



ОТЧЁТ ПО ЛАБРАТОРНОЙ РАБОТЕ №5

Объектно-ориентированное программирование

Подготовил Шкода Глеб Ярославович
Студент 2 курса факультета ИКТ Университета ИТМО
Группа K32211
Преподаватель Иванов Сергей Евгеньевич

Упражнение 1

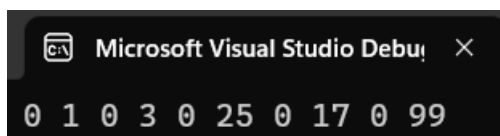
Вернёмся к коду написанному в рамках упражнения 2 из работы 3.

```
1 using System;
  0 references
2 class Program {
  0 references
3     static void Main() {
4
5         var input = Console.ReadLine().Split();
6         int k = int.Parse(input[0]), m = int.Parse(input[1]);
7         int ans = 0;
8         for (int i = 1; i <= 100; i++) {
9             if (i > k && i < m)
10                 continue;
11             ans += i;
12         }
13         Console.WriteLine(ans);
14     }
15 }
```

Теперь создадим новый массив, в котором будут храниться целые числа. После чего выведем его, подставляя на место чётных элементов нули.

```
1 using System;
  0 references
2 class Program {
  0 references
3     static void Main() {
4
5         int[] myArray = { 100, 1, 32, 3, 14, 25, 6, 17, 8, 99 };
6         int ans = 0;
7         for (int i = 0; i < myArray.Length; i++) {
8             if (myArray[i] % 2 == 0)
9                 myArray[i] = 0;
10             Console.Write("{0} ", myArray[i]);
11         }
12     }
13 }
```

Протестируем написанный код:



```
Microsoft Visual Studio Debug Console
0 1 0 3 0 25 0 17 0 99
```

Теперь пусть длину массива будет задавать пользователь. Тогда на вход подаётся число n и n элементов массива. После чего программа выведет их на экран в строчку.

```

1  using System;
   0 references
2  class Program {
   0 references
3      static void Main() {
4
5          int n = int.Parse(Console.ReadLine());
6          int[] myArray = new int[n];
7          int ans = 0;
8          for (int i = 0; i < n; i++) {
9              Console.Write("a[{0}] = ", i);
10             myArray[i] = int.Parse(Console.ReadLine());
11         }
12
13         foreach (int x in myArray)
14             Console.Write("{0} ", x);
15     }
16 }

```

Результат тестирования:

```

Microsoft Visual Studio
5
a[0] = 2
a[1] = 5
a[2] = 8
a[3] = -2
a[4] = 0
2 5 8 -2 0

```

Упражнение 2

Напишем программу для перемножения заранее заданных матриц 2x2, которая будет выводить результат перемножения на экран.

```

1  using System;
   0 references
2  class Program {
   0 references
3      static void Main() {
4
5          int[,] a = new int[2, 2], b = new int[2, 2], result = new int[2, 2];
6          int k = 1;
7          for (int i = 0; i < 2; i++) {
8              for (int j = 0; j < 2; j++) {
9                  a[i, j] = k;
10                 b[i, j] = 4 + k;
11                 k++;
12             }
13         }
14
15         result[0, 0] = a[0, 0] * b[0, 0] + a[0, 1] * b[1, 0];
16         result[0, 1] = a[0, 0] * b[0, 1] + a[0, 1] * b[1, 1];
17         result[1, 0] = a[1, 0] * b[0, 0] + a[1, 1] * b[1, 0];
18         result[1, 1] = a[1, 0] * b[0, 1] + a[1, 1] * b[1, 1];
19         Console.WriteLine("{0} {1}", result[0, 0], result[0, 1]);
20         Console.WriteLine("{0} {1}", result[1, 0], result[1, 1]);
21     }
22 }
23

```

Протестируем программу и убедимся, что результат совпадает с тем, который приводят в условии:

```
Microsoft Visual S
19 22
43 50
```

Теперь вынесим метод вывод итоговой матрицы в отдельный метод Output, при чём на этом раз вывод будет организован с помощью вложенного цикла.

```
private static void Output(int[,] result) {
    for (int r = 0; r < result.GetLength(0); r++) {
        for (int c = 0; c < result.GetLength(1); c++) {
            Console.Write("{0} ", result[r, c]);
        }
        Console.WriteLine();
    }
}
```

После изменений в коде, вывод программы не изменился.

Аналогичным образом вынесим и метод перемножения матриц.

```
private static int[,] Multiply(int[,] a, int[,] b) {
    int[,] result = new int[2, 2];
    result[0, 0] = a[0, 0] * b[0, 0] + a[0, 1] * b[1, 0];
    result[0, 1] = a[0, 0] * b[0, 1] + a[0, 1] * b[1, 1];
    result[1, 0] = a[1, 0] * b[0, 0] + a[1, 1] * b[1, 0];
    result[1, 1] = a[1, 0] * b[0, 1] + a[1, 1] * b[1, 1];
    return result;
}
```

В этом случае вызов функции перемножения и вывод ответа из метода main примет вид:

```
Output(Multiply(a, b));
```

После изменений в коде, вывод программы не изменился.

Обновим метод Multiply. Теперь расчёт матрицы result будет проходить с помощью вложенного цикла.

```
private static int[,] Multiply(int[,] a, int[,] b) {
    int[,] result = new int[2, 2];
    for (int r = 0; r < result.GetLength(0); r++) {
        for (int c = 0; c < result.GetLength(1); c++) {
            result[r, c] += a[r, 0] * b[0, c] + a[r, 1] * b[1, c];
        }
    }
    return result;
}
```

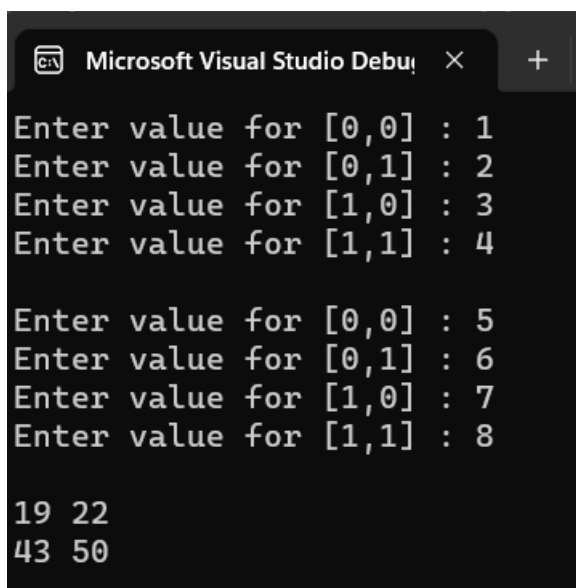
Создадим метод Input, с помощью которого будем вводить исходные матрицы.

```
2 references
private static void Input(int[,] a) {
    for (int r = 0; r < a.GetLength(0); r++) {
        for (int c = 0; c < a.GetLength(1); c++) {
            Console.WriteLine("Enter value for [{0},{1}] : ", r, c);
            string s = System.Console.ReadLine();
            a[r, c] = int.Parse(s);
        }
    }
    Console.WriteLine();
}
```

Теперь достаточно вызвать метод Input из main для a и b, чтобы считать их с консоли.

```
int[,] a = new int[2, 2], b = new int[2, 2];
Input(a);
Input(b);
```

Протестируем, получившуюся в итоге программу:



```
Microsoft Visual Studio Debug Console
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
```

Как видно, из результатов тестирования, после всех изменений программа по прежнему выводит верный ответ.

Итоговый вид программы после всех изменений:

```

1  using System;
   0 references
2  class Program {
   1 reference
3      private static void Output(int[,] result) {
4          for (int r = 0; r < result.GetLength(0); r++) {
5              for (int c = 0; c < result.GetLength(1); c++) {
6                  Console.Write("{0} ", result[r, c]);
7              }
8              Console.WriteLine();
9          }
10     }
   1 reference
11     private static int[,] Multiply(int[,] a, int[,] b) {
12         int[,] result = new int[2, 2];
13         for (int r = 0; r < result.GetLength(0); r++) {
14             for (int c = 0; c < result.GetLength(1); c++) {
15                 result[r, c] += a[r, 0] * b[0, c] + a[r, 1] * b[1, c];
16             }
17         }
18         return result;
19     }
20
   2 references
21     private static void Input(int[,] a) {
22         for (int r = 0; r < a.GetLength(0); r++) {
23             for (int c = 0; c < a.GetLength(1); c++) {
24                 Console.Write("Enter value for [{0},{1}] : ", r, c);
25                 string s = System.Console.ReadLine();
26                 a[r, c] = int.Parse(s);
27             }
28         }
29         Console.WriteLine();
30     }
   0 references
31     static void Main() {
32
33         int[,] a = new int[2, 2], b = new int[2, 2];
34         Input(a);
35         Input(b);
36         Output(Multiply(a, b));
37     }
38 }

```

Упражнение 3

Сначала реализуем методы из основной части упражнения:

```

4 references
private static int Sum(int[] a) {
    int ans = 0;
    foreach (int x in a)
        ans += x;
    return ans;
}

0 references
private static int Average(int[] a) {
    return (Sum(a) / a.Length);
}

2 references
private static int SumOfEven(int[] a) {
    int ans = 0;
    foreach (int x in a) {
        if (x % 2 == 0)
            ans += x;
    }
    return ans;
}

1 reference
private static int SumOfOdd(int[] a) {
    return (Sum(a) - SumOfEven(a));
}

2 references
private static int SumOfPositive(int[] a) {
    int ans = 0;
    foreach (int x in a) {
        if (x > 0)
            ans += x;
    }
    return ans;
}

1 reference
private static int SumOfNegative(int[] a) {
    return (Sum(a) - SumOfPositive(a));
}

```

Обратим внимание, что некоторые методы реализованы на основе других, чтобы избежать написание большого объёма похожего кода.

Теперь реализуем методы из дополнительных заданий:

```

private static int iMin(int[] a) {
    int mi = int.MaxValue, ans = -1;
    for (int i = 0; i < a.Length; i++) {
        if (a[i] < mi) {
            mi = a[i];
            ans = i;
        }
    }
    return ans;
}

3 references
private static int iMax(int[] a) {
    int ma = int.MinValue, ans = -1;
    for (int i = 0; i < a.Length; i++) {
        if (a[i] > ma) {
            ma = a[i];
            ans = i;
        }
    }
    return ans;
}

```

```
private static int SumBetween(int[] a) {
    int ans = 0;
    for (int i = Math.Min(iMax(a), iMin(a)) + 1; i < Math.Max(iMax(a), iMin(a)); i++)
        ans += a[i];
    return ans;
}
```

После этого в методе main реализуем ввод массива с клавиатуры и вызов всех, ранее созданных методов, с целью тестирования.

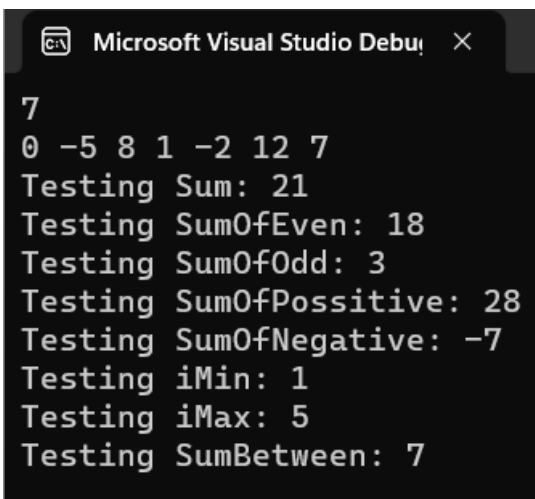
```
static void Main() {

    int n = int.Parse(Console.ReadLine());
    int[] a = new int[n];
    var input = Console.ReadLine().Split();
    for (int i = 0; i < n; i++)
        a[i] = int.Parse(input[i]);

    Console.WriteLine("Testing Sum: {0}", Sum(a));
    Console.WriteLine("Testing SumOfEven: {0}", SumOfEven(a));
    Console.WriteLine("Testing SumOfOdd: {0}", SumOfOdd(a));
    Console.WriteLine("Testing SumOfPositive: {0}", SumOfPositive(a));
    Console.WriteLine("Testing SumOfNegative: {0}", SumOfNegative(a));
    Console.WriteLine("Testing iMin: {0}", iMin(a));
    Console.WriteLine("Testing iMax: {0}", iMax(a));
    Console.WriteLine("Testing SumBetween: {0}", SumBetween(a));

}
```

Проверим работу программы на произвольном массиве:



```
Microsoft Visual Studio Debug Console
7
0 -5 8 1 -2 12 7
Testing Sum: 21
Testing SumOfEven: 18
Testing SumOfOdd: 3
Testing SumOfPositive: 28
Testing SumOfNegative: -7
Testing iMin: 1
Testing iMax: 5
Testing SumBetween: 7
```

Несложно убедиться, проверив работу методов вручную, что все 8 ответов являются правильными.