**Массивы. Методы для работы с массивами.**

**Список вопросовтеоретической части.**

Свойство **массива**:22

-length

Методы класса **Array**для работы с массивами:

-sort() / sort(Comparator)

-toString() / deepToString()

-asList()

-binarySearch() / binarySearch(Comparator)

-equals() / deepEquals()

-hashCode() / deepHashCode()

-fill()

-copyOf() / copyOfRange()

Методкласса**System**дляработысмассивами:

-arrayCopy

**Список задач практической части:**

**Красным отмечены задачи, которые обязательны к решению**

**Все задачи оформлять в новом проекте Arrays в виде static-методов в классе ArrayUtil в пакете arrays, которые принимают массив в качестве аргументов(кроме вспомогательных методов), а так же другие необходимые аргументы для решения конкретной задачи. Методы называть так же, как называется задача.Вызовы методов выполнять в главном методе программы mainв классе Program в пакете program. Нельзя использовать стандартные методы для работы с массивами**

1. **fill**

Дан массив целых чисел. Заполнить данный массив с клавиатуры. Scannerподать как аргумент метода

2. **fillRandom**

Дан массив целых чисел. Заполнить данный массив числами в диапазоне от **aдо b**

**Указание:**

Для генерации случайного числа в диапазоне от a до b реализовать отдельный метод **getRandomNumber**

3. **toString**

Дан массив целых чисел. Преобразовать данный массив в строковое значение в формате **JSON**

**Пример:**

[10, 2, 6, 4]

4. **evenIndexes**

Дан массив целых чисел.Выведите все элементы массива с четными индексами. В программе запрещено использовать условную инструкцию для проверки четности индексов.

**Решение оформить 2 способами:**

-возвращать строковое представление массива в формате **JSON**

**-**используя ArrayList для накопления элементов, удовлетворяющих условию задачи

5. **evenData**

Дан массив целых чисел. Выведите все четные элементы массива.

**Указание:**

Для проверки четности элемента массива реализовать отдельный метод **isEven,** который будет проверять, является ли переданный ей элемент четным.

**Решение оформить 2 способами:**

-возвращать строковое представление массива в формате **JSON**

**-**используя ArrayList для накопления элементов, удовлетворяющих условию задачи

6. **countPositive**

Дан массив целых чисел. Определить количество положительных элементов в данном массиве.

7. **greatPrev**

Дан массив целых чисел. Выведите все элементы массива, которые больше предыдущего элемента.

**Решение оформить 2 способами:**

-возвращать строковое представление массива в формате **JSON**

**-**используя ArrayListдля накопления элементов, удовлетворяющих условию задачи

8. **sameNeighbours**

Дан массив целых чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет - не выводите ничего. Если таких пар соседей несколько - выведите первую пару.

**Указание**

-Вернуть результат в виде массива

9. **greaterNeighbours**

Дан массив целых чисел. Определите, сколько в этом массиве элементов, которые больше двух своих соседей и выведите количество таких элементов.

10. **max**

Дан массивцелых чисел. Выведите значение наибольшего элемента в массиве

11. **maxValues**

Дан массивцелых чисел. Выведите все индексы наибольшего значения данного массива

**Указание:**

-Вернуть результат в виде массива

-Вернуть результат в виде массива используя ArrayList

12. **minPositive**

Дан массив целых чисел. Выведите значение наименьшего из всех положительных элементов в массиве. Известно, что в массиве есть хотя бы один положительный элемент.

13. **minOdd**

Дан массив целых чисел. Выведите значение наименьшего нечетного элемента массива, а если в массиве нет нечетных элементов см. способы реализации.

**Указание:**

Для проверки нечетности элемента использовать заранее написанную функцию из **п.5**

**Решение оформить 2 способами:**

**-**Вернуть значение 0 при отсутствии элементов, удовлетворяющих условию задачи

**-**Вернуть значение **null,** указав тип возвращаемого значения объектную оболочку типа **int** – **Integer**.

14. **nearest**

Дан массив целых чисел и числоx. Найдите в данном массиве элемент, ближайший к x(элемент с минимальной абсолютной разницей с х).Если таких чисел несколько, выведите см. способы реализации.

**Решение оформить 3 способами:**

**-**Вернуть **последний** элемент, удовлетворяющий условиям задачи

**-**Вернуть **все** элементы, удовлетворяющие условиям задачи, используя массивы

**-**Вернуть **все** элементы, удовлетворяющие условиям задачи, используя **ArrayList**

**14.1 nearest**

Найдите в массиве два ближайших элемента (то есть два элемента с минимальной абсолютной разностью)

**14.2 testerClass**

Протестировать данный метод на случайно сгенерированном массиве длиной 1000 элементов, при верности теста выводить soutв формате: **---------test номер\_теста issuccessful**, при неверности выводить serrв формате: **---------testномер\_теста isinvalid.**

15. **indexInsertion**

Дана невозрастающая последовательность целых чисел в виде массива и число x. Вывести номер этой последовательности, куда можно было бы вставить данных элемент х, чтобы правило не возрастания сохранилось, учитывая, что нумерация элементов в последовательности начинается с 1. Если в последовательности уже есть элементы, равные х, то считается, что данный элемент вставляется после них.

**Указание:**

Решение оформить в виде алгоритма линейного поиска циклом **while**

16. **countDifferent**

Дан массив целых чисел, упорядоченный по неубыванию элементов в нем. Определите, сколько в нем различных элементов (количество неравных друг другу элементов).

17.**reverse**

Дан массивцелых чисел. Переставьте элементы данного массива в обратном порядке. Метод должен производить разворот исходного массива, ничего не возвращая.

18. **reverseNeighbours**

Дан массивцелых чисел.Переставьте соседние элементы массива. Если элементов нечетное число, то последний элемент остается на своем месте.

19. **delete**

Дан массив целых чисел и номер элемента в массиве k. Удалите из массива элемент с индексом k.

**Решение оформить 2 способами:**

-использовать вспомогательный массив для хранения элементов

**-**использовать ArrayListдля выполнения операции удаления

20. **insert**

Дан массивцелых чисел, число k и значение C. Необходимо вставить в массив на позицию с индексом k элемент, равный C

**Решение оформить 2 способами:**

-использовать вспомогательный массив для хранения элементов

**-**использовать ArrayListдля выполнения операции вставки

21. **shift**

Дан массив целых чисел и параметр k. Если данный параметр истинный, то выполнить циклический сдвиг массива на одну позицию вправо, если ложный, то влево. Метод должен производить операции над массивом, ничего не возвращая.

22. **shift**

Дан массив целых чисел и числоk. Выполнить циклический сдвиг массива на |k| позиций вправо, если k>0 или влево, если k<0. Метод должен производить операции над массивом, ничего не возвращая.

23. **countEqualPairs**

Дан массивцелых чисел. Посчитайте, сколько в нем пар элементов, равных друг другу. Считается, что любые два элемента, равные друг другу образуют одну пару, которую необходимо посчитать.

24. **countDifferent**

Дан массивцелых чисел. Посчитайте, сколько в нем различных элементов, не изменяя самого массива.

**Указание**:

Необходимо считать те элементы, которые встретились нам впервые. Чтобы проверить, встретился ли нам элемент A[i] впервые, необходимо проверить, встречается ли значение A[i] среди элементов с индексами, меньшими i.

25. **median**

Дан массив целых чисел. Найти ***медиану*** массива: элемент, который стоял бы ровно посередине массива, если массив упорядочить.

При решении этой задачи нельзя модифицировать данный массив (в том числе и сортировать его), использовать вспомогательные массивы.

**Указание:**

**-**Если размер массива является четным, либо в массиве не все элементы являются различными, то вывести значение **null.**

**-**Для определения, все ли элементы массива являются различными использовать отдельный метод.

26.**unique**

Дан массивцелых чисел. Выведите те его элементы, которые встречаются в массиве только один раз. Элементы нужно выводить в том порядке, в котором они встречаются в массиве.

**Решение оформить 2 способами:**

-использовать вспомогательный массив для хранения элементов

**-**использовать ArrayListдля накопления элементов, удовлетворяющих условию задачи

27. **frequent**

Дан массив целых чисел. Не изменяя массива определить, какое число в этом массиве встречается чаще всего.

Если таких чисел несколько, выведите см. способы реализации.

**Решение оформить 3 способами:**

**-**Вернуть **первый** элемент, удовлетворяющий условиям задачи

**-**Вернуть **все** элементы, удовлетворяющие условиям задачи, используя дополнительный массив

**-**Вернуть **все** элементы, удовлетворяющие условиям задачи, используя **ArrayList**

28. **kBonachi**

Назовем последовательность чисел последовательностью k-боначчи, если каждый элемент этой последовательности является суммой k предыдущих членов последовательности. В частности, последовательность 2-боначчи является последовательностью Фибоначчи.

Более формально, i−й элемент последовательности ki равен 1, если 0≤i≤k−1 и равен сумме k предыдущих членов последовательности ki−1+ki−2+...+ki−k при i≥k.

Даны два числа k и n (k≥2, n≥0). Вычислите n-й член последовательности k-боначчи kn.

**Решение оформить 2 способами:**

-использовать массив для хранения элементов

**-**использовать ArrayListдля накопления элементов

29. **search**

Дан массив целых чисел и число key. Методом линейного поиска при помощи цикла for найти индекс вхождения числа **key** в массиве

**29.1 search**

Модернизировать программу таким образом, чтобы она вернула индексы всех вхождений числа **key**в массиве.

**Решение оформить 2 способами:**

-использовать массив для хранения элементов

**-**использовать ArrayListдля накопления элементов

30. **equals**

Даны два массива целых чисел. Сравнить их на равенство.

**Указание:**

Под равенством двух массивов понимать их тождественное равенство, то есть два массива с одинаковыми длинами, одинаковыми элементами, но разным порядком этих элементов считаются разными

31. **equalsIgnoreCase**

Даны два массива целых чисел. Сравнить их на равенство по содержанию, то есть не учитывая порядок этих элементов.

32. **copyOf**

Дан массив, сделать его копию

**Указание:**

Метод должен принимать на вход 2 параметра: исходный массив, который надо скопировать и длину нового массива. Если длина исходного массива больше длины нового массива, то копируется только часть исходного массива, если длина исходного массива меньше длины нового массива, то в новый массив исходный массив копируется полностью, а оставшаяся незаполненная часть записывается нулями, если же размер нового массива совпадает с размером исходного, то в новый массив копируется полностью исходный.

32. **copyOfRange**

Дан массив и два целых числа b и e, сделать копию исходного массива начиная с индекса а и заканчивая b

**Указание:**

Метод должен принимать на вход 3 параметра: исходный массив, который надо скопировать и границы копирования. Если длина исходного массива меньше числа a, то вернуть **null**.Если длина исходного массива меньше числа b, то в новый массив исходный массив копируется полностью, а оставшаяся незаполненная часть записывается нулями.

32. **Sort**

Дан массив целых чисел. Выполнить сортировку массива пузырьком и сортировкой отбором(линейная сортировка).

**Решение оформить 2 способами:**

**-**Выполнить данные сортировки

-Выполнить данные сортировки, принимая в качестве аргумента **Comparator** для сравнения элементов

32.1 **isOrder**

Дан массив целых чисел. Определить, является ли он упорядоченным

32.2 **TesterClass**

В класс **Tester** в пакете **tests**, определить в нем метод, который тестирует метод Протестировать методы **sort**на случайно сгенерированных 1000 массивах, применяя метод **isOrder**для проверки, является ли массив упорядоченным после выполнения сортировки, при верности теста выводить soutв формате: **---------test номер\_теста issuccessful**, при неверности выводить serrв формате: **---------testномер\_теста isinvalid.**

33. **fillPrimeRandom**

Дан массив. Заполнить исходный массив целыми случайными простыми числами

**Указание:**

Для проверки простоты числа написать отдельный метод **isPrime**

34. **getPermutations**

Дан массив целых чисел. Сгенерировать все возможные перестановки данного массива

**Указание:**

-Число перестановок длины nравно **n!**

**-**Вернуть все перестановки массива в виде таблицы(двумерного массива)

35. **averagePoint**

Вводится следующая система вычисления среднего балла работоспособности разработчиков в компании «Сбербанк-Технологии»: вычисляется среднее арифметическое всех баллов, поставленныхтимлидом за определенный период времени, и ставится ближайшая целая оценка, не превосходящая среднего арифметического.

При этом если у разработчика есть «двойка», а следующий за ней балл – выше «двойки», то двойка считается отработанной, и при вычислении среднего арифметического не учитывается.

Вводится **N**натуральных чисел в диапазоне от 2 до 5 через пробел – баллы разработчика за определённый период времени.

Вычислить натуральное число (от 2 до 5) – его среднийбалл за текущий период.

36. **countPassengersTime**

Для изучения пассажиропотока в метро было записано время входа и время выхода в метро каждого пассажира. На основании этих данных определите, сколько пассажиров было в метро в некоторый заданный момент времени T.

Программа получает на вход **двумерный массив** данных**P** из N строк – числа пассажировза текущий деньи двух столбцов, а так же время **T**. В первом столбце массива определяется время входа P(i, 1) и во втором – времявыхода P(i, 2) каждого пассажира,где P(i,1)≤P(i,2). Время задается в минутах от начала работы метрополитена.

Вычислите одно значение: количество пассажиров в момент времени **T**. Если какой-то пассажир в момент T входит или выходит, то его тоже необходимо посчитать.

37. **towards**

Дан массив вещественных чисел. Сформировать его строковое представление в формате**JSON**следующем порядке: первое число, последнее, второе, предпоследнее и так далее все числа (см. пример).

**Пример:**

Входные данные:1 2 3 4 5

Выходные данные:[1,5,2, 4,3]

38. **rushHour**

Дан массив целых чисел – количество пассажиров в метро за каждый час его работы и число k – продолжительность часа пик. Определить K подряд идущих часов работы метрополитена с максимальным суммарным числом пассажиров и вычислить суммарное число пассажиров за эти часы.

39. **equalItems**

Дан массив. Найдите элементы, равные друг другу.

**Решение оформить 2 способами:**

-использовать массив для хранения элементов(для определения количества равных использовать метод из задачи 23)

**-**использовать ArrayListдля накопления элементов

40. **group**

Дан массив строковых данных. Сформировать на основании данного массива новый массив, элементами которого будут новые строки, скрепленные между собой по принципу их равности. То есть необходимо реализовать алгоритм группировки строк по их равенству.

**Пример:**

**Входные данные:** [“aa”, “b”, “a”, “bb”, “aa”, “bb”]

**Выходные данные:** [“aaaa”, “b”, “a”, “bbbb”]

**Указание:**

Для сравнения строк использовать механизм str1.equals(str2). Оператор == для строк использовать нельзя

**Решение оформить 2 способами:**

-использовать массив для хранения элементов

**-**использовать ArrayListдля накопления элементов

41. **setDiam**

На плоскости даны N точек, заданных своими координатами. Найти две наиболее удаленные точки и вычислить расстояние между ними.

Программа получает на вход двумерный массив**T**целых чисел из Nстрок – количества точек. В первом столбце T(i, 1) записана координата по оси х, во втором столбцеT(i,2) –координата y.

Вычислить одно действительное число — наибольшее расстояние между двумя из данных точек.

42. **debts**

В одном карточном клубе состоит N джентльменов. Иногда азарт некоторых из них берет верх над благоразумием, и кто-то проигрывает больше денег, чем у него есть с собой. В этом случае проигравший обычно берет в долг у кого-то из посетителей клуба, чтобы расплатиться с партнерами по игре. Чтобы начать новый год “с чистого листа”, джентльмены решили собраться в клубе и оплатить все долговые расписки, которые накопились у них друг к другу. Однако выяснилось, что иногда одни и те же джентльмены в разные дни выступали как в роли должников, так и в роли кредиторов. Поскольку истинные джентльмены считают мелочный подсчет денег ниже своего достоинства, то расчетами придется заняться уполномоченному программисту.

Написать программу, которая по заданным распискам вычислит, сколько всего должен каждый джентльмен выплатить другим (или получить с других).

На вход методу подается двумерный массив **D** состоящий из K строк — количество долговых расписок и 3 столбцов, число **N**— количество джентльменов. **D**(i, 1) - номер джентльмена взявшего в долг;**D**(i, 2) - номер джентльмена давшего деньги; **D**(i, 3)- сумма.

Вычислить массив **balance** из N чисел — суммы, которые должны получить соответствующие джентльмены. **balance**(i) положительное число, если этот джентльмен должен получить деньги от других, отрицательное — если он должен отдать деньги другим.

43. **pascalTriangle**

По данному числу N сформировать первые N+1 строку треугольника Паскаля.

44. **temperatureMinimum**

Метеорологи ведут многолетние наблюдения за тем, в каком году была минимальная температура в данный день года. Например, абсолютный минимум температуры в Москве 8 марта был -32 градуса (1890).

В течение **k** лет метеорологи вели наблюдения за **n** днями года. Для каждого из этих **n** дней укажите минимальную температуру, которая была в этот день за **k** лет наблюдений.

Данные подаются в виде двумерного массива из **k**строк и **n**столбцов. Далее идет k строк, i-я строка содержит n чисел: значения температур для n дней наблюдений i-го года.

Программа должна сформировать массив из **n** чисел: минимальное значение температуры для каждого из дней наблюдений.

45. **minimumPrecipitation**

Даны результаты метеорологических наблюдений: количество осадков в каждый из 31 дня марта. Метеорологи хотят определить, какая из недель марта была наименее дождливой. Неделя — это семь дней с понедельника до воскресенья, то есть в марте может быть три или четыре полные недели.

Программа получает на вход массив из 31 целого неотрицательное число через пробел: количество осадков для каждого из дней и число**k** от 1 до 7: день недели, на который приходится 1 марта (1 означает понедельник, 2  вторник и так далее).

Программа должна определить неделю с наименьшим суммарным числом осадков и вывести суммарное число остатков на этой неделе.