

Практическая работа №6

Использование массивов при решении вычислительных задач

Массив представляет набор однотипных значений. Объявление массива похоже на объявление обычной переменной, которая хранит одиночное значение, причем есть два способа объявления массива:

```
тип_данных название_массива [];
```

```
тип_данных [] название_массива;
```

Определим массив чисел:

```
int nums[];
```

После объявления массива мы можем инициализировать его:

```
nums = new int[4]; // массив из 4 чисел
```

Создание массива производится с помощью следующей конструкции:

```
new тип данных [количество элементов],
```

где new - ключевое слово, выделяющее память для указанного в скобках количества элементов. Например, `nums = new int[4];` - в этом выражении создается массив из четырех элементов `int`, и каждый элемент будет иметь значение по умолчанию – число 0.

Также можно инициализировать массив сразу при его объявлении:

```
int nums[] = new int[4]; // массив из 4 чисел
```

```
int[] nums2 = new int[5]; // массив из 5 чисел
```

При подобной инициализации все элементы массива имеют значение по умолчанию. Для числовых типов (в том числе для типа `char`) это число 0, для типа `boolean` это значение `false`, а для остальных объектов – значение `null`. Для типа `int` значением по умолчанию является число 0, поэтому определенный выше массив `nums` будет состоять из четырех нулей. Однако также можно задать конкретные значения для элементов массива при его создании:

// эти два способа равнозначны

```
int[] nums = new int[] { 1, 2, 3, 5 };
```

```
int[] nums2 = { 1, 2, 3, 5 };
```

Стоит отметить, что в этом случае в квадратных скобках не указывается размер массива, так как он вычисляется по количеству элементов в фигурных скобках.

После создания массива мы можем обратиться к любому его элементу по индексу, который передается в квадратных скобках после названия переменной массива:

```
int[] nums = new int[4];  
// устанавливаем значения элементов массива  
nums[0] = 1;  
nums[1] = 2;  
nums[2] = 4;  
nums[3] = 100;  
// получаем значение третьего элемента массива  
System.out.println(nums[2]); // 4
```

Индексация элементов массива начинается с 0, поэтому, чтобы обратиться к третьему элементу в массиве, нам надо использовать выражение `nums[2]`.

Так как у нас массив определен только для 4 элементов, мы не можем обратиться, например, к шестому элементу: `nums[5] = 5;`. Если мы так попытаемся сделать, то мы получим ошибку.

Важнейшее свойство, которым обладают массивы, является свойство `length`, возвращающее длину массива, то есть количество его элементов:

```
int[] nums = { 1, 2, 3, 4, 5 };  
int length = nums.length; // 5
```

Нередко бывает неизвестным последний индекс, и чтобы получить последний элемент массива, мы можем использовать это свойство:

```
int last = nums[nums.length-1];
```

Ранее мы рассматривали одномерные массивы, которые можно представить как цепочку или строку однотипных значений. Но массивы также могут быть и многомерными. Наиболее известный многомерный массив - таблица, представляющая двумерный массив:

```
int[][] nums2 = { { 0, 1, 2 }, { 3, 4, 5 } };
```

Поскольку массив `nums2` двумерный, он представляет собой простую таблицу. Его также можно было создать следующим образом: `int[][] nums2 = new int[2][3];`. Количество квадратных скобок указывает на размерность массива. А числа в скобках - на количество строк и столбцов.

Также, используя индексы, мы можем использовать элементы массива в программе:

```
// установим элемент первого столбца второй строки  
nums2[1][0]=44;  
System.out.println(nums2[1][0]);
```

Многомерные массивы могут быть также представлены как «зубчатые массивы». В вышеприведенном примере двумерный массив имел 3 строки и три столбца, поэтому у нас получалась ровная таблица. Но мы можем каждому элементу в двумерном массиве присвоить отдельный массив с различным количеством элементов:

```
int[][] nums = new int[3][];  
nums[0] = new int[2];  
nums[1] = new int[3];  
nums[2] = new int[5];
```

Специальная версия цикла for предназначена для перебора элементов в наборах элементов, например, в массивах и коллекциях. Она аналогична действию цикла foreach , который имеется в других языках программирования. Формальное ее объявление:

```
for (тип_данных название_переменной : контейнер){  
    // действия  
}
```

Например:

```
int[] array = new int[] { 1, 2, 3, 4, 5 };  
for (int a : array){  
    System.out.println(a);  
}
```

В качестве контейнера в данном случае выступает массив данных типа int. Затем объявляется переменная с типом int. То же самое можно было бы сделать и с помощью обычной версии for:

```
int[] array = new int[] { 1, 2, 3, 4, 5 };  
for (int i = 0; i < array.length; i++){  
    System.out.println(array[i]);  
}
```

В то же время, эта версия цикла for более гибкая по сравнению for (int i : array). В частности, в этой версии мы можем изменять элементы:

```
int[] array = new int[] { 1, 2, 3, 4, 5 };  
for (int i=0; i<array.length;i++){  
    array[i] = array[i] * 2;  
    System.out.println(array[i]);  
}
```

Перебор многомерных массивов в цикле:

```
int[][] nums = new int[][]  
{  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9}  
};  
for (int i = 0; i < nums.length; i++){  
    for(int j=0; j < nums[i].length; j++){  
        System.out.printf("%d ", nums[i][j]);  
    }  
    System.out.println();  
}
```

Сначала создается цикл для перебора по строкам, а затем внутри первого цикла создается внутренний цикл для перебора по столбцам конкретной строки. Подобным образом можно перебирать и трехмерные массивы, и наборы с большим количеством размерностей.

Практические задания

1. Дан одномерный массив целых чисел. Вывести на экран:
 - а) сначала его четные элементы, затем нечетные;
 - б) сначала его положительные элементы, затем отрицательные;
 - в) сначала его однозначные значения, затем двузначные.
2. Дан одномерный массив целых чисел. Найти количество элементов, значение которых больше среднего арифметического минимального и максимального элементов массива, и напечатать их порядковые номера.
3. Дан одномерный массив вещественных чисел. Все его положительные значения увеличить на единицу, отрицательные – уменьшить на единицу, нулевые – заменить средним арифметическим всех элементов массива.
4. Дан двумерный массив целых чисел. Определить:
 - а) максимальный элемент массива и его координаты. Если элементов с максимальным значением несколько, то должны быть найдены координаты самого верхнего и самого левого из них;
 - б) минимальный элемент массива и его координаты. Если элементов с минимальным значением несколько, то должны быть найдены координаты самого нижнего и самого правого из них.
5. Дан двумерный массив целых чисел, состоящий из n строк и n столбцов. Выяснить, является ли данный массив магическим квадратом. В магическом квадрате суммы элементов по всем строкам, столбцам и двум диагоналям равны.