

5 МНОГОМЕРНЫЕ МАССИВЫ В ЯЗЫКЕ C#

5.1 Цель пятой темы

Изучение правил объявления и инициализации многомерных массивов и приобретение практических навыков по созданию консольных приложений с использованием многомерных массивов – матриц.

5.2 Теоретические сведения

5.2.1 Понятие многомерных массивов

Разделение массивов на одномерные массивы и многомерные носит исторический характер. Никакой принципиальной разницы между ними нет. Одномерные массивы - это частный случай многомерных массивов.

Одномерные массивы позволяют задавать такие математические структуры, как векторы, двумерные - матрицы, трехмерные - кубы данных, массивы большей размерности - многомерные кубы данных.

Размерность массива это характеристика типа. Формат записи при объявлении многомерного массива имеет следующий вид:

<тип>[, ... ,] <объявители>;

Например: int[,] matri; int[,] kubi;

Размерность массива задается запятыми, которые устанавливаются в квадратных скобках.

Число запятых, увеличенное на единицу, и задает размерность массива. Что касается объявителей, то все, что сказано для одномерных массивов, справедливо и для многомерных. Можно лишь отметить, что хотя явная инициализация с использованием многомерных константных массивов возможна, но применяется редко из-за громоздкости такой структуры. Мы будем ее применять в разделе графы при задании матрицы смежности.

Обычно инициализацию многомерных массивов выполняют программно при создании массива.

5.2.2 Массивы массивов

Еще одним видом массивов C# являются массивы массивов, называемые также «ломаными» массивами (jagged arrays).

Такой массив массивов можно рассматривать как одномерный массив, его элементы являются массивами, элементы которых, в свою очередь снова могут быть массивами, и так может продолжаться до некоторого уровня вложенности.

В каких ситуациях может возникать необходимость в таких структурах данных? Эти массивы могут применяться для представления деревьев, у которых узлы могут иметь произвольное число потомков. Таковым может быть, например, генеалогическое дерево. Вершины первого уровня - Fathers, представляющие отцов, могут задаваться одномерным

массивом, так что `Fathers[i]` - это *i*-й отец. Вершины второго уровня представляются массивом массивов - `Children`, так что `Children[i]` - это массив детей *i*-го отца, а `Children[i][j]` - это *j*-й ребенок *i*-го отца. Для представления внуков понадобится третий уровень, так что `GrandChildren [i][j][k]` будет представлять *k*-го внука *j*-го ребенка *i*-го отца.

Есть некоторые особенности в объявлении и инициализации таких массивов. Если при объявлении типа многомерных массивов для указания размерности использовались запятые, то для «ломанных» массивов применяется более ясная символика - совокупности пар квадратных скобок; например, `int[][]` задает массив, элементы которого - одномерные массивы элементов типа `int`.

Сложнее с созданием самих массивов и их инициализацией. Здесь нельзя вызвать конструктор `newint[3][5]`, поскольку он не задает «ломанный» массив. Фактически нужно вызывать конструктор для каждого массива на самом нижнем уровне. В этом и состоит сложность объявления таких массивов. Например: для абстрактного массива массивов `masmasi` целого типа выполним объявление и инициализацию следующим образом:

```
int[][]masmasi = new int[3][]
{
    newint[] {5,7,9,11},
    newint[] {2,8},
    newint[] {6,12,4}
};
```

Массив `masmasi` имеет всего два уровня. Можно считать, что у него три элемента, каждый из которых является массивом. Для каждого такого массива необходимо вызвать конструктор `new`, чтобы создать внутренний массив. В данном примере элементы внутренних массивов получают значение, будучи явно инициализированы константными массивами. Этот же массив можно объявить и инициализировать не явно следующим образом:

```
int[][]masmasi = new int[3][]
{
    newint[4],
    newint[2],
    newint[3]
};
```

В этом случае элементы массива получают при инициализации нулевые значения. Реальную инициализацию нужно будет выполнять программным путем. Стоит заметить, что в конструкторе верхнего уровня константу 3 можно опустить и писать просто `newint[][]`. Самое забавное, что вызов этого конструктора можно вообще опустить, он будет подразумеваться:

```
int[][]masmasi =
{
    newint[4],
    newint[2],
```

newint[3]

};

Но вот конструкторы нижнего уровня необходимы.

5.2.3 Алгоритмы линейной алгебры

Матрицей называется набор чисел, состоящий из m строк и n столбцов. Для программиста матрица - это двумерный массив. Матрица называется квадратной, если $m = n$, и прямоугольной - в противном случае. Числа m и n определяют размерность матрицы. Над прямоугольными матрицами определены операции транспонирования, сложения, умножения.

Пусть A - матрица размерности $m \times n$ (из m строк и n столбцов) с элементами $a_{i,j}$. Транспонированной матрицей $B = A^T$ называют матрицу размерности $n \times m$, элементы которой $b_{i,j} = a_{j,i}$. В транспонированной матрице строки исходной матрицы становятся столбцами.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix} \quad B = A^T = \begin{pmatrix} a_{1,1} & a_{2,1} & \dots & a_{m,1} \\ a_{1,2} & a_{2,2} & \dots & a_{m,2} \\ \dots & \dots & \dots & \dots \\ a_{1,n} & a_{2,n} & \dots & a_{m,n} \end{pmatrix}$$

Операция сложения определена над прямоугольными матрицами одинаковой размерности. Пусть A , B , C - прямоугольные матрицы размерности $m \times n$. Тогда сумма матриц определяется естественным образом:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix} \quad B = \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \dots & \dots & \dots & \dots \\ b_{m,1} & b_{m,2} & \dots & b_{m,n} \end{pmatrix}$$

$$C = A + B = \begin{pmatrix} a_{1,1} + b_{1,1} & a_{1,2} + b_{1,2} & \dots & a_{1,n} + b_{1,n} \\ a_{2,1} + b_{2,1} & a_{2,2} + b_{2,2} & \dots & a_{2,n} + b_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} + b_{m,1} & a_{m,2} + b_{m,2} & \dots & a_{m,n} + b_{m,n} \end{pmatrix}$$

Операция умножения определена над прямоугольными матрицами, у которых число столбцов первого сомножителя равно числу строк второго сомножителя.

Матрица произведения имеет число строк, равное числу строк первого сомножителя, и число столбцов, равное числу столбцов второго сомножителя.

Пусть A - матрица размерности $m \times p$, B - размерности $p \times n$, тогда матрица $C = A * B$ имеет размерность $m \times n$. Элементы матрицы произведения определяются как сумма попарных произведений элементов строки первого сомножителя на элементы столбца второго сомножителя.

$$A = \|a_{i,j}\| \quad i = 1, \dots, m; \quad j = 1, \dots, p; \quad B = \|b_{j,k}\| \quad j = 1, \dots, p; \quad k = 1, \dots, n$$

$$C = A * B = \|c_{i,k}\| \quad i = 1, \dots, m; \quad k = 1, \dots, n;$$

$$c_{i,k} = \sum_{j=1}^p a_{i,j} * b_{j,k};$$

Умножение всегда определено для прямой и транспонированной матрицы. Если A - прямоугольная матрица размерности $m \times n$, то всегда определена квадратная матрица B размерности $m \times m$:

$$B = A * A^T = B^T = (A * A^T)^T = (A^T)^T * A^T = A * A^T$$

Результатом такого произведения является симметричная матрица. Квадратная матрица называется симметричной, если $a_{i,j} = a_{j,i}$ для всех i и j , или, что то же, если $A = A^T$. Операции транспонирования, сложения и умножения обладают следующими свойствами:

$$(A^T)^T = A; \quad (A + B)^T = A^T + B^T; \quad (A * B)^T = B^T * A^T$$

5.2.4 Квадратные матрицы

Квадратная матрица называется диагональной, если все элементы, кроме диагональных элементов, равны нулю, то есть $a_{i,j} = 0$ при $i \neq j$.

Квадратная матрица называется единичной, если все элементы, кроме диагональных, равны нулю, а диагональные элементы равны единице. То есть $a_{i,j} = 0$ при $i \neq j$ и $a_{i,j} = 1$ при $i = j$. Единичная матрица обозначается обычно буквой E , и она играет роль единицы при умножении матриц, поскольку для любой квадратной матрицы A и единичной матрицы E той же размерности имеют место соотношения:

$$A * E = E * A = A$$

Для квадратных матриц определена функция над ее элементами, называемая определителем. Обозначается определитель обычно с помощью одинарных линий вокруг набора чисел, задающих матрицу:

$$D(A) = \begin{vmatrix} a_{1,1}, a_{1,2} \dots a_{1,n} \\ a_{2,1}, a_{2,2} \dots a_{2,n} \\ \dots \\ a_{n,1}, a_{n,2} \dots a_{n,n} \end{vmatrix}$$

Функция, задающая определитель, обладает рядом важных свойств.

Определитель диагональной матрицы равен произведению диагональных элементов. Отсюда следует, что определитель матрицы E равен 1.

Определитель матрицы не меняется при выполнении над матрицей элементарных преобразований. Под элементарной операцией (преобразованием) понимается прибавление к любой строке матрицы линейной комбинации других ее строк. В частности, если к строке матрицы с номером j прибавить строку с номером k ($k \neq j$), умноженную на некоторое число, то определитель матрицы не изменится.

Если все элементы одной строки матрицы умножить на некоторое число q , то определитель матрицы изменится в q раз (умножается на q).

Если переставить местами строки j и k , то модуль определителя не изменится, но изменится знак, если разность $|k - j|$ является нечетным числом.

Определитель произведения матриц равен произведению определителей:

$$D(A * B) = D(A) * D(B)$$

Не приводя общего формального определения, рассмотрим ниже алгоритм вычисления определителя матрицы, основанный на его свойствах.

Если определитель квадратной матрицы A не равен нулю, то существует обратная матрица, обозначаемая как A^{-1} . Прямая и обратная матрицы связаны соотношением:

$$A * A^{-1} = A^{-1} * A = E$$

Операции транспонирования, умножения и обращения матриц связаны соотношениями:

$$(A^T)^{-1} = (A^{-1})^T; \quad (A * B)^{-1} = B^{-1} * A^{-1}$$

Множество квадратных матриц одной размерности с определителем, отличным от нуля образуют группу по умножению. В группе есть единичный элемент, для каждого элемента существует обратный к нему, и произведение элементов принадлежит группе.

Определители широко используются при нахождении корней системы из n линейных уравнений с n неизвестными.

5.3 Пример выполнения задания на лабораторную работу

Рассмотрим чисто учебную задачу, позволяющую сформировать матрицы $A - 5 \times 3$ и $B - 3 \times 4$ случайных целых чисел в диапазоне от 0 до 5.

Необходимо напечатать эти матрицы, а затем найти и напечатать матрицу C , представляющую собой результат произведения матрицы $A * B$. Программа должна содержать меню.

Алгоритм произведения матриц подробно рассмотрен в теоретическом разделе 5 модуля данных методических указаний. Диапазон значений матриц A и B выбран маленьким для простоты проверки значений матрицы C .

Исходный код программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        public static void sozd(int[,], ma, int[,], mb)
        {
            Random rnd = new Random();
            Console.WriteLine("Матрицы созданы!!");
            Console.WriteLine("Матрица A 5*3:");
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    ma[i, j] = rnd.Next() % 6;
                    Console.Write(ma[i, j] + "\t");
                }
            }
        }
    }
}
```

```

    }
    Console.WriteLine();
}
Console.WriteLine();
Console.WriteLine("Матрица В 3*4:");
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 4; j++)
    {
        mb[i, j] = rnd.Next() % 6;
        Console.Write(mb[i, j] + "\t");
    }
    Console.WriteLine();
}
Console.WriteLine();

public static void umnmatr(int[, ] mc, int[, ] ma, int[, ] mb)
{
    int S;
    for (int i = 0; i < 5; i++)
    for (int j = 0; j < 4; j++)
    {
        S = 0;
        for (int k = 0; k < 3; k++)
            S = S + ma[i, k] * mb[k, j];
        mc[i, j] = S;
    }
    Console.WriteLine("Матрица С 5*4:");
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            Console.Write(mc[i, j] + "\t");
        }
        Console.WriteLine();
    }
    Console.WriteLine();
}

static void Main()
{
    int[, ] a = new int[5, 3];
    int[, ] b = new int[3, 4];
    int[, ] c = new int[5, 4];
    int k = 0;
    string buf;
    while (k < 3)
    {
        Console.WriteLine("1 - Создать и напечатать матрицы А 5*3 и В 3*4");
        Console.WriteLine("2 - Найти и напечатать матрицу С = А * В");
    }
}

```

```

Console.WriteLine("3 - Выход из программы");
Console.WriteLine("Введите пункт меню программы");
buf = Console.ReadLine();
k = Convert.ToInt32(buf);
switch (k)
{
case 1: sozd(a,b); break;
case 2: umnmatr(c,a,b); break;
default: break;
}
}
}
}

```

Работа программы:

1 - Создать и напечатать матрицы A 5*3 и B 3*4

2 - Найти и напечатать матрицу C = A * B

3 - Выход из программы

Введите пункт меню программы

1

Матрицы созданы!!

Матрица A 5*3:

2	1	3
5	5	3
2	0	2
4	4	0
4	2	3

Матрица B 3*4:

0	5	0	3
0	3	5	3
3	0	5	2

1 - Создать и напечатать матрицы A 5*3 и B 3*4

2 - Найти и напечатать матрицу C = A * B

3 - Выход из программы

Введите пункт меню программы

2

Матрица C 5*4:

9	13	20	15
9	40	40	36
6	10	10	10
0	32	20	24
9	26	25	24

1 - Создать и напечатать матрицы A 5*3 и B 3*4

2 - Найти и напечатать матрицу $C = A * B$

3 - Выход из программы

Введите пункт меню программы

5.4 Домашнее задание на лабораторную работу

Сформировать матрицу 5×5 случайных целых чисел в диапазоне от -20 до 60. Напечатать матрицу. Найти максимальный элемент и удалить строку и столбец матрицы, на пересечении которых находится максимальный элемент. Оставшиеся значения переписать в новую матрицу 4×4 . Напечатать новую матрицу. В программе использовать меню.

5.5 Индивидуальные задания для СРС

5.5.1 Индивидуальное задание 1 студента:

5.5.1.1 Сформировать матрицу $A_{5 \times 5}$ случайных целых чисел в диапазоне от минус 50 до 50. Напечатать матрицу. Найти минимальный элемент, и, если он не в первой строке, то поменять местами первую строку и строку, в которой находится минимальный элемент. Напечатать новую матрицу.

5.5.1.2 Текст некоторой строки вводится в режиме диалога и содержит простое предложение, в котором слова отделяются символом «пробел». Добавить лишние «пробелы» между словами так, чтобы длина строки стала кратна 60, а «пробелы» распределились по строке равномерно.

5.5.1.3 Сформировать матрицу $A_{7 \times 7}$ случайных чисел в диапазоне от минус 20 до 60. Напечатать ее. Вычислить и напечатать сумму значений элементов расположенных «ниже» главной и побочной диагоналей матрицы.

5.5.2 Индивидуальное задание 2 студента:

5.5.2.1 Сформировать матрицу $A_{6 \times 6}$ случайных целых чисел в диапазоне от 0 до 100. Напечатать матрицу. Все числа в каждой строке матрицы расположить в убывающем порядке. Напечатать новую матрицу.

5.5.2.2 Строка содержит простое предложение, в котором слова отделяются символом «пробел». Распечатать слова, имеющие максимальную и минимальную длину.

5.5.2.3 Сформировать матрицу $A_{10 \times 10}$ случайных целых чисел в диапазоне от 0 до 9. Напечатать матрицу. Подсчитать сколько в исходной матрице 0, 1, 2, ..., 9.

5.5.3 Индивидуальное задание 3 студента:

5.5.3.1 Сформировать матрицу $A_{5 \times 5}$ случайных целых чисел в диапазоне от минус 20 до 70. Напечатать матрицу. Найти максимальный элемент и удалить в исходной матрице столбец, в котором находится максимальный элемент. Переписать оставшиеся значения в новую матрицу $B_{5 \times 4}$. Напечатать новую матрицу.

5.5.3.2 Строка, введенное в режиме диалога, содержит условие оператора `if` языка C#, включающее не более 2 скобок. Определить, правильно ли записано введенное в режиме диалога условие.

5.5.3.3 Сформировать матрицу $A_{6 \times 6}$ случайных целых чисел в диапазоне от минус 30 до 50. Напечатать матрицу. Числа главной и побочной диагоналей матрицы, расположить в возрастающем порядке. Напечатать новую матрицу.

5.5.4 Индивидуальное задание 4 студента:

5.5.4.1 Сформировать матрицу $A_{6 \times 6}$ случайных целых чисел в диапазоне от минус 50 до 50. Напечатать матрицу. Подсчитать сумму положительных и отрицательных элементов. Если сумма отрицательных чисел по модулю больше суммы положительных, то начиная с первого отрицательного числа менять знаки чисел на плюс до тех пор, пока сумма положительных по модулю не станет больше суммы отрицательных. Напечатать новую матрицу.

5.5.4.2 Строка, введенное в режиме диалога, содержит простое предложение, в котором слова отделяются символом «пробел». Сформировать новую строку, в которой поменять местами слова исходной строки, имеющие максимальную и минимальную длину.

5.5.4.3 Сформировать матрицу $A_{5 \times 5}$ случайных чисел в диапазоне от 0 до 20. Напечатать ее. Найти и напечатать только повторяющиеся числа матрицы и их частоту повторения.

5.5.5 Индивидуальное задание 5 студента:

5.5.5.1 Сформировать и напечатать матрицу $A_{6 \times 6}$ целых случайных чисел в диапазоне от 0 до 20. Расположить столбцы матрицы в порядке возрастания их сумм элементов. Напечатать новую матрицу.

5.5.5.2 Текст некоторой строки вводится в режиме диалога и содержит простое предложение, в котором слова отделяются символом «пробел». Сформировать новую строку, в которой слова строки расположены в порядке возрастания в них числа букв.

5.5.5.3 Сформировать матрицу $A_{10 \times 10}$ случайных целых чисел в диапазоне от 0 до 20. Напечатать матрицу. Найти и напечатать номер строки, в которой больше всего одинаковых чисел.

5.5.6 Индивидуальное задание 6 студента:

5.5.6.1 Сформировать матрицу $A_{6 \times 6}$ случайных чисел в диапазоне от 0 до 20. Напечатать ее. Построить и напечатать вектор из повторяющихся чисел матрицы.

5.5.6.2 Строка, введенное в режиме диалога, содержит простое предложение, в котором слова отделяются символом «пробел». Определить есть ли в тексте одинаковые слова. Напечатать их с указанием частоты встречаемости в предложении.

5.5.6.3 Сформировать и напечатать матрицу $A_{8 \times 8}$ целых случайных чисел в диапазоне от минус 50 до 50. Напечатать все диагонали матрицы параллельные главной.

5.6 Контрольные вопросы для защиты отчета на СРСП

5.6.1 Понятие многомерного массива. Примеры.

5.6.2 Как объявляется и инициализируется многомерный массив? Пример.

5.6.3 Приведите пример и объясните как формировать матрицу $A_{5 \times 5}$ вещественных чисел в диапазоне от 0 до 10.

5.6.4 Приведите пример и объясните как организация вывода матрицы $A_{5 \times 5}$ на экран монитора.

5.6.5 Объясните алгоритм поиска максимального числа в матрице $A_{5 \times 5}$.

5.6.6 Понятие массива массивов. Объявление и инициализация таких массивов. Пример.

5.6.7 Объясните алгоритм сложения двух матриц. Пример.

5.6.8 Объясните алгоритм умножения двух матриц. Пример.

5.6.9 Понятие строковой переменной в языке C#. Пример.

5.6.10 Для чего предназначены escape-последовательностей в языке C#? Примеры.

5.6.11 Как сравниваются строковые переменные в языке C#? Пример.

5.6.12 Как можно удалить из текста некоторый известный фрагмент? Пример.

5.6.13 Как можно вставить в текст некоторый известный фрагмент? Пример.

5.6.14 Как можно определить количество символов в некотором тексте? Пример.

5.6.15 Назначение методов Split и Join. Примеры.