2. Algorithmization

***1. Дайте определение массиву. Как осуществляется индексация элементов массива. Как необходимо обращаться к i-му элементу массива.***

Массив — это структура данных, в которой хранятся элементы одного типа. Его можно представить, как набор пронумерованных ячеек, в каждую из которых можно поместить какие-то данные (один элемент данных в одну ячейку). Доступ к конкретной ячейке осуществляется через её номер. Номер элемента в массиве также называют индексом. Обращаеся arr[i].

***2. Приведите способы объявления и инициализации одномерных и двумерных массивов примитивных и ссылочных типов. Укажите разницу, между массивами примитивных и ссылочных типов.***

Объявить массив можно одним из двух способов. Они равноправны, но первый из них лучше соответствует стилю Java. Второй же — наследие языка Си. 1) dataType[] arrayName; 2) dataType arrayName[];

int[] myArray; // объявление массива

myArray = new int[10]; // создание, то есть, выделение памяти для массива на 10 элементов типа int

После создания массива с помощью new, в его ячейках записаны значения по умолчанию. Для численных типов (как в нашем примере) это будет 0, для boolean — false, для ссылочных типов — null.

Инициализация массива — это заполнение его конкретными данными (не по умолчанию). String[] seasons = new String[4]; seasons[0] = "Winter"; Инициализацию также можно провести по-другому, совместив с инициализацией и объявлением: String[] seasons = new String[] {"Winter", "Spring", "Summer", "Autumn"} или String[] seasons = {"Winter", "Spring", "Summer", "Autumn"}.

Многомерный массив объявляется и создается следующим образом: Int[][] myTwoDimentionalArray = new int [8][8];

***3. Объясните, что значит клонирование массива, как в Java можно клонировать массив, в чем состоит разница в клонировании массивов примитивных и ссылочных типов.***

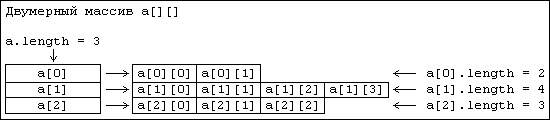
Существует несколько способов провести копирование массива: спользуя метод System.arraycopy; clone; Arrays.copyOf или Arrays.copyOfRange; провести копирование вручную в цикле.

int[] a = {1,2,3}; int[] b = a.clone();b[0] = 5; System.out.println(a[0]);  
System.out.println(b[0]);//Outputs:1 //5. А ссылочные типы изменяются везде.

***4. Объясните, что представляет собой двумерный массив в Java, что такое «рваный массив». Как узнать количество строк и количество элементов в каждой строке для «рваного» массива?***

Для двумерного массива не существует такого понятия как его размеры. Можно определить размер массива только по первому индексу. Причина кроется в организации массива. Он представляет собой массив ссылок на массивы, каждый из которых содержит реальные данные. И эти массивы могут иметь разную длину!

Рваный массив – это массив с разной длинной строк.



***5. Объясните ситуации, когда в java-коде могут возникнуть следующие исключительные ситуации java.lang.ArrayIndexOutOfBoundsException и java.lang.ArrayStoreException.***

ArrayIndexOutOfBoundsException – это исключение, появляющееся во время выполнения. Оно возникает тогда, когда мы пытаемся обратиться к элементу массива по отрицательному или превышающему размер массива индексу.

ArrayStoreException – это исключение, появляющееся во время выполнения. Оно возникает тогда, когда мы пытаемся присвоить элементу массива несовместимый тип.

***6. Объясните, зачем при кодировании разделять решаемую задачу на методы. Поясните, как вы понимаете выражение: «Один метод не должен выполнять две задачи».***

Методы с большими списками аргументов и множеством строк кода трудно модифицировать/отлаживать. Такие функции должны быть разбиты на несколько меньших методов, где каждый из них выполняет четко определенный шаг, который является частью более крупной задачи.

***7. Объясните, как в Java передаются параметры в методы, в чем особенность передачи в метод значения примитивного типа, а в чем ссылочного.***

Примитивные типы передаются по значению. Объекты передаются в метод по ссылке. Поэтому все изменения ссылочного объекта влияют на переданный объект. Если переменная примитивный тип, то передается только значение переменной, то есть копия этого значения. И изменения копии никак не влияет на начальное значение.

***8. Объясните, как в метод передать массив. И как массив вернуть из метода. Можно ли в методе изменить размер переданного массива.***

public void printArray(int[] array) {} printArray(new int[]{3, 1, 2, 6, 4, 2}); public static int[] reverse(int[] list) {}

Массивы - неизменяемые обекты. Если нужно вствлять/удалять объекты из чего либо – лучше использовать интерфейс Collection и его наследников.

***9. Поясните, что означает выражение «вернуть значение из метода». Как можно вернуть значение из метода. Есть ли разница при возврате значений примитивного и ссылочного типов.***

public static int[] reverse(int[] list) {return list;}

10. Перечислите известные вам алгоритмы сортировки значений, приведите код, реализующий эти алгоритмы.

Сортировка пузырьком

public static void bubbleSort(int[] array) {

    boolean sorted = false;

    int temp;

    while(!sorted) {

        sorted = true;

        for (int i = 0; i < array.length - 1; i++) {

            if (array[i] > array[i+1]) {

                temp = array[i];

                array[i] = array[i+1];

                array[i+1] = temp;

                sorted = false;

            }

        }

    }

}

Сортировка вставками

public static void insertionSort(int[] array) {

    for (int i = 1; i < array.length; i++) {

        int current = array[i];

        int j = i - 1;

        while(j >= 0 && current < array[j]) {

            array[j+1] = array[j];

            j--;

        }

         // в этой точке мы вышли, так что j так же -1

         // или в первом элементе, где текущий >= a[j]

        array[j+1] = current;

    }

}

Сортировка выбором

public static void selectionSort(int[] array) {

    for (int i = 0; i < array.length; i++) {

        int min = array[i];

        int minId = i;

        for (int j = i+1; j < array.length; j++) {

            if (array[j] < min) {

                min = array[j];

                minId = j;

            }

        }

        // замена

        int temp = array[i];

        array[i] = min;

        array[minId] = temp;

    }

}

Сортировка слиянием

Пирамидальная сортировка

Быстрая сортировка