**Лекция 8. Многомерные массивы**

**Цель лекции:** Формирование у студентов знаний, связанных с использованием многомерных массивов.

В результате изучения данной лекции студенты будут способны:

* Продемонстрировать знание синтаксиса объявления и инциализации многомерных массивов

В программировании чаще всего применяются одномерные массивы, хотя и многомерные не так уж и редки. *Многомерным* называется такой массив, который отличается двумя или более измерениями, причем доступ к каждому элементу такого массива осуществляется с помощью определенной комбинации двух или более индексов.

**Двумерные массивы**

Простейшей формой многомерного массива является двумерный массив. Местоположение любого элемента в двумерном массиве обозначается двумя индексами. Такой массив можно представить в виде таблицы, на строки которой указывает один индекс, а на столбцы — другой.

В следующей строке кода объявляется двумерный массив integer размерами 10×20.

int[,] table = new int[10, 20];

Обратите особое внимание на объявление этого массива. Как видите, оба его размера разделяются запятой. В первой части этого объявления синтаксическое обозначение [,] означает, что создается переменная ссылки на двумерный массив. Если же память распределяется для массива с помощью оператора new, то используется следующее синтаксическое обозначение.

int[10, 20]

В данном объявлении создается массив размерами 10×20, но и в этом случае его размеры разделяются запятой.

Для доступа к элементу двумерного массива следует указать оба индекса, разделив их запятой. Например, в следующей строке кода элементу массива table с координатами местоположения (3,5) присваивается значение 10.

table[3, 5] = 10;

Ниже приведен более наглядный пример в виде небольшой программы, в которой двумерный массив сначала заполняется числами от 1 до 12, а затем выводится его содержимое.

// Продемонстрировать двумерный массив.

using System;

class TwoD {

static void Main() {

int t, i;

int[,] table = new int[3, 4];

for(t=0; t < 3; ++t) {

for(i=0; i < 4; ++i) {

table[t,i] = (t\*4)+i+1;

Console.Write(table[t,i] + " ");

}

Console.WriteLine();

}

}

}

В данном примере элемент массива table[0,0] будет иметь значение 1, элемент массива table[0,1] — значение 2, элемент массива table[0,2] — значение 3 и т.д.

А значение элемента массива table[2,3] окажется равным 12.

**Массивы трех и более измерений**

В C# допускаются массивы трех и более измерений. Ниже приведена общая форма объявления многомерного массива.

*тип*[*,...,*] *имя\_массива = new тип*[*размер1, размер2, ... размерN*];

Например, в приведенном ниже объявлении создается трехмерный целочисленный массив размерами 4×10×3.

int[,,] multidim = new int[4, 10, 3];

А в следующем операторе элементу массива multidim с координатами местоположения (2,4,1) присваивается значение 100.

multidim[2, 4, 1] = 100;

Ниже приведен пример программы, в которой сначала организуется трехмерный массив, содержащий матрицу значений 3×3×3, а затем значения элементов этого массива суммируются по одной из диагоналей матрицы.

// Суммировать значения по одной из диагоналей матрицы 3×3×3.

using System;

class ThreeDMatrix {

static void Main() {

int[,,] m = new int[3, 3, 3];

int sum = 0;

int n = 1;

for(int x=0; x < 3; x++)

for(int y=0; у < 3; y++)

for(int z=0; z < 3; z++)

m[x, y, z] = n++;

sum = m[0, 0, 0] + m[1, 1, 1] + m[2, 2, 2];

Console.WriteLine("Сумма значений по первой диагонали: " + sum);

}

}

Вот какой результат дает выполнение этой программы.

Сумма значений по первой диагонали: 42

**Инициализация многомерных массивов**

Для инициализации многомерного массива достаточно заключить в фигурные скобки список инициализаторов каждого его размера. Ниже в качестве примера приведена общая форма инициализации двумерного массива:

*тип*[,] *имя\_массива* = {

{val, val, val, ..., val},

{val, val, val, ..., val},

{val, val, val, ..., val}

};

где *val* обозначает инициализирующее значение, а каждый внутренний блок — отдельный ряд. Первое значение в каждом ряду сохраняется на первой позиции в массиве, второе значение — на второй позиции и т.д. Обратите внимание на то, что блоки инициализаторов разделяются запятыми, а после завершающей эти блоки закрывающей фигурной скобки ставится точка с запятой.

В качестве примера ниже приведена программа, в которой двумерный массив sqrs инициализируется числами от 1 до 10 и квадратами этих чисел.

// Инициализировать двумерный массив.

using System;

class Squares {

static void Main() {

int[,] sqrs = {

{ 1, 1 },

{ 2, 4 },

{ 3, 9 },

{ 4, 16 },

{ 5, 25 },

{ 6, 36 },

{ 7, 49 },

{ 8, 64 },

{ 9, 81 },

{ 10, 100 }

};

int i, j;

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

for(j=0; j < 2; j++)

Console.Write(sqrs[i,j] + " ");

Console.WriteLine();

}

}

}

При выполнении этой программы получается следующий результат.

1 1

2 4

3 9

4 16

5 25

6 36

7 49

8 64

9 81

10 100