine();
}
}

```

2) Создайте новый проект, в котором будет отдельный класс, а в основной программе этот класс будет использован.

Добавьте в проект новый класс (Проект – Добавить класс) и назовите этот класс Worker. В класс добавьте два общедоступных поля: имя и возраст, и одно скрытое поле: вес (обратите внимание, что пространство имен не надо переименовывать):

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Worker
    {
        public int age = 0;
        public string name;
        private float weight;
    }
}

```

Иванов, 35, 76кг

```

static void Main(string[] args)
{
    Worker wrkr = new Worker();
    wrkr.age = 35;
    wrkr.name = "Иванов";
    Console.WriteLine(wrkr.name + ", " + wrkr.age + ", ");
    Console.ReadLine();
}
}

```

3) В теле функции Main создайте объект класса Worker:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Worker wrk1 = new Worker();
            wrk1.age = 34;
            wrk1.name = "Иванов";
            Console.WriteLine(wrk1.name + ", " + wrk1.age);
            Console.ReadLine();
        }
    }
}

```

```

internal class Worker
{
    public int age = 0;
    public string name;
    public float weight;
}

```

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		92

```
}  
}
```

Запустите программу на выполнение.

Попробуйте записать значение в поле `weight`. Почему данные не записались?

4) Для записи и чтения данных из скрытых полей используют методы. Добавим вовнутрь класса `Worker` новый метод (действие) который будет отвечать за еду, если человек чего-то там съест, купитъего вес должен будет увеличиться на количество съеденного.

```
publicvoid eat (float how_much)  
{  
    weight= weight + how_much;  
}
```

Если поле вес скрытое, то мы в него не только писать не можем, но и читать тоже не можем. Для чтения данных из скрытого поля необходимо использовать еще один метод:

```
publicfloat get_weight()  
{  
return weight;  
}
```

Почему в последних двух функциях после слова `public` идут различные слова? Что они обозначают и на что влияют?

```
internalclassWorker  
{  
publicint age = 0;  
publicstring name;  
privatefloat weight;  
publicvoid eat (float how_much)  
{  
    weight= weight + how_much;  
}  
publicfloat get_weight()  
{  
return weight;  
}  
}
```

5) Теперь эти два метода надо использовать в нашей программе. Заставьте рабочего съесть 2, а затем 3 кг пищи, а потом проверьте его вес.

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		93

```

wrk1.eat(2);
wrk1.eat(3);
float ves;
ves=wrk1.get_weight();
Console.WriteLine(wrk1.name + ", " + wrk1.age + ", вес: " + ves);

```

Запустите программу на выполнение. Проверьте работоспособность.

```
17, Кирилл, вес: 65
```

```

internalclassProgram
{
staticvoid Main(string[] args)
{
Worker sasa = new Worker();
sasa.name = "Кирилл";
sasa.age =17;
sasa.eat(2);
sasa.eat(3);
float ves;
ves = sasa.get_weight();
Console.WriteLine(sasa.age + ", " + sasa.name + ", вес: " + ves);
Console.ReadKey();
}
}

```

```

internalclassWorker
{
publicint age = 0;
publicstring name;
privatefloat weight = 60;
publicvoid eat(float how_much)
{
weight = weight + how_much;
}
publicfloat get_weight()
{
return weight;
}
}

```

6) Усовершенствуйте метод eat таким образом, что если рабочий за раз съедает более чем 10 кг, то его возраст увеличивается на год, а вес увеличивается только на половину съеденного.

```
Принесли: 6
Кирилл съел 6 кг
5, Кирилл, вес: 66
```

```

internalclassProgram
{
staticvoid Main(string[] args)

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		94

```

    {
        Worker sasa = new Worker();
        sasa.name = "Кирилл";
        sasa.age = 5;
        Console.WriteLine("Принесли: ");
        int kg = int.Parse(Console.ReadLine());
        Console.WriteLine($"{sasa.name} съел{kg}кг ");
        sasa.eat(kg);
        float ves;
        ves = sasa.get_weight();
        Console.WriteLine(sasa.age + ", " + sasa.name + ", вес: " + ves);
        Console.ReadKey();
    }
}

```

```

internalclassWorker
{
    publicint age = 0;
    publicstring name;
    privatefloat weight = 60;
    publicvoid eat(float how_much)
    {
        if (how_much > 10)
        {
            weight = weight + (how_much / 2);
            age++;
        }
        else
        {
            weight = weight + how_much;
        }
    }
    publicfloat get_weight()
    {
        return weight;
    }
}
}

```

7) Измените программу так, чтобы имя рабочего и его первоначальный возраст вводились с клавиатуры и вносились в соответствующие переменные.

```

Вовут: Кирилл
Ему лет: 17
Принесли: 6
Кирилл съел 6 кг
17, Кирилл, вес: 56

```

```

internalclassProgram
{

```

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		95

```

staticvoid Main(string[] args)
{
    Worker sasa = new Worker();
    Console.Write("Зовут: ");
string name = Console.ReadLine();
    sasa.name = name;
    Console.Write("Емулет: ");
int age = Convert.ToInt32(Console.ReadLine());
    sasa.age = age;
    Console.Write("Принесли: ");
int kg = int.Parse(Console.ReadLine());
    Console.WriteLine($"{sasa.name} съел{kg}кг ");
    sasa.eat(kg);
float ves;
    ves = sasa.get_weight();
    Console.WriteLine(sasa.age + ", " + sasa.name + ", вес: " + ves);
    Console.ReadKey();
}
}

```

```

internalclassWorker
{
publicint age;
publicstring name;
privatefloat weight = 50;
publicvoid eat(float how_much)
{
if (how_much > 10)
{
    weight = weight + (how_much / 2);
    age++;
}
else
{
    weight = weight + how_much;
}
}

publicfloat get_weight()
{
return weight;
}
}

```

Запустите программу. Проверьте ее работоспособность.

8) Добавьте рабочему еще одно скрытое поле, которое будет отвечать за настройку и будет иметь первоначальное значение равное 10.

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		96

Добавьте три метода: гулять (метод должен увеличивать настроение на 1), танцевать (метод должен увеличивать настроение на 2) и работать (метод должен уменьшать настроение на 2).

Дополните основную программу так, чтобы рабочий после еды два раза погулял и три раза потанцевал.

```
internalclass Program
{
    staticvoid Main(string[] args)
    {
        Worker sasa = new Worker();
        Console.WriteLine("Зовут: ");
        string name = Console.ReadLine();
        sasa.name = name;
        Console.WriteLine("Емулет: ");
        int age = Convert.ToInt32(Console.ReadLine());
        sasa.age = age;
        Console.WriteLine("Принесли: ");
        int kg = int.Parse(Console.ReadLine());
        Console.WriteLine($"{sasa.name} съел{kg}кг ");
        sasa.eat(kg);
        float ves;
        ves = sasa.get_weight();
        int dance = int.Parse(Console.ReadLine());
        Console.WriteLine($"Онтанцевал{dance}раз(а)");
        sasa.dance(dance);
        int walk = int.Parse(Console.ReadLine());
        Console.WriteLine($"Онгулял{walk}раз(а)");
        sasa.walk(walk);
        int work = int.Parse(Console.ReadLine());
        Console.WriteLine($"Он работал {work} раз(а)");
        sasa.work(work);
        float mood;
        mood = sasa.get_mood();
        Console.WriteLine(sasa.age + " лет, " + sasa.name + ", вес: " + ves + ", настроение: " + mood);
        Console.ReadKey();
    }
}
```

```
internalclass Worker
{
    publicint age;
    publicstring name;
    privatefloat weight = 50;
    privatefloat mood = 10;
    publicvoid eat(float how_much)
    {
        if (how_much > 10)
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		97

```

        {
            weight = weight + (how_much / 2);
        }
        age++;
    }
    else
    {
        weight = weight + how_much;
    }
}

public float get_weight()
{
    return weight;
}

public void dance(float how_much)
{
    mood = mood + how_much * 2;
    Console.WriteLine("Он устал");
}

public void walk(float how_much)
{
    mood = mood + how_much;
    Console.WriteLine("Он устал");
}

public void work(float how_much)
{
    mood = mood - how_much * 2;
    Console.WriteLine("Он очень утомился");
}

public float get_mood()
{
    return mood;
}
}
}

```

```

Зовут: Кирилл
Ему лет: 17
Принесли: 20
Кирилл съел 20 кг
20
Он танцевал 20 раз(а)
Он устал
14
Он гулял 14 раз(а)
Он устал
20
Он работал 20 раз(а)
Он очень утомился
18 лет, Кирилл, вес: 60, настроение: 24

```

1) Добавьте в класс функцию, которая будет возвращать текущее настроение пользователя.

					КОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		98

```

internal class Worker
{
    public int age;
    public string name;
    private float weight = 50;
    private float mood = 10;
    public void eat(float how_much)
    {
        if (how_much > 10)
        {
            weight = weight + (how_much / 2);
            age++;
        }
        else
        {
            weight = weight + how_much;
        }
    }

    public float get_weight()
    {
        return weight;
    }
    public void dance(float how_much)
    {
        mood = mood + how_much * 2;
        Console.WriteLine("Он устал");
    }
    public void walk(float how_much)
    {
        if (how_much > 0)
        {
            mood = mood + how_much;
        }
        Console.WriteLine("Он устал");
    }
    public void work(float how_much)
    {
        mood = mood - how_much * 2;
        Console.WriteLine("Он очень утомился");
    }
    public float get_mood()
    {
        return mood;
    }
}

internal class Program
{
    static void Main(string[] args)
    {
        Worker sasa = new Worker();
    }
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		99

```

Console.Write("Зовут: ");
string name = Console.ReadLine();
sasa.name = name;
Console.Write("Ему лет: ");
int age = Convert.ToInt32(Console.ReadLine());
sasa.age = age;
Console.Write("Принесли: ");
int kg = int.Parse(Console.ReadLine());
Console.WriteLine($"{sasa.name} съел {kg} кг ");
sasa.eat(kg);
float ves;
ves = sasa.get_weight();
int dance = int.Parse(Console.ReadLine());
Console.WriteLine($"Он танцевал {dance} раз(а)");
sasa.dance(dance);
int walk = int.Parse(Console.ReadLine());
Console.WriteLine($"Он гулял {walk} раз(а)");
sasa.walk(walk);
int work = int.Parse(Console.ReadLine());
Console.WriteLine($"Он работал {work} раз(а)");
sasa.work(work);
float mood;
mood = sasa.get_mood();
Console.WriteLine(sasa.age + " лет, " + sasa.name + ", вес: " + ves + ", настроение: " + mood);

Console.WriteLine("Он отдохнет?");
Console.WriteLine("Да - 1; Нет - 0");
int sleep = int.Parse(Console.ReadLine());
if (sleep > 0)
{
    mood = 10;
    Console.WriteLine("Он отдохнул");
}
else
{
    Console.WriteLine("Данное настроение: " + mood);
}
Console.WriteLine("Параметры изменились: " + sasa.age + ", " + sasa.name + ", вес: " + ves + ",
настроение: " + mood);

    Console.ReadKey();
}
}
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		100

```

Зовут: Кирилл
Ему лет: 17
Принесли: 20
Кирилл съел 20 кг
12
Он танцевал 12 раз(а)
Он устал
23
Он гулял 23 раз(а)
Он устал
21
Он работал 21 раз(а)
Он очень утомился
18 лет, Кирилл, вес: 60, настроение: 15
Он отдохнет?
Да - 1; Нет - 0
1
Он отдохнул
Параметры изменились: 18, Кирилл, вес: 60, настроение: 10

```

10) Добавьте в основную программу метод Работать 9 раз (можно в цикле) и выведите настроение пользователя на экран.

Настроение получилось отрицательным? – ужасно. Измените, метод Работать таким образом, что бы настроение никогда не было меньше нуля (т.е. если настроение было 1 и человек поработал, то оно должно стать не меньше 0).

Проверьте заново работоспособность программы.

```

Зовут: Кирилл
Ему лет: 17
Принесли: 20
Кирилл съел 20 кг
21
Он танцевал 21 раз(а)
20
Он гулял 20 раз(а)
11
Он работал 11 раз(а)
18, Кирилл, вес: 60, настроение: 50
Он отдохнет?
Да - 1; Нет - 0
0
Данное настроение: 50
Параметры изменились: 18, Кирилл, вес: 60, настроение: 50

```

```

static void Main(string[] args)
{
    Worker sasa = new Worker();
    Console.Write("Зовут: ");
    string name = Console.ReadLine();
    sasa.name = name;
    Console.Write("Ему лет: ");
    int age = Convert.ToInt32(Console.ReadLine());
    sasa.age = age;
    Console.Write("Принесли: ");
    int kg = int.Parse(Console.ReadLine());
    Console.WriteLine($"{sasa.name} съел {kg} кг ");
    sasa.eat(kg);
    float ves;
    ves = sasa.get_weight();
    int dance = int.Parse(Console.ReadLine());
    Console.WriteLine($"Он танцевал {dance} раз(а)");
    sasa.dance(dance);
    int walk = int.Parse(Console.ReadLine());
    Console.WriteLine($"Он гулял {walk} раз(а)");
    sasa.walk(walk);
    int work = int.Parse(Console.ReadLine());
    Console.WriteLine($"Он работал {work} раз(а)");
    sasa.work(work);
    float mood;

    mood = sasa.get_mood();

    Console.WriteLine(sasa.age + ", " + sasa.name + ", вес: " + ves + ", настроение: " + mood);

    Console.WriteLine("Он отдохнет?");
    Console.WriteLine("Да - 1; Нет - 0");
    int sleep = int.Parse(Console.ReadLine());
    if (sleep > 0)
    {
        mood = 10;
        Console.WriteLine("Он отдохнул");
    }
    else
    {
        Console.WriteLine("Данное настроение: " + mood);
    }
    Console.WriteLine("Параметры изменились: " + sasa.age + ", " + sasa.name + ", вес: " + ves + ",
настроение: " + mood);
    Console.ReadKey();
}
}

internal class Worker
{
    public int age;
    public string name;
}

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		102

```

private float weight = 50;
private float mood = 10;
public void eat (float how_much)
{
    if (how_much > 10)
    {
        weight = weight + (how_much / 2);
        age++;
    }
    else
    {
        weight = weight + how_much;
    }
}

public float get_weight()
{
    return weight;
}
public void dance(float how_much)
{
    if (how_much > 0)
    {
        mood = mood + how_much * 2;
    }
}
public void walk(float how_much)
{
    if (how_much > 0)
    {
        mood = mood + how_much;
    }
}
public void work(float how_much)
{
    if (how_much > 0)
    {
        mood = mood - how_much * 2;
        if (mood < 0)
        {
            mood = 1;
        }
    }
}
public float get_mood()
{
    return mood;
}
}

```

Практическая работа № (14)

Задание 1.1

```
using System;
using System.Text;
using System.IO;
namespace MyProgram
{
    class Program
    {
        static void Main()
        {
            using (StreamReader fileIn =
                new StreamReader("C:/fo/input.txt"))
            {
                using (StreamWriter fileOut =
                    new StreamWriter("C:/fo/ouput.txt", false))
                {
                    string line = fileIn.ReadToEnd();
                    StringBuilder a = new StringBuilder(line);

                    for (int i = 0; i < a.Length;)
                    {
                        if (char.IsPunctuation(a[i]))
                        {
                            a.Remove(i, 1);
                        }
                        else
```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		104


```

static void Main(string[] args)
{
    using (System.IO.StreamReader sr = new
System.IO.StreamReader("C:/fo/input.txt"))
    {
        while (sr.Peek() > -1)
        {
            try
            {
                Console.WriteLine(sr.ReadLine()[0]);
            }
            catch { Console.WriteLine(" "); }
        }
    }
    Console.ReadKey();
}
}
}

```

Задание 1.2

```

static void Main()
{
    using (StreamReader fileIn =
new StreamReader("D:/Example/input.txt"))
    {
        using (StreamWriter fileOut =

```

					<i>KKOO.OAИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		106

```

new StreamWriter("D:/Example/output.txt",
false))
{
    string line = fileIn.ReadToEnd();
    StringBuilder a = new StringBuilder(line);
// удаляем из строк все знаки пунктуации,
    // а также заменяем в ней все знаки
    // табуляции и перевод строки на пробел
for (int i = 0; i < a.Length;)
    {
        if (char.IsPunctuation(a[i]))
        {
            a.Remove(i, 1);
        }
        else
        {
            if (char.IsWhiteSpace(a[i]))
            {
                a[i] = ' ';
            }
            ++i;
        }
    }
    string[] s = a.ToString().Trim().Split(' ',
(char)StringSplitOptions.RemoveEmptyEntries);
    Console.WriteLine("Искомые слова: ");
    int z = 0;
    foreach (string str in s)

```

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		107


```

Console.WriteLine("Оригинальная строка: {0}; изменённая строка: {1}.",
s, s.Substring(start, end - start));
    }
}
static void Main(string[] args)
{
    MakeTextFile();
    PrintPartOfStrings(4, 11);
    Console.ReadLine();
}

```

Задание2.1

```

static void Main(string[] args)
{
    string data;

    using (StreamReader sr = new StreamReader("C:/fos/f.txt"))
    {
        data = sr.ReadLine();
    }

    string[] arrayString = Regex.Split(data, "[ ,:;]");

    using (StreamWriter sr1 = new StreamWriter(new
FileStream("C:/fos/g.txt", FileMode.OpenOrCreate)),
        sr2 = new StreamWriter(new FileStream("C:/fos/h.txt",
FileMode.OpenOrCreate)))
    {

```

					<i>ККОО.ОАИП4211121.000</i>	<i>Лис</i>
<i>Изм.</i>	<i>Лис</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>		109

					ККОО.ОАИП4211121.000	Лис
Изм.	Лис	№ докум.	Подпись	Дат		112