***Практическая работа № 3***

***Массивы***

***Цель лабораторной работы****:* изучить свойства компонента **dataGridView**. Написать программу с использованием одномерных и двухмерных массивов.

## Одномерные массивы

Массив - набор элементов одного и того же типа, объединенных общим именем. Массивы в С# можно использовать по аналогии с тем, как они используются в других языках программирования. Однако С#-массивы имеют существенные отличия: они относятся к ссылочным типам данных, более того - реализованы как объекты. Фактически имя массива является ссылкой на область кучи (динамической памяти), в которой последовательно размещается набор элементов определенного типа. Выделение памяти под элементы происходит на этапе инициализации массива. А за освобождением памяти следит система сборки мусора - неиспользуемые массивы автоматически утилизируются данной системой.

Рассмотрим в данной лабораторной работе одномерные массивы. *Одномерный массив* - это фиксированное количество элементов одного и того же типа, объединенных общим именем, где каждый элемент имеет свой номер. Нумерация элементов массива в С# начинается с нуля, то есть, если массив состоит из 10 элементов, то его элементы будут иметь следующие номера: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Одномерный массив в С# реализуется как объект, поэтому его создание представляет собой двухступенчатый процесс. Сначала объявляется ссылочная переменная на массив, затем выделяется память под требуемое количество элементов базового типа, и ссылочной переменной присваивается адрес нулевого элемента в массиве. Базовый тип определяет тип данных каждого элемента массива. Количество элементов, которые будут храниться в массиве, определяется размер массива.

В общем случае процесс объявления переменной типа массив, и выделение необходимого объема памяти может быть разделено. Кроме того на этапе объявления массива можно произвести его инициализацию. Поэтому для объявления одномерного массива может использоваться одна из следующих форм записи:

**базовый\_тип [] имя\_\_массива;**

Описана ссылка на одномерный массив, которая в дальнейшем может быть использована для адресации на уже существующий массив. Например:  **int [] a;**

**базовый\_тип [] имя\_\_массива = new базовый\_тип [размер];**

Объявлен одномерный массив заданного типа и выделена память под одномерный массив указанной размерности. Адрес данной области памяти записан в ссылочную переменную. Элементы массива равны нулю (В C# элементам массива присваиваются начальные значения по умолчанию в зависимости от базового типа. Для арифметических типов - нули, для ссылочных типов - null, для символов - пробел). Например: **int []a=new int [10];**

**базовый\_тип [] имя\_\_массива={список инициализации};**

Выделена память под одномерный массив, размерность которого соответствует количеству элементов в списке инициализации. Адрес этой области памяти записан в ссылочную переменную. Значение элементов массива соответствует списку инициализации. Например: **int []a={0, 1, 2, 3};**

Обращения к элементам массива происходит с помощью индекса, для этого нужно указать имя массива и в квадратных скобках его номер. Например, **a[0]**, **b[10]**, **c[i]**.

Так как массив представляет собой набор элементов, объединенных общим именем, то обработка массива обычно производится в цикле. Например:

**int[] myArray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };**

**int i;**

**for (i = 0; i < 10; ++i)**

**Console.WriteLine(myArray[i]);**

## 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 двухмерного 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 двухмерного 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 } },

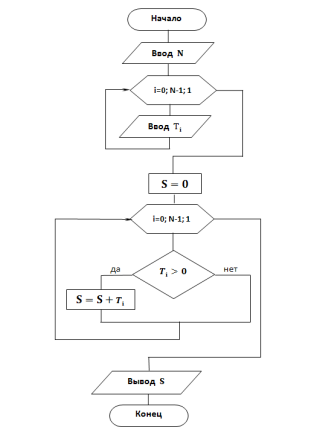
{ { 7, 8, 9 }, { 10, 11, 12 } } };

## Элемент управления DataGridView

При работе с двухмерными 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 позволяет записать во вторую строку в 7 ячейку знак звездочка.

## Порядок выполнения задания

Задание 1. Дан массив T, состоящий из n элементов. Найти сумму положительных элементов массива T(n).



Задание 2 : Создать программу для определения целочисленной матрицы 15 на 15. Разработать обработчик для поиска минимального элемента на дополнительной диаганали матрицы. Результат, после нажатия кнопки типа **Button**, вывести в **textBox**.

Окно программы приведено на рис.

Изображение выглядит как стол

Автоматически созданное описание

*Окно программы для работы с двухмерным массивом*

Текст обработчика события нажатия на кнопку приведен ниже.

Изображение выглядит как текст

Автоматически созданное описание

## Задание 3. Индивидуальные задания

1. Дана матрица A(N,M). Найти наименьший элемент в каждой строке матрицы. Вывести исходную матрицу и результаты вычислений.
2. Дана матрица A(N,M). Вычислить сумму второй строки и произведение первого столбца. Вывести исходную матрицу и результаты вычислений.
3. Дана матрица A(N,M). Найти наибольший элемент в главной диагонали. Вывести матрицу и наибольший элемент.
4. Дана матрица A(N,M). Найти сумму элементов главной диагонали и эту сумму поставить на место последнего элемента. Вывести исходную и полученную матрицу.
5. Дана матрица A(N,M). Вычислить наибольший элемент матрицы. Вывести исходную матрицу и наибольший элемент.
6. Дана матрица A(N,M).Найти количество положительных элементов.
7. Дана матрица A(N,M).Найти количество отрицательных элементов.
8. Дана матрица F(N,M). Найти наименьший элемент в каждом столбце. Вывести матрицу и найденные элементы.
9. Найти наибольший элемент главной диагонали матрицы A(N,M) и вывести всю строку, в которой он находится.
10. Найти наибольшие элементы каждой строки матрицы Z(N,M) и поместить их на главную диагональ. Вывести полученную матрицу.
11. Вычислить суммы элементов матрицы Y(N,M) по столбцам и вывести их.
12. Найти наибольший элемент матрицы A(N,M) и записать нули в ту строку и столбец, где он находится. Вывести наибольший элемент, исходную и полученную матрицу.
13. Дана матрица R(N,M). Найти наименьший элемент в каждой строке и записать его на место первого элемента строки. Вывести исходную и полученную матрицы.
14. Определить количество положительных элементов каждой строки матрицы A(N,M) и запомнить их в одномерном массиве N. Массив N вывести.
15. Вычислить и вывести сумму элементов матрицы A(N,M), расположенных над главной диагональю матрицы.

**Контрольные вопросы**

1. Какие массивы называются статическими?
2. Какие массивы называются динамическими?