

ПРАКТИЧЕСКАЯ РАБОТА 7. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ МНОГОМЕРНЫХ МАССИВОВ

Цель практической работы: изучить свойства компонента **dataGridView**. Написать программу с использованием двумерных массивов.

7.1. Двухмерные массивы

Многомерные массивы имеют более одного измерения. Чаще всего используются двумерные массивы, которые представляют собой таблицы. Каждый элемент массива имеет два индекса, первый определяет номер строки, второй - номер столбца, на пересечении которых находится элемент. Нумерация строк и столбцов начинается с нуля. Объявить двумерный массив можно одним из предложенных способов:

```
тип [,] имя__массива;  
тип [,] имя__массива = new тип [размер1, размер2];  
тип [,] имя__массива={{элементы 1-ой строки}, ... , {элементы n-ой строки}};  
тип [,] имя__массива= new тип [,]{{элементы 1-ой строки}, ... , {элементы n-ой строки}};  
строки}};
```

Пример кода, использующего многомерные массивы:

```
// объявление и инициализация двумерного массива  
int[,] array2D = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };  
// Объявление такого массива с указанием размерности (кол-во строки столбцов)  
int[,] array2Da = new int[4, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };  
// Объявление двумерного массива элементами, которого являются строки  
string[,] array2Db = new string[3, 2] { { "one", "two" }, { "three", "four" },  
                                         { "five", "six" } };  
  
// Объявление трехмерного массива  
int[, ,] array3D = new int[, ,] { { { 1, 2, 3 }, { 4, 5, 6 } },  
                                   { { 7, 8, 9 }, { 10, 11, 12 } } };  
// Объявление трехмерного массива с указанием размерности  
int[, ,] array3Da = new int[2, 2, 3] { { { 1, 2, 3 }, { 4, 5, 6 } },  
                                         { { 7, 8, 9 }, { 10, 11, 12 } } };  
  
// Доступ к элементам массива  
System.Console.WriteLine(array2D[0, 0]);  
System.Console.WriteLine(array2D[0, 1]);  
System.Console.WriteLine(array2D[1, 0]);  
System.Console.WriteLine(array2D[1, 1]);  
System.Console.WriteLine(array2D[3, 0]);  
System.Console.WriteLine(array2Db[1, 0]);  
System.Console.WriteLine(array3Da[1, 0, 1]);  
System.Console.WriteLine(array3D[1, 1, 2]);  
  
// Результаты работы программы (выводятся в консоль):  
// 1  
// 2  
// 3  
// 4  
// 7
```

```
// three
// 8
// 12
```

7.2. Элемент управления DataGridView

При работе с двумерными массивами ввод и вывод информации на экран удобно организовывать в виде таблиц. Элемент управления **DataGridView** может быть использован для отображения информации в виде двумерной таблицы. Для обращения к ячейке в этом элементе необходимо указать номер строки и номер столбца. Например: **dataGridView1.Rows[2].Cells[7].Value = "*" ;** данный код позволит записать во вторую строку в 7 ячейку знак звездочка.

7.3. Порядок выполнения задания

Задание: Создать программу для определения целочисленной матрицы 5 на 5. Разработать обработчик для поиска минимального элемента на дополнительной диагонали матрицы. Результат, после нажатия кнопки типа **Button**, вывести в **textBox**.

Окно программы приведено на рис. 7.1.

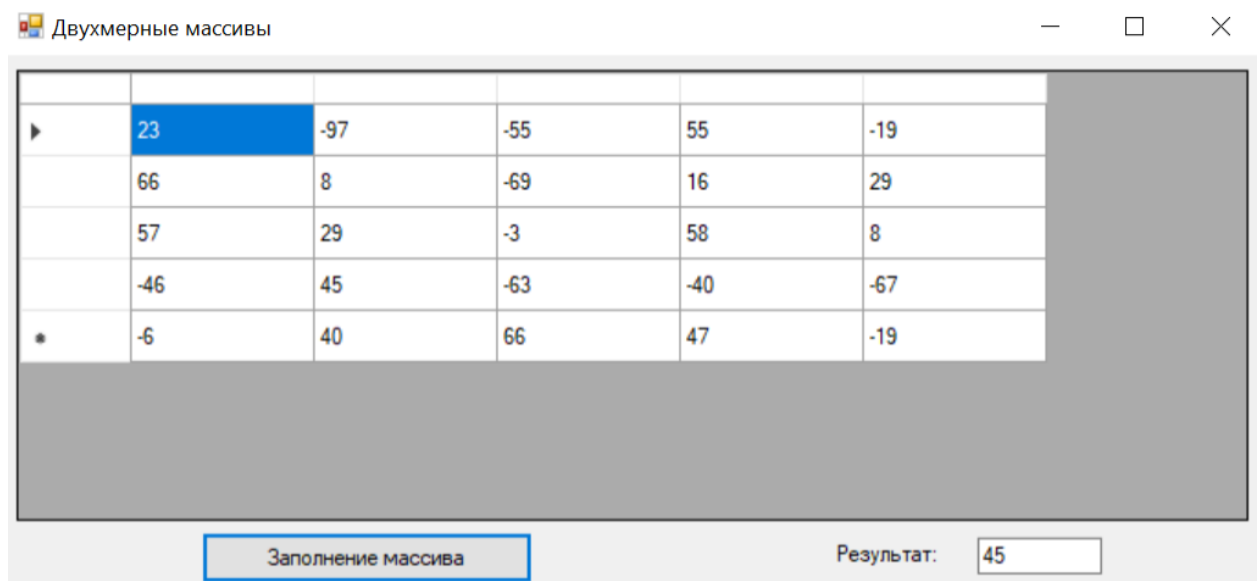


Рис. 7.1. Окно программы для работы с двухмерным массивом

Текст обработчика события нажатия на кнопку приведен ниже.

```
private void button1_Click(object sender, EventArgs e)
{
    dataGridView1.RowCount = 15; //Указываем количество строк
    dataGridView1.ColumnCount = 15; //Указываем количество столбцов
    int[,] a = new int[15,15]; //Инициализируем массив
    int i,j;
    //Заполняем матрицу случайными числами
    Random rand = new Random();
    for (i=0; i<15; i++)
```

```

        for (j=0; j<15; j++)
            a[i,j] = rand.Next(-100,100);
        //Выводим матрицу в dataGridView1
        for (i=0; i<15; i++)
            for (j=0; j<15; j++)
                dataGridView1.Rows[i].Cells[j].Value =
Convert.ToString(a[i,j]);
        //производим поиск максимального элемента на дополнительной
диагонали
        int m = int.MinValue;
        for (i = 0; i < 15; i++)
            if (a[i, 14 - i] > m) m = a[i, 14 - i];
        // выводим результат
        textBox1.Text = Convert.ToString(m);
    }

```

7.4. Индивидуальные задания

1. Найти максимальный элемент среди элементов находящихся выше главной диагонали квадратной матрицы. Указать его индексы.
2. Найти максимальный и минимальный элементы среди элементов, расположенных на побочной диагонали квадратной матрицы.
3. Поменять местами в матрице минимальный и максимальный элементы.
4. Найти в каждой строке матрицы минимальный элемент. Выделить максимальное из полученных значений.
5. Вычислить, в какой строке матрицы сумма элементов максимальная, а в какой - минимальная.
6. Найти в квадратной матрице сумму элементов, расположенных ниже побочной диагонали.
7. Найти в матрице максимальный и минимальный элементы, поменять местами строки, в которых они расположены.
8. Найти на главной диагонали квадратной матрицы максимальный и минимальный элементы. Поменять местами строки, в которых они расположены.
9. Найти для каждой строки матрицы сумму максимального и минимального элементов.
10. В строке матрицы с максимальным элементом обнулить все элементы, кроме максимального.
11. Для столбцов матрицы с четными номерами найти максимальный элемент, для столбцов с нечетными - минимальный.
12. Сформировать одномерный массив, элементы которого находятся суммированием элементов строк матрицы, в которых находятся максимальный и минимальный элементы.
13. Среди элементов квадратной матрицы, расположенных ниже побочной диагонали найти минимальный и максимальный элементы.
14. Найти в каждой строке матрицы минимальный среди положительных элементов.

15. Поменять местами в матрице максимальный и минимальный положительные элементы.
16. Обнулить столбцы матрицы, в которых находятся максимальный и минимальный элементы.