

Лабораторная работа №7: Массивы

Цели и задачи

1. Научиться использовать перечисления (enum)
2. Узнать о применении перечислений

Начало работы

Убедитесь в работоспособности вашего ПК. При обнаружении неполадок, сообщите преподавателю.

Для работы вам потребуется:

1. Среда разработки Eclipse (возможно так же использование другой IDE)
2. Java Development Kit 10 и выше.
3. Тетрадь и ручка для записи важных моментов

Теоретическая часть

Массив (англ. Array) это объект, хранящий в себе фиксированное количество значений одного типа. Другими словами, массив — это нумерованный набор переменных. Переменная в массиве называется элементом массива, а ее позиция в массиве задается индексом. Например, нам нужно хранить 50 различных имен, согласитесь, неудобно для каждого имени создавать отдельную переменную, поэтому мы будем использовать массив. Нумерация элементов массива начинается с 0, а длина массива устанавливается в момент его создания и фиксируется.

Для того чтобы создать массив нужно его объявить, зарезервировать для него память и инициализировать.

Объявление массива

При создании массива в Java первым делом его нужно объявить. Это можно сделать следующим образом:

```
int[] myFirstArray;
```

Можно также объявить массив так:

```
int mySecondArray[];
```

Однако, это не приветствуется соглашением по оформлению кода в Java, поскольку скобки обозначают то, что мы имеем дело с массивом и логичнее, когда они находятся рядом с обозначением типа.

Исходя из данного примера, мы объявили 2 массива с именами myFirstArray и mySecondArray. Оба массива будут содержать элементы типа int.

Подобным образом можно объявить массив любого типа:

```
byte[] anArrayOfBytes;  
short[] anArrayOfShorts;  
long[] anArrayOfLongs;  
float[] anArrayOfFloats;  
double[] anArrayOfDoubles;  
boolean[] anArrayOfBooleans;  
char[] anArrayOfChars;  
String[] anArrayOfStrings;  
...  
...
```

Тип массива задается следующим образом `type[]`, где `type` это тип данных содержащихся в нем элементов. Скобки являются специальным обозначением того, что переменные содержатся в массиве. Имя массива может быть любым, однако, оно должно соответствовать правилам именования переменных.

Массивы можно создавать не только из переменных базовых типов, но и из произвольных объектов.

При объявлении массива в языке Java не указывается его размер и не резервируется память для него. Происходит лишь создание ссылки на массив.

Резервация памяти для массива и его инициализация

Далее, для объявленного `myFirstArray`, зарезервируем память при помощи ключевого слова `new`.

```
int[] myFirstArray;  
myFirstArray = new int[15];
```

В нашем примере мы создали массив из 15 элементов типа `int` и присвоили его ранее объявленной переменной `myFirstArray`.

Объявлять имя массива и резервировать для него память также можно на одной строке.

```
int[] myArray = new int[10];
```

При создании массива с помощью ключевого слова `new`, все элементы массива автоматически инициализированы нулевыми значениями. Для того, чтобы присвоить элементам массива свои начальные значения, необходимо провести его инициализацию. Инициализацию можно проводить как поэлементно

```
myFirstArray[0] = 10; // инициализация первого элемента  
myFirstArray[1] = 20; // инициализация второго элемента  
myFirstArray[2] = 30; // и т.д.
```

так и в цикле, с помощью индекса проходя все элементы массива и присваивая им значения.

```
for(int i = 0; i < 15; i++){  
    myFirstArray[i] = 10;  
}
```

Как видно из предыдущих примеров, для того, чтобы обратиться к элементу массива, нужно указать его имя и, затем, в квадратных скобках — индекс элемента. Элемент массива с конкретным индексом ведёт себя также, как и переменная.

Рассмотрим создание и инициализацию массива на следующем примере. В нем мы создаем массив, содержащий цифры 0-9 и выводим значения на консоль.

```
//создание и инициализация массива  
int[] numberArray = new int[10];  
for(int i = 0; i < 10; i++){  
    numberArray[i] = i;  
}  
//вывод значений на консоль  
for(int i = 0; i < 10; i++){  
    System.out.println((i+1) + "-й элемент массива = " + numberArray[i]);  
}
```

Упрощенная форма записи

Для создания и инициализации массива можно также использовать упрощенную запись. Она не содержит слово `new`, а в скобках перечисляются начальные значения массива.

```
int[] myColor = {255, 255, 0};
```

Здесь длина массива определяется числом значений, расположенных между скобками и разделенных запятыми. Такая запись больше подходит для создания небольших массивов.

Определение размера массива

Размер массива не всегда очевиден, поэтому для того, чтобы его узнать следует использовать свойство `length`, которое возвращает длину массива.

```
myColor.length;
```

Данный код поможет нам узнать, что длина массива `myColor` равна 3.

Пример

Задано 4 числа, необходимо найти минимальное

```
int[] numbers = {-9, 6, 0, -59};
int min = numbers[0];
for(int i = 0; i < numbers.length; i++){
    if(min > numbers[i])
        min = numbers[i];
}
System.out.println(min); // -59
```

Самостоятельная работа

1. Создайте массив, содержащий 10 первых нечетных чисел. Выведете элементы массива на консоль в одну строку, разделяя запятой.
2. Дан массив размерности N, найти наименьший элемент массива и вывести на консоль (если наименьших элементов несколько — вывести их все).
3. Найти среднее арифметическое всех элементов массива.