

Тема: *Многомерные массивы*

Мария Александровна Сокольская

Многомерные массивы и основные задачи работы с ними

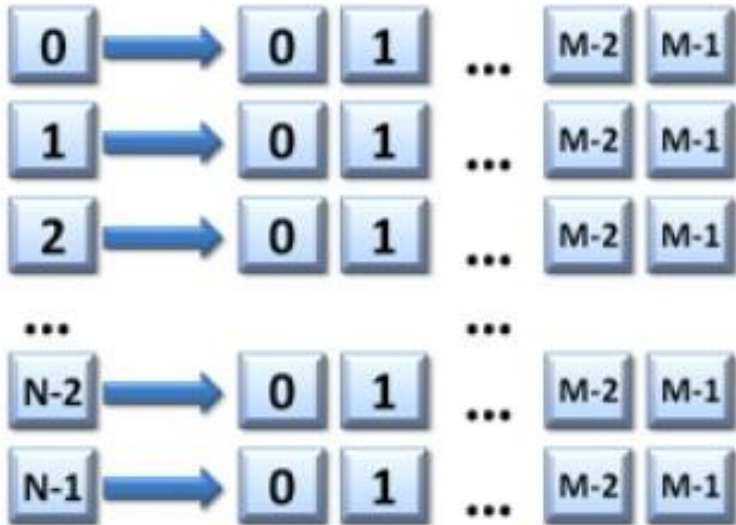


Определение: многомерный массив – это массив, который в качестве своих элементов *содержит другие массивы*.

Задачи:

1. Выделить память нужного размера под массив
2. Записывать данные в нужную ячейку
3. Читать данные из ячейки

Двумерные массивы



Двумерные массивы представляются в виде матрицы, каждая строка которой – обычный одномерный массив, а объединение строк – двумерный массив

Трёхмерный массив можно представить как набор матриц.

Для обращения к элементу *n*-мерного массива, нужно указать *n* индексов, т.е. для обращения к элементу **двумерного** массива, требуется **два** индекса

Объявление двумерных массивов



Синтаксис:

<тип> <имя массива>[][];

или

<тип> [][] <имя массива>;

Например:

int array[][], i; //array – массив, i – переменная

float [][]array1, f; //array1, f - массивы

Создание двумерных массивов



Синтаксис:

<имя массива> = new **<тип_элементов_массива>**
[количество_строк][количество_столбцов]

array = new int [10][10]; //создаем таблицу 10x10

Можно совместить объявление массива с созданием:

char str[][] = new char [5][20];

Инициализация элементов массива.



По умолчанию Java заполняет массив нулями. Можно заполнить массив собственными значениями:

```
int array[][] = {{2, 3, 6} {5, 7, 9} {8, 9, 1}};
```

Количество **строк** двумерного массива сохраняется в свойстве `length`

```
array.length
```

Для получения длины конкретной строки нужно указать ее индекс:

```
array[i].length
```

Строки матрицы можно переставлять:

```
int []temp=array[i];
```

```
array[i]=array[i+1];
```

```
array[i+1]=temp;
```

Доступ к элементам массива



Синтаксис:

<имя массива>[номер строки][номер столбца]

Например:

*array[2][3]=12; /*присвоить элементу массива на пересечении второй строки и третьего столбца значение 12.*/*

Обработка двумерных массивов



Для обработки двумерных массивов используются два вложенных друг в друга цикла с разными счетчиками

Программные блоки работы в двумерными массивами:



Пример. Заполнение двумерного массива



```
int array [][]= new int [6] [4];  
Random rand = new Random();  
//блок ввода массива  
for(int i=0; i<6; i++){ //цикл по строкам  
    for (int j=0; j<4; j++){ //цикл по столбцам  
        array [i] [j] = rand.nextInt(10);  
    }  
}
```

Пример. Замена четных элементов на 0, а нечетных на 1



//блок обработки массива

```
for(int i=0; i<6; i++){  
    for(int j=0; j<4; j++){  
        if (array[i][j]%2==0)  
            array[i][j]=0;  
        else  
            array[i][j]=1;  
    }  
}
```



Пример. Вывод массива

//блок вывода массива

```
for(int i=0; i<6; i++){  
    for(int j=0; j<4; j++){  
        out.print(array[i][j]+" ");  
    }  
    out.println();  
}
```

Задание:

Соберите блоки в единую работающую программу

Задача



Найти минимум и максимум в двумерном массиве.

Массив заполнить случайными числами.

Доп. задание: вывести **индексы** минимума и максимума (например: “min=1 у элемента с индексами array[2][3]”)

Основные задачи, встречающиеся при работе с двумерными массивами



1. Перестановка элементов строки/столбца
2. Вычисление суммы элементов строки/столбца
3. Нахождение часто повторяющегося значения
4. Нахождение среднего арифметического и среднего геометрического значений элементов строки/столбца
5. Заполнение массива по определенному закону
6. Подсчет значений (положительных или отрицательных, четных или нечетных и т.д.)
7. Перенос элементов из одной части массива в другую
8. Сортировки
9. Поиск заданного элемента

Задание



1. № 354

2. № 355

3. № 361

4. № 363

5*. Напишите программу автоматической расстановки кораблей для игры «Морской бой» (можно ввести обозначение: 0 – клетка свободна, 1 – клетка занята кораблем).