

Здравствуйтесь, второй урок будет по массивам. Вам уже давали задание, где массивы необходимы, поэтому я посчитал нужным это разобрать.

Начнём с того, что массив – это просто куча однотипных данных, собранных в список (грубо говоря) для удобства использования. В целом это нужно, чтобы не создавать миллиард переменных, а просто обращаться к определённому элементу массива. Как ты уже скорее всего знаешь, элементам в массиве присваивается индекс (по-сути номер элемента), и начинается отсчёт с нуля, то есть первый элемент массива будет под индексом «0».

Это мы разобрали, теперь к определению массива. Чтобы создать массив, нужно задать тип данных этого массива, задать его имя и, если он известен, размер массива, получим:

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

Где:

- 1) `int[]` – тип данных (квадратные скобки обозначают, что это именно массив);
- 2) `array` – название массива (можно использовать любое, кроме системных названий);
- 3) `10` – размер массива (то есть индексы будут от 0 до 9).

С этим разобрались, это одномерные массивы (можно представить как просто список чего-то), но есть так же и двумерные, трёхмерные и т.д. Но сейчас не будем забивать тебе этим голову.

И так начнём переходить к задачам

1) Первым заданием будет создание массива с 5 элементами (целые числа, то есть – `int`), после чего надо будет, чтобы пользователь ввёл значение каждого элемента, а в конце можешь сделать на выбор, либо чтобы код вывел значение третьего элемента массива (помним, что отсчёт идёт с нуля, значит третий элемент будет под номером «2»), либо чтобы он сложил все элементы массива и вывел полученное значение.

Небольшое пояснение, чтобы не ломать голову на первом же задании – здесь нужно использовать цикл `for`, чтобы перебирать каждый элемент массива. Пример:

```
for (int i = 0; i < 5; i++) {  
    arr[i] = inp.nextInt();  
}
```

Здесь мы создаём переменную `i`, которой даём начальное значение 0 (так как первый элемент под индексом 0), далее даём условие, что `i < 5`, так как у нас 5 элементов массива, после чего задаём изменение переменной, чтобы после каждого ввода числа мы переходили к следующему элементу массива.

Само действие внутри цикла заключается в следующем – мы берём элемент массива под номером `i` и присваиваем ему значение, введённое с консоли (то есть через `Scanner`).

Здесь:

- 1) `arr[i]` – название массива и индекс элемента;
- 2) `inp` – произвольное название сканнера;
- 3) `nextInt` – параметр, считывающий целые числа.

Решение задачи будет в файле с ответами (оба варианта с выводом).

Вторым заданием будет такое

2)Создать массив чисел типа int. Размер массива пусть пользователь введёт сам, то есть сначала будет создание переменной, которую введёт пользователь, а потом уже создание массива. Далее нужно через цикл задать значение всем элементам массива и вывести наибольший и наименьший элемент (для этого нужно использовать команду Arrays.sort(arr);), но на этот раз пусть каждый элемент будет случайным, то есть используем импортный оператор Random. Чтобы его вызвать, нужно:

1)Прописать в начале наравне с импортом сканнера, вот такую строку:

```
import java.util.Random;
```

и ещё сразу подключить обработчик массивов, то есть

```
import java.util.Arrays;
```

2)В самом коде создать новый рандом через вот такой код:

```
Random rand = new Random();
```

3)Вызвать этот рандом, когда присваиваем значение переменной, например:

```
int a = rand.nextInt();
```

4)Для сортировки массива пишем:

```
Arrays.sort(arr);
```

Таким образом массив отсортируется по возрастанию, а значит наименьшее число будет первым элементом, а наибольшее – последним.

В джаве с рандомом есть проблема небольшая с определением границ генерации числа, так как например с целыми числами мы можем задать границу только максимального числа, например

```
rand.nextInt(25);
```

Здесь будет сгенерировано число от 0 до 25 (не включая 25), и это проблема, если, например нам надо сгенерировать отрицательное число, и да, мы можем для этого, например просто умножать случайное число на -1, но это костыль.

Однако да, есть вариант, как ввести границы генерации числа, но выглядят они супер не интуитивно, однако если тебе это когда-либо понадобится, то вот:

```
Math.floor(Math.random() *(max - min + 1) + min);
```

Здесь да, мы задаём минимально и максимальное значение, а также используем не импортный оператор Random, а математическую команду, такая тоже есть, да. В целом надеюсь такое нигде не понадобится, так как сделано это в джаве не очень, особенно при генерации чисел с плавающей точкой (то есть десятичных дробей), так как там диапазон вообще нельзя менять изначально, и он составляет от 0.0 до 1.0, а это не круто вообще, но для такого и приходят на помощь эти глупые формулы.

К счастью для нашего задания нужно целое число, поэтому просто пишем rand.nextInt() и не тратим нервы. Да и в целом не забывай голову такими проблемами с некоторыми аспектами языка, это просто надо принять иногда.

Решение задачи будет в файле с ответами конечно же. Понимаю, что в этом задании много новой информации может быть, поэтому если где-то возникнут вопросы или проблемы, обязательно обсуждаем это, а то я сам тут пока разбирался как что работает, немного запутался. И можешь спокойно посмотреть ответ раньше решения собственного, если прям будут проблемы. Главное в итоге понять, как работают эти элементы кода и закрепить собственным написанием.

В целом из этих двух заданий станет понятен принцип действия массивов, но для закрепления можешь сама поиграться с этим, например создав массив строк, чтобы получился такой словарь.

Либо можно ввести слово и через команду `charAt(i)` сделать массив символов, из которых состоит слово. Для этого понадобится цикл `for`, в котором будет перебор каждого символа в слове по очереди. После чего можно вывести слово в обратном порядке. Пример такого решения так же добавлю в файл с ответами.

Ещё можешь поиграться с командами, выводя длину массива, значение определённого элемента, всех чётных/нечётных элементов массива и т.д. Для получения длины массива нужно использовать команду `arr.length`.

В целом это всё по одномерным массивам, остальное дело практики больше, поэтому советую действительно поиграться самой с простыми задачками, которые описал выше, меняя параметры в них.