

Двумерные массивы

Снова здравствуйте. На этот раз продолжение прошлой темы – массивы. Теперь рассмотрим двумерные массивы. Если обычные (одномерные) массивы можно объяснить как список, то двумерные можно представить в виде таблицы, у нас имеется два измерения, то есть, по-сути, появляются строки и столбцы.

Приведём пример. Допустим у нас есть двумерный массив целых чисел. Для обозначения двумерных массивов мы просто добавляем ещё одну квадратную скобку в объявлении.

```
int[][] arr = new int[5][5];
```

Здесь всё так же, как с одномерными, просто добавляется ещё одно измерение, мы точно так же сначала определяем тип данных, затем две квадратные скобки, чтобы показать, что это именно двумерный массив, затем название “arr”, после чего в скобках мы уже указываем размер, то есть в этом массиве будет 5 столбцов и 5 строк (итого 25 элементов).

Если мы заполним его числами и выведем как таблицу, выглядеть это будет примерно так:

2	3	1	2	6
1	7	8	2	9
2	1	4	3	4
2	3	6	8	2
4	2	3	1	1

Как видим, да, это по своей сути таблица, да и понимать двумерные массивы именно в таком виде намного легче.

Здесь особо не о чем распинаться, но всё же тема достойна упоминания, у вас явно будут задания с этим связанные (если ещё не были на момент прочтения).

Чтобы нам заполнить такой массив вручную или даже с помощью случайных чисел (так как вручную будет долго), нам нужен уже не один цикл for, а целых два, причём один будет внутри другого. Выглядеть это должно вот так:

```
for(int i = 0; i < 5; i++) {  
    for(int j = 0; j < 5; j++) {  
        Инструкции к заполнению, например “arr[i][j] = rand.nextInt(6);”  
    }  
}
```

Теперь разбираемся, что здесь происходит:

1) Создаём цикл, в котором создаём переменную, которая будет увеличиваться каждое повторение цикла, пока не приблизится к количеству “столбцов”;

2) Внутри этого цикла создаём такой же, но с другой переменной (название её должно быть другим, так как, хоть это и временная переменная, у нас цикл в цикле), которая делает то же самое, но она уже будет приближаться к количеству “строк” (у нас столбцов и строк поровну, но надеюсь ты понимаешь принцип, если нет, всегда готов объяснить);

3) Здесь уже задаём инструкции к заполнению нашего массива, причём стоит отметить, как мы обращаемся к элементам массива, если в обычных массивах мы просто писали название и индекс, то здесь уже надо писать два индекса в отдельных квадратных скобках (индексы так же начинаются с нуля);

Здесь я понимаю, что может возникнуть вопрос “Зачем нам два цикла и что они вообще перебирают?”, ответ настолько же прост, насколько к этому сложно прийти самому изначально.

Дело в том, что первый цикл перебирает столбцы, то есть первый индекс, а второй цикл перебирает строки.

Как это можно представить? Сначала у нас берётся первый столбец, это первое повторение первого цикла, и уже в этом всё начинают перебирать строки этого столбца. Как только строки заканчиваются, второй цикл изжил себя и столбец переключается на следующий (то есть `i++`), там происходит то же самое, перебор строк и так далее. Надеюсь хоть немного понятно.

Для вывода массива нам нужно просто создать такую же конструкцию, но вместо инструкций к заполнению, вписать наш любимый “`System.Out.print();`”.

Здесь заметь, мы пишем именно “`print`”, а не “`println`”, так как если мы будем выводить каждый элемент на новую строку, их будет супер много и это будет неудобно (если элементов мало, такое может быть нормально в целом). Таким образом мы выведем просто все элементы в одну строку, но как нам вывести их в виде именно таблицы, как я показывал в примере? Всё просто, после второго цикла добавить пустой “`System.Out.println();`”, чтобы после каждого раза, когда мы проходим все столбцы, оно просто спускалось на строку вниз и продолжало уже там. Финальный вид вывода будет примерно таким:

```
for (int i = 0; i < 5; i++) {  
    for (int j = 0; j < 5; j++) {  
        System.out.print(arr[i][j] + " ");  
    }  
    System.out.println();  
}
```

Зачем там пробел после каждого элемента? Чтобы элементы не слиплись в одно большое число/слово. Готовый код с заполнением и выводом будет в файле с ответами самым первым, чтобы ты могла посмотреть перед тем, как практиковаться.

На этом теорию в целом можно закончить, единственное хочу добавить, что таким же образом можно делать и трёхмерные массивы, и большей размерности, просто добавляя квадратные скобки в создании массива. Как представить себе трёхмерную таблицу? В целом можно, конечно, что-то с координатными плоскостями придумать (что-то вроде маленьких кубиков, из которых состоит большой куб), но обычно дальше двумерных массивов всё равно не заходят, так как надобности особо нет, не забивай особо этим голову сейчас.

Перейдём к задачкам, я их дам немного, так как тема небольшая, да и если прям напрягаться в ней, можно с ума сойти спокойно.

1) Для начала тебе нужно самостоятельно создать двумерный массив целых чисел (можешь взять и дробные, если хочешь), размеры массива можешь выбрать любые, но для наглядности и быстроты работы кода лучше не брать слишком много, а то прописывать тебе несколько тысяч элементов, например он может долго иногда. Дальше тебе нужно с помощью циклов, описанных выше, заполнить массив числами так, чтобы каждое последующее изменялось одинаково относительно предыдущего (например, чтобы к нему просто каждый раз прибавлялось одно и то же число или можно использовать умножение, здесь у тебя простор фантазии). После заполнения массива тебе нужно вывести его в виде таблицы с помощью циклов, как я уже показал.

Ответ будет в файле с ответами, но если возникнут проблемы, можешь спрашивать.

Ну и ещё одной, думаю, хватит после такой сложной темы.

2) Создай двумерный массив, состоящий из строк. Далее сделай, чтобы пользователь ввёл все значения сам через консоль (для такого лучше делать небольшой массив). Теперь через сложение строк и циклы тебе нужно сложить все элементы массива в одну большую строку (такое предложение чтобы получилось бессмысленное) и затем сделать вывод этой строки. Выводить массив в целом не надо, но если захочешь попрактиковать вложенные циклы, то конечно почему нет.

Ответ в файле с ответами. Удачи.