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. Массив в Java представляет собой класс, при этом имя объекта класса массива является объектной ссылкой на динамическую память, в которой хранятся элементы массива. 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. Индексирование элементов начинается с нуля.

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

* Для объявления ссылки на массив можно записать пустые квадратные скобки после имени типа, например: int a[], int a[] []. Аналогичный результат получится при записи int []a. Существует два способа создания массива: с помощью оператора new или с помощью прямой инициализации присваиванием значений элементам массива в фигурных скобках. Значения элементов неинициализированного массива, для которого выделена память, устанавливаются в значения по умолчанию. Для численных типов (как в нашем примере) это будет 0, для boolean — false, для ссылочных типов — null. Массивы ссылочных значений, удовлетворяющих следующему условию: если тип переменной – массив на основе типа A, то значение типа массив на основе типа B допустимо тогда и только тогда, когда B приводимо к A

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

-

1. **Объясните, что представляет собой двумерных массив в Java, что такое “рваный массив”. Как узнать количество строк и количество элементов в каждой строке для “рваного” массива?**

- Массив массивов называется двумерным. Он похож на таблицу, у которой есть номер строки и номер столбца. Рваным называется массив массивов у которого разная длина строк в столбцах. Для нахождения количества строк использовать **length:**

int[][] matrix = new int[6][];  
matrix[0] = new int[10];  
matrix[1] = new int[50];  
System.*out*.println(matrix.length); // число строк = 6  
System.*out*.println(matrix[1].length); // число элементов во второй строке = 50

1. **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.

- Если попытаться записать в ячейку массива ссылку на объект неправильного типа, возникнет исключение ArrayStoreException.

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

- Разделение решаемой задачи на методы:

1) Уменьшает повторение кода.

2) Упрощает читаемость кода.

3) Улучшает в дальнейшем поддерживаемость кода.

4) Позволяет распараллелить задачу между разработчиками.

Метод содержит некоторую законченную последовательность действий (инструкций), направленных на решение отдельной задачи и его можно многократно использовать.

Т.е. если метод считает пироги и утюги одновременно, то использовать его для подсчета только утюгов или пирогов некорректно.

1. **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.

1. **Объясните, как в метод передать массив. И как массив вернуть из метода. Можно ли в методе изменить размер переданного массива.**
2. public static void main(String[] args){  
     
    int[] arr1 = {3, 1, 2, 5, 4};  
     
    *sum*(arr1); // передача массива arr1 в метод sum.  
     
    int[] arr = *m1*(); // точка вызова метода m1  
    for (int j : arr) System.*out*.print(j + " ");  
   }  
     
   public static void sum(int[] arrrr)  
   {  
    int sum = 0;  
    for (int j : arrrr) sum += j;  
    System.*out*.println(" Сумма : " + sum);  
     
   }  
     
   public static int[] m1()  
   {  
    return new int[] {1,2,3, 6}; // возвращаем массив из метода m1  
   }

Изменить размер переданного массива в методе нельзя ArrayIndexOutOfBoundsException (но не уверен)

1. **Поясните, что означает выражение ‘вернуть значение из метода’. Как можно вернуть значение из метода. Есть ли разница при возврате значений примитивного и ссылочного типов.**

- Вернуть значение из метода - получить данные (некий результат выполнения метода), которые приходят на его место после вызова. Возвращает значение из метода оператор **return.**

1. **Перечислите известные вам алгоритмы сортировки значений, приведите код, реализующий это алгоритмы.**

- Сортировка выбором, сортировка вставками, пузырьковая сортировка, сортировка Шелла. (код см. задачи)