Отчет по практической работе №14

Наименование работы: события.

Индивидуальные задания (вариант №9):

1. Для каждого варианта создать событие, в случае необходимости создать несколько событий. ВСЕ методы должны запускаться после выполнения события. Событие должно запускаться по какому-либо условию.

Создать приложение выполняющее сортировку массива данных и визуального отображения процесса сортировки на экране. Первый метод производит сортировку по возрастанию, второй по убыванию. После каждого перемещения элемента результирующий производится графический вывод текущего состояния сортировки. Каждый метод работает с отдельным экземпляром массива данных. Состояние сортировки выводится в двух элементах.

Ход работы:

1. Разработал программу в соответствии с заданием №1. Были реализованы методы, делегаты, события.

Листинг программы:

using System;

namespace Lab14

{

public delegate void AccountHandler(string message);

class Program

{

public static event AccountHandler Notify;

static int[] arr;

static int n;

static void Main(string[] args)

{

Console.Write("Введите длину массива: ");

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

arr = new int[n];

Random rnd = new Random();

Console.Write("Исходный массив: ");

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

{

arr[i] = rnd.Next(-100, 100);

Console.Write(arr[i] + " ");

}

Notify += Message;

Up();

Notify -= Message;

Notify += Msg;

Down();

}

public static void Message(string message)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine(message);

Console.ResetColor();

}

public static void Msg(string message)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(message);

Console.ResetColor();

}

public static void Up()

{

int temp;

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

{

for(int j = i + 1; j < n; j++)

{

if(arr[i] > arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

Console.WriteLine();

Notify?.Invoke($"Сортируем число: {temp}");

}

}

}

Console.WriteLine();

Console.Write("Отсортированный массив: ");

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

{

Console.Write(arr[i] + " ");

}

}

public static void Down()

{

int temp;

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

{

for (int j = i + 1; j < n; j++)

{

if (arr[i] < arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

Console.WriteLine();

Notify?.Invoke($"Сортируем число: {temp}");

}

}

}

Console.WriteLine();

Console.Write("Отсортированный массив: ");

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

{

Console.Write(arr[i] + " ");

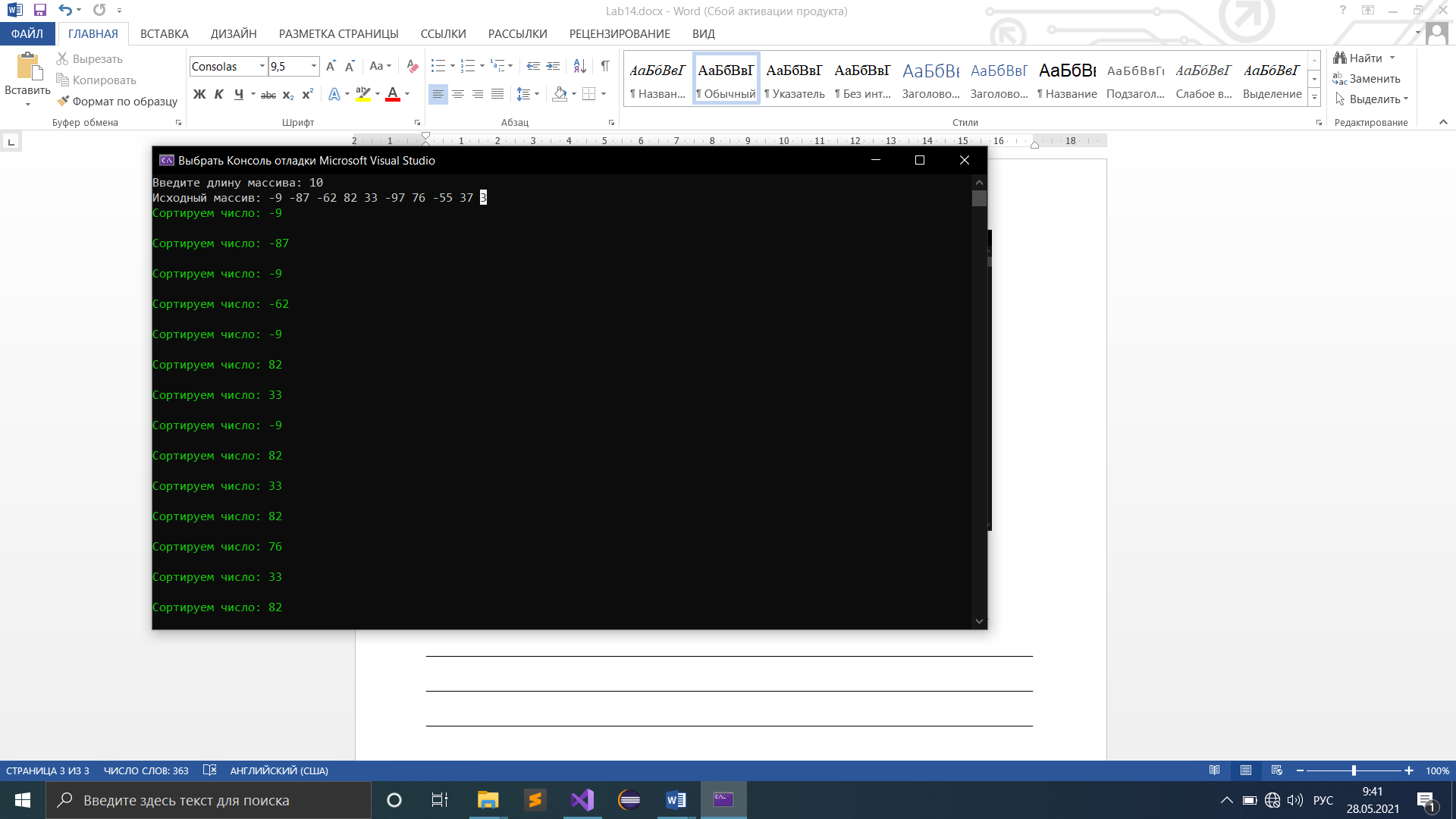
}

}

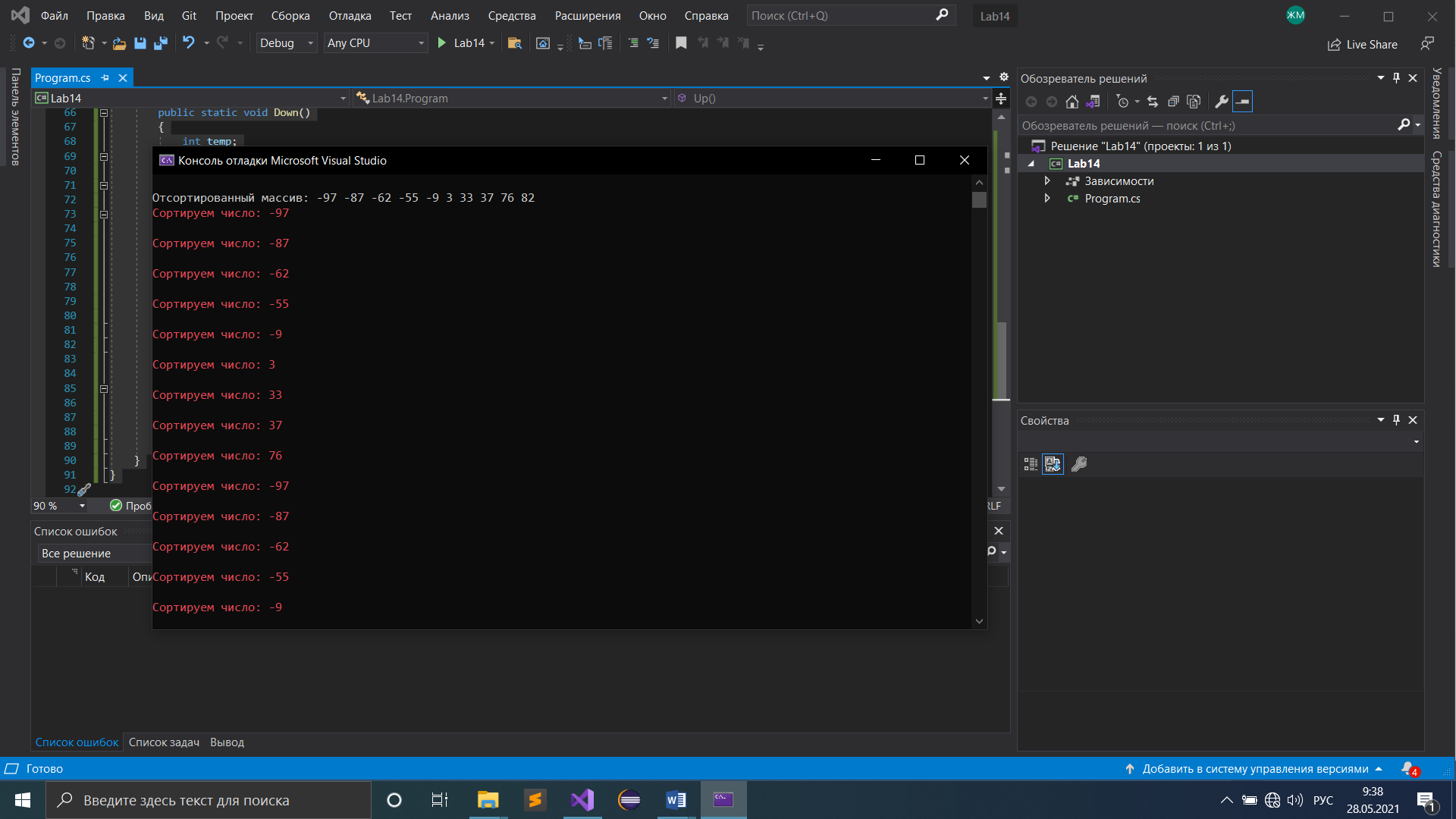
}

}

Результат выполнения программы Рис 1.1.



**Рисунок 1.1 – Пример работы программы c событиями**



**Рисунок 1.2 – Пример работы программы c событиями**

Отметка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_