**Практическая работа №3\_2**

**Двумерные массивы**

***Цель работы:*** получение практических навыков по разработке программ с использованием двумерных массивов на языке программирования C#.

***Задачи:***

1. Изучить теоретический материал

2. Написать программы для решения практических заданий

3. Ответить на контрольные вопросы.

4. Оформить отчет.

***Теоретический материал***

Массивы характеризуются таким понятием как **ранг** или количество измерений. Выше мы рассматривали массивы, которые имеют одно измерение (то есть их ранг равен 1) - такие массивы можно представлять в виде ряда (строки или столбца) элемента. Но массивы также бывают многомерными. У таких массивов количество измерений (то есть ранг) больше 1.

Массивы которые имеют два измерения (ранг равен 2) называют двухмерными. Например, создадим одномерный и двухмерный массивы, которые имеют одинаковые элементы:

|  |  |
| --- | --- |
| 1  2  3 | int[] nums1 = new int[] { 0, 1, 2, 3, 4, 5 };    int[,] nums2 = { { 0, 1, 2 }, { 3, 4, 5 } }; |

Визуально оба массива можно представить следующим образом:

**Одномерный массив nums1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |

**Двухмерный массив nums2**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| 3 | 4 | 5 |

Поскольку массив nums2 двухмерный, он представляет собой простую таблицу. Все возможные способы определения двухмерных массивов:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | int[,] nums1;  int[,] nums2 = new int[2, 3];  int[,] nums3 = new int[2, 3] { { 0, 1, 2 }, { 3, 4, 5 } };  int[,] nums4 = new int[,] { { 0, 1, 2 }, { 3, 4, 5 } };  int[,] nums5 = new [,]{ { 0, 1, 2 }, { 3, 4, 5 } };  int[,] nums6 = { { 0, 1, 2 }, { 3, 4, 5 } }; |

Массивы могут иметь и большее количество измерений. Объявление трехмерного массива могло бы выглядеть так:

1 int[,,] nums3 = new int[2, 3, 4];

Соответственно могут быть и четырехмерные массивы и массивы с большим количеством измерений. Но на практике обычно используются одномерные и двухмерные массивы.

Определенную сложность может представлять перебор многомерного массива. Прежде всего надо учитывать, что длина такого массива - это совокупное количество элементов.

Массив массивов

От многомерных массивов надо отличать массив массивов или так называемый "зубчатый массив":

int[][] nums = new int[3][];

nums[0] = new int[2] { 1, 2 }; // выделяем память для первого подмассива

nums[1] = new int[3] { 1, 2, 3 }; // выделяем память для второго подмассива

nums[2] = new int[5] { 1, 2, 3, 4, 5 }; // выделяем память для третьего подмассива

Здесь две группы квадратных скобок указывают, что это массив массивов, то есть такой массив, который в свою очередь содержит в себе другие массивы. Причем длина массива указывается только в первых квадратных скобках, все последующие квадратные скобки должны быть пусты: new int[3][]. В данном случае у нас массив nums содержит три массива. Причем размерность каждого из этих массивов может не совпадать.

Альтернативное определение массива массивов:

int[][] numbers = {

new int[] { 1, 2 },

new int[] { 1, 2, 3 },

new int[] { 1, 2, 3, 4, 5 }

};

Зубчатый массив nums

|  |  |
| --- | --- |
| 1 | 2 |
| 1 | 2 | 3 |
| 1 | 2 | 3 | 4 | 5 |

Используя вложенные циклы, можно перебирать зубчатые массивы. Например:

int[][] numbers = new int[3][];

numbers[0] = new int[] { 1, 2 };

numbers[1] = new int[] { 1, 2, 3 };

numbers[2] = new int[] { 1, 2, 3, 4, 5 };

foreach(int[] row in numbers)

{

foreach(int number in row)

{

Console.Write($"{number} \t");

}

Console.WriteLine();

}

// перебор с помощью цикла for

for (int i = 0; i<numbers.Length;i++)

{

for (int j =0; j<numbers[i].Length; j++)

{

Console.Write($"{numbers[i][j]} \t");

}

Console.WriteLine();

}

**Основные понятия массивов**

Суммируем основные понятия массивов:

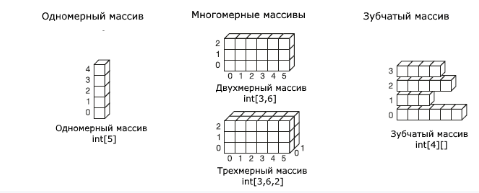
* **Ранг** (rank): количество измерений массива
* **Длина измерения** (dimension length): длина отдельного измерения массива
* **Длина массива** (array length): количество всех элементов массива

Например, возьмем массив

|  |  |
| --- | --- |
| 1 | int[,] numbers = new int[3, 4]; |

Массив numbers двухмерный, то есть он имеет два измерения, поэтому его ранг равен 2. Длина первого измерения - 3, длина второго измерения - 4. Длина массива (то есть общее количество элементов) - 12.

Примеры массивов:



Класс System.Array

Все массивы в C# построены на основе класса **Array** из пространства имен System. Этот класс определяет ряд свойств и методов, которые мы можем использовать при работе с массивами. Основные свойства и методы:

* Свойство **Length** возвращает длину массива
* Свойство **Rank** возвращает размерность массива
* **int BinarySearch (Array array, object? value)** выполняет бинарный поиск в отсортированном массиве и возвращает индекс найденного элемента
* **void Clear (Array array)** очищает массив, устанавливая для всех его элементов значение по умолчанию
* **void Copy (Array sourceArray, int sourceIndex, Array destinationArray, int destinationIndex, int length)** копирует из массива sourceArray начиная с индекс sourceIndex length элементов в массив destinationArray начиная с индекса destinationIndex
* **bool Exists<T> (T[] array, Predicate<T> match)** проверяет, содержит ли массив array элементы, которые удовлеворяют условию делегата match
* **void Fill<T> (T[] array, T value)** заполняет массив array значением value
* **T? Find<T> (T[] array, Predicate<T> match)** находит первый элемент, который удовлеворяет определенному условию из делегата match. Если элемент не найден, то возвращается null
* **T? FindLast<T> (T[] array, Predicate<T> match)** находит последний элемент, который удовлеворяет определенному условию из делегата match. Если элемент не найден, то возвращается null
* **int FindIndex<T> (T[] array, Predicate<T> match)** возвращает индекс первого вхождения элемента, который удовлеворяет определенному условию делегата match
* **int FindLastIndex<T> (T[] array, Predicate<T> match)** возвращает индекс последнего вхождения элемента, который удовлеворяет определенному условию
* **T[] FindAll<T> (T[] array, Predicate<T> match)** возвращает все элементы в виде массива, которые удовлеворяет определенному условию из делегата match
* **int IndexOf (Array array, object? value)** возвращает индекс первого вхождения элемента в массив
* **int LastIndexOf (Array array, object? value)** возвращает индекс последнего вхождения элемента в массив
* **void Resize<T> (ref T[]? array, int newSize)** изменяет размер одномерного массива
* **void Reverse (Array array)** располагает элементы массива в обратном порядке
* **void Sort (Array array)** сортирует элементы одномерного массива

***Практические задания***

В двумерном массиве, состоящем из n строк и m столбцов, вычислить:

*\*количество строк и стобцов задавать с клавиатуры*

*\*массивы заполнять случайными числами*

|  |  |
| --- | --- |
| Вариант | Задание |
| 1 | * количество строк, не содержащих ни одного нулевого элемента * максимальное из чисел, встречающихся в заданной матрице более одного раза. |
| 2 | * + произведение элементов в тех строках, которые не содержат отрицательных элементов * максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы |
| 3 | * Найти номер столбца, в котором находится самая длинная серия одинаковых элементов. * Определить сумму элементов в тех строках, которые не содержат отрицательных элементов. |
| 4 | * Определить количество столбцов, не содержащих ни одного нулевого элемента; * Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик |
| 5 | * Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. * Определить номера строк и столбцов всех седловых точек матрицы. |
| 6 | * количество нулей в каждой строке * максимальное из чисел, встречающихся в заданной матрице более одного раза |
| 7 | * Написать программу, которая проверяет, является ли введенная с клавиатуры матрица магическим квадратом. Магическим квадратом называется матрица, сумма элементов которой в каждой строке, в каждом столбце и по каждой диагонали одинакова. * Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента |
| 8 | * Найти ее строки, содержащие равное количество положительных и отрицательных элементов (нулевые элементы матрицы не учитываются). Если таких строк нет, то вывести соответствующее сообщение. * Найти количество отрицательных элементов в тех строках, которые содержат хотя бы один нулевой элемент. |
| 9 | * Упорядочить ее столбцы так, чтобы их последние элементы образовывали убывающую последовательность. * Найти минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы. |
| 10 | * + количество столбцов, содержащих хотя бы один нулевой элемент * номер строки, в которой находится самая длинная серия одинаковых элементов |
| 11 | * Определить количество и координаты особых элементов матрицы. Элемент считается особым, если он больше суммы остальных элементов своего столбца и при этом в его строке слева от него находятся элементы меньше него, а справа больше него. (Особый элемент может быть крайним в строке) * Определить количество столбцов, не содержащих ни одного нулевого элемента |
| 12 | * + сумму элементов в тех столбцах, которые не содержат отрицательных элементов * минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы |
| 13 | * Определить сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент. * Найти количество строк, среднее арифметическое элементов которых меньше заданной величины. |
| 14 | * + сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент   + номера строк и столбцов всех седловых точек матрицы * Примечание: Матрица А имеет седловую точку Аij, если Аij является минимальным элементом в I-той строке и максимальным в j-том столбце. |
| 15 | * Найти сумму элементов в тех столбцах, которые содержат хотя бы один нулевой элемент. * Определить количество столбцов, не содержащих ни одного отрицательного элемента |
| 16 | * Найти ее строки, все элементы которых различны. * Определить сумму элементов в тех строках, которые не содержат отрицательных элементов. |
| 17 | * + количество строк, содержащих хотя бы один нулевой элемент * номер столбца с максимальной суммой элементов |
| 18 | * Найти произведение элементов в тех строках, которые не содержат положительных элементов. * Найти минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы. |
| 19 | * Найти номер строки, в которой находится самая длинная серия одинаковых элементов. * Произведение элементов в тех строках, которые не содержат отрицательных элементов. |
| 20 | * Найти номер первого из ее столбцов, содержащих максимальное количество одинаковых элементов. * Определить количество строк, не содержащих ни одного нулевого элемента. |
| 21 | * Найти максимальное из чисел, встречающихся в заданной матрице более одного раза. * Найти номер первой из строк, не содержащих ни одного положительного элемента. |
| 22 | * Определить максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы. * Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент. |
| 23 | * Найти сумму модулей элементов, расположенных выше главной диагонали. * Определить минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы |
| 24 | * Найти ее строки, элементы которых упорядочены по возрастанию. * Найти сумму модулей элементов, расположенных выше главной диагонали. |
| 25 | * Определить минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы. * Упорядочить ее строки так, чтобы их минимальные элементы образовывали убывающую последовательность. |

***Контрольные вопросы***

1. Что такое ранг массива?
2. В чем отличие многомерного массива от зубчатого?
3. Какой цикл может быть использован для перебора массивов?
4. В чем отличие длины измерения от длины массива?

***Содержание отчета***

1. Титульный лист

2. Цели, задачи работы

Для каждого задания:

3. Текст задания

4. Листинг программы

5. Результаты работы

6. Ответы на контрольные вопросы