

Массивы

В отличие от обычных переменных, которые хранят ровно одно значение, массивы используются для хранения целого набора значений. Количество значений в массиве называется его *размером* или *длиной*, а сами значения - *элементами* массива. Значений может не быть вовсе – в этом случае массив считается пустым, а его длина равной нулю.

Элементы массива не имеют собственных имен, доступ к ним осуществляется при помощи индексации. Если массив имеет длину N, отличную от нуля, то корректными значениями индекса являются числа от 0 до N-1. Значение индекса массива имеет тип `int`, следовательно, максимально возможный размер массива равняется максимальному значению типа `int` – 2147483647. При обращении к элементу можно также использовать значения типов `byte`, `short` или `char`, поскольку эти типы автоматически расширяются до `int`. Попытка использовать в качестве индекса значение типа `long` приведет к ошибке времени компиляции.

Все значения массива имеют одинаковый тип, и говорится, что массив основан на этом *базовом типе*. Массивы могут быть основаны как на примитивных, так и на ссылочных типах данных. Объявление массива похоже на объявление обычной переменной: сначала указывается базовый тип, затем идет имя массива. На то, что используемый тип является именно массивом, указывает пара квадратных скобок:

базовый тип имя массива [];

Также существует альтернативная форма описания массива – пара квадратных скобок располагается между базовым типом и именем массива:

базовый тип [] имя массива;

Создание переменной типа массив еще не создает экземпляры этого массива. Такие переменные имеют объектный тип и хранят ссылки на объекты, однако изначально имеют значение `null`. Чтобы создать экземпляр массива, нужно воспользоваться оператором выделения памяти – записать ключевое слово `new`, после чего указать тип массива, и в квадратных скобках - размер массива:

имя массива = new базовый тип [размер массива];

Также массив можно инициализировать сразу при объявлении массива:

базовый тип имя массива [] = new базовый тип [размер массива];

базовый тип [] имя массива = new базовый тип [размер массива];

При подобной инициализации все элементы массива имеют значение по умолчанию. Для числовых типов это значение 0, для типа `boolean` – значение `false`, а для остальных объектов - значение `null`. Однако для инициализации массива также можно задать конкретные значения его элементов, перечислив их в фигурных скобках через запятую:

базовый тип имя массива [] = {элементы массива};
базовый тип [] имя массива= {элементы массива};
базовый тип имя массива [] = new базовый тип []{элементы массива};
базовый тип [] имя массива = new базовый тип []{элементы массива};

В данном случае в квадратных скобках не указывается размер массива, так как он вычисляется автоматически по количеству элементов в фигурных скобках.

Базовый тип также может быть и массивом – таким способом формируется массив массивов, или *многомерный массив*. В этом случае для доступа к элементам каждого из массивов используется собственный индекс, а количество индексов называется *размерностью* массива. В отличие от размера, размерность массива ограничивается только объёмом доступной для приложения памяти.

Если создать двумерный массив и определить переменную *x*, которая на него ссылается, то, используя *x* и два числа в паре квадратных скобок каждое (например, *x[0][0]*), можно обратиться к любому элементу двумерного массива. Но, в то же время, используя *x* и одно число в паре квадратных скобок (например, *x[0]*), можно обратиться к одномерному массиву, который является элементом двумерного массива. Его можно проинициализировать новым массивом с некоторой другой длиной, и массив перестанет быть прямоугольным – он примет произвольную, *ступенчатую* форму. Таким образом, более правильный подход заключается в том, что в языке Java нет двумерных, и вообще многомерных массивов, а есть массивы, базовыми типами которых являются также массивы. Например, тип *int[]* означает «массив чисел», а *int[][]* означает «массив массивов чисел». Выражение *int x[][] = new int[2][3]* описывает три объекта: один массив массивов размером 2 и два массива целых чисел размером 3.

При создании многомерных массивов с помощью операции *new* необходимо указывать все пары квадратных скобок соответственно количеству измерений. Но заполненной обязательно должна быть лишь самая левая пара – это значение задаст длину самого верхнего массива массивов. Если заполнить следующую пару, то этот массив заполнится не значениями по умолчанию *null*, а новыми созданными массивами с меньшей на единицу размерностью. Если заполнена вторая пара скобок, то можно заполнить третью, и так далее. Например, запись *int x[][]=new int[3][];* описывает один объект - массив массивов размером 3, и заполняет его значениями *null*. Далее каждый из трёх элементов данного массива возможно проинициализировать по отдельности массивом целых чисел произвольного размера.

Аналогично, для создания многомерных массивов можно использовать инициализаторы. В этом случае используется столько вложенных пар фигурных скобок, сколько требуется. Например, запись *int i[][] = { {1,2}, null, {3}, {} };* описывает четыре объекта – один массив массивов длиной 4 и три массива чисел с длинами 2, 1, 0 соответственно.