

23c

Реализовать класс Graph с методами:

- `ArrayList<Edge> getEdgesList()` – возвращает представление графа в виде списка ребер
- `int [][] getAdjecencyMatrix()` – возвращает матрицу смежности
- `int [][] getIncidenceMatrix()` – возвращает матрицу инцидентности
- `ArrayList<SuperNode> getAdjecencyLists()` – возвращает списки (структуру) смежности

Классы Edge и SuperNode, как и атрибуты самого Graph продумайте самостоятельно.

23b

Реализовать в классе Graph статические методы, позволяющие сконvertировать одно из четырех указанных представлений графа в другое (т.е. всего 12 методов)

24c

Найти с помощью алгоритма перебора с возвратами все строки длины n из строчных английских букв, чтобы в строке было не более трех гласных.

24b

Найти с помощью алгоритма перебора с возвратами все целые положительные числа с m цифрами, у которых сумма цифр меньше числа k.

25b

Найти раскраску графа (вводится список ребер, создается объект класса Graph) с помощью алгоритма перебора с возвратами.

26c

Скопировать один текстовый файл в другой посимвольно с помощью FileReader и FileWriter, а потом построчно с помощью BufferedReader и PrintWriter. Сравнить с помощью nanoTime() время работы обоих способов.

26b

Напишите программу, которая создает около 10 объектов класса Graph и сериализует их в каком-нибудь файле. А затем напишите другую программу, которая осуществляет десериализацию, проверяя при этом, четно ли количество вершин или нечетно. Выведите количество графов с четным количеством вершин на экран.

27b

Реализуйте поток, который выводит каждую секунду текущее время в определенном часовом поясе (читаем про `java.util.Calendar` и UTC). Затем реализуйте программу, которая получает на вход строки с наименованием города и его часовым поясом, затем для каждого города создает свой поток.

Вход:

Samara

+4

Buenos Aires

-3

London

0

Выход:

Samara: 14:15:10

Buenos Aires: 7:15:10

London: 10:15:10

Samara: 14:15:11

Buenos Aires: 7:15:11

London: 10:15:11

и т.д.