

Задача 10. Массивы / списки объектов (классов)

В задаче обязательно использовать свои (описанные в вашей программе) составные типы данных (структуры или классы).

Для представления набора объектов лучше использовать списки (массивы также допустимы, но менее удобны для данной задачи).

Данная задача должна быть оформлена двумя способами:

1) в виде консольного приложения;

2) в виде оконного приложения, где массив объектов можно задать в JTable. При этом должна быть возможность загрузить данные из файла в JTable (реализованная в виде двух функций: чтение данных из файла в двумерный массив и отображение двумерного массива в JTable), а также сохранить данные из JTable в файл (реализованная в виде двух функций: чтение данных из JTable в двумерный массив и запись двумерного массива в файл).

Функции, реализующие логику задачи и чтение / запись данных из файлов / в файлы, должны быть оформлены в виде отдельного модуля (в отдельном файле). Этот модуль без каких-либо изменений должен использоваться в двух программах: с консольным интерфейсом (файлы для чтения / записи задаются в параметрах командной строки) и оконным интерфейсом.

Данные в файлах должны храниться в следующем виде: каждая отдельная строка описывает отдельный объект. Соответственно строки состоят из нескольких элементов (чисел, строк), каждый из которых описывает отдельный атрибут объекта. Например, набор линий вида $ax + by + c = 0$ будет храниться в текстовом файле в следующем виде:

```
4 5 4
10 2 12
5 34 23
и т.д.
```

, где каждая строка описывает отдельную линию с параметрами a , b , c .

Заранее придумать не менее 5 различных тестов, охватывающих как типичные, так и все возможные граничные (наиболее невероятные и показательные) ситуации. (Сохранить в текстовых файлах input01.txt, input02.txt и т. д.)

Решение, естественно, должно быть оформлено в виде отдельной функции. В реализации обязательно использовать вспомогательные функции примерно так же, как в предыдущих Задачах № 8 - 9 (в этой задаче, какие именно вспомогательные функции использовать, не расписано, вы должны самостоятельно решить, какие именно будут удобны для вашего варианта).

Варианты:

1. Для набора линий вида $ax + by + c = 0$ найти наибольшее кол-во параллельных линий (совпадающие линии считаются параллельными). Вернуть наибольшее множество параллельных линий в виде списка / массива индексов линий из переданного набора. В случае существования нескольких различных наборов параллельных линий из максимального кол-ва элементов – выбрать произвольный.
2. Для набора линий вида $ax + by + c = 0$ найти ближайшую к началу координат точку пересечения двух произвольных линий из набора. Также указать, какие линии из набора при пересечении дают такую точку. В случае существования нескольких ближайших к началу координат точек пересечения – выбрать произвольную.

3. Отрезок на прямой линии задается координатой начала A и координатой окончания B (точки начала и окончания отрезка принадлежат отрезку). Задан набор отрезков, которые могут накладываться друг на друга. Необходимо выполнить объединение этих отрезков, представив результат также в виде набора отрезков.
Подсказка: перед объединением необходимо отсортировать отрезки по координате начала отрезка.
4. Входные данные соответствуют предыдущей задаче, только теперь вам необходимо среди заданного набора отрезков найти два отрезка, сумма длин объединения которых будет максимальной (здесь возможны два варианта, если эти два отрезка не пересекались, то объединение будет состоять из этих же двух отрезков, если же отрезки пересекались, то их объединением будет один отрезок).
5. Коробки описываются длиной, шириной и высотой. Задан набор коробок. Найти всевозможные пары коробок, где первая коробка может быть вложена во вторую коробку (при вложении коробки разрешается вращать).
6. Коробки описываются длиной, шириной и высотой. Задан набор коробок. Упорядочить коробки по объему, для одинакового объема – по самой длинной стороне, для одинаковой самой длинной стороне – по самой короткой стороне в обратном порядке. Обязательно реализовать функцию сравнения двух коробок для нужного порядка сортировки.
7. Отфильтровать набор линий вида $ax + by + c = 0$ следующим образом: из каждого параллельного подмножества линий данного набора оставить одну единственную линию – самую верхнюю (а в случае линий, параллельных OY , – самую правую).
8. Для набора прямоугольников, стороны которых параллельны OX и OY , заданных координатами 2-х диагональных вершин, найти прямоугольник, внутри которого расположено максимальное кол-во других прямоугольников (граница вложенного прямоугольника может проходить по границе внешнего прямоугольника). В случае существования нескольких подходящих прямоугольников – выбрать максимальной площади (если и таких будет несколько – то произвольный).
9. Для набора прямоугольников, стороны которых параллельны OX и OY , заданных координатами 2-х диагональных вершин, найти прямоугольник, который лежит внутри максимального кол-ва других прямоугольников (граница вложенного прямоугольника может проходить по границе внешнего прямоугольника). В случае существования нескольких подходящих прямоугольников – выбрать минимального периметра (если и таких будет несколько – то произвольный).
10. Для набора прямоугольников, стороны которых параллельны OX и OY , заданных координатами 2-х диагональных вершин, найти все прямоугольники, которые не перекрываются никакими другими прямоугольниками (т.е. если вырезать прямоугольники нужного размера и раскладывать по координатам на листе бумаги, то нужные прямоугольники не будут накладываться на другие прямоугольники, но могут касаться сторонами).
11. Задача 8, только применительно к набору окружностей (кругов).
12. Задача 9, только применительно к набору окружностей (кругов).

13. Задача 10, только применительно к набору окружностей (кругов).
14. Для набора точек на плоскости найти такие три точки, для которых площадь треугольника с вершинами в данных точках будет максимальной. В случае существования нескольких подходящих троек точек – выбрать произвольную.
15. Для набора точек на плоскости найти такие три точки, для которых периметр треугольника с вершинами в данных точках будет минимальным. В случае существования нескольких подходящих троек точек – выбрать произвольную.
16. Упорядочить набор треугольников, заданных тремя вершинами, по площади (от меньшей к большей).
17. Для набора треугольников, заданных тремя вершинами, найти такие, которые одновременно лежат во всех 4-х четвертях (на координатной сетке).
Треугольник описывает как три точки (отдельный тип – структура или класс), каждая точка – как две координаты (отдельный тип – структура или класс).
18. Для набора треугольников, заданных тремя вершинами, найти такие, которые полностью лежат строго в одной четверти (на координатной сетке) – неважно в какой из 4-х, главное, что в одной.
Треугольник описывает как три точки (отдельный тип – структура или класс), каждая точка – как две координаты (отдельный тип – структура или класс).
19. (*) Разбить набор треугольников, заданных тремя вершинами, на подмножества подобных треугольников (если для какого-то треугольника не будет найдено подобных треугольников, то этот треугольник будет образовывать подмножество из одного треугольника).
Полученный ответ представить в виде структуры данных `List<List<Triangle>>` (или `Triangle[][]`, но это менее удобно).
20. Отсортировать заданный в произвольном порядке набор студентов (каждый студент описывается в виде: ФИО, номер курса, номер группы) сначала по номеру курса, затем по номеру группы, затем по ФИО.
21. При поступлении на некоторую специальность подали заявление множество абитуриентов (абитуриент описывается следующим образом: ФИО, оценка по русскому языку, оценка по математике, оценка по физике). На соответствующей специальности N бюджетных мест, отберите тех студентов, которые поступят. Приоритет имеют студенты с максимальной суммой баллов, затем с большим баллом по математике, затем по физике.
22. Входные данные соответствуют предыдущей задаче, однако дополнительно появляется информация, принес ли абитуриент оригинал аттестата. В первую волну поступят студенты среди первых N по приоритету, которые принесли оригинал аттестата. Отберите студентов, которые поступят в первую волну приема.
23. Входные данные соответствуют предыдущей задаче, только теперь вам необходимо среди заданного набора отрезков найти три непересекающихся отрезка, сумма длин которых будет максимальной.

24. (*) В магазине продаются различные конфеты. Конфеты описываются названием и ценой за килограмм. У вас есть N денег. Вы хотите купить максимальное количество различных конфет на указанную сумму (по килограмму каждого вида). Однако если можно купить различные конфеты, вы предпочитаете более дорогие, полагая, что они более хорошие. Какие конфеты вы выберете, сколько денег у вас останется?
- Примечание: предположим, вы выяснили, что максимально можете купить K кг конфет, тогда первый кг вы выберете самых дорогих конфет при условии, что вам хватит денег на $K-1$ кг каких-то других конфет (также вы действуете для 2-го вида конфет и т.д.).
25. Прайс-лист содержит следующую информацию о планшетах: название модели, объем памяти, рейтинг модели среди пользователей (цифра от 1 до 5), стоимость. У вас есть N денег, вам необходимо купить самый хороший планшет, на который вам хватит денег. Для вас приоритетным является объем памяти, в случае одинакового объема – рейтинг модели. Если подходящих моделей будет несколько, вы выберете Samsung или Asus (присутствует в названии модели), а затем все остальные. Если по этим критериям вам подходят несколько моделей – выбирайте любой.
26. Входные данные соответствуют предыдущей задаче, однако теперь вам надо купить K самых дешевых планшетов с объемом памяти не ниже M и рейтингом не ниже R (призы победителям олимпиады). Какие планшет вы выберете и сколько денег вы потратите?
27. Задан набор студентов в виде (ФИО, пол, номер курса, средний балл). Необходимо на новогодний бал отправить лучших студентов – по одной паре (парень и девушка) с каждого курса с лучшей успеваемостью. Если есть несколько студентов с максимальной успеваемостью – выбрать случайным образом (именно случайным, т.е. при следующем выборе вполне возможно отобраны будут другие студенты). Если какой-то курс полностью состоит из представителей одного пола, то с такого курса на балл никого не отправлять.
28. Входные данные соответствуют предыдущей задаче, однако теперь вам необходимо распределить N ($N \geq$ кол-ву курсов) повышенных стипендий следующим образом: необходимо с каждого курса дать стипендию лучшему по успеваемости студенту, а остальные стипендии раздать лучшим оставшимся студентам, не обращая внимания на курс.
29. (*) Входные данные соответствуют предыдущей задаче, но теперь студентов будем отчислять. Необходимо выбрать студентов для отчисления следующим образом: всех, у кого средний балл ниже X . Однако после отчисления на каждом курсе должно остаться не менее N студентов. Если по критерию отчисления «средний балл $< X$ » для какого-то курса будет отчислено слишком много студентов, то отчислить самых слабых, оставив на данном курсе ровно N студентов. Если при этом окажется так, что несколько студентов с одинаковыми баллами окажется на границе отчисления, то не отчислять никого (например, надо до N отчислить 3 человек, а у нас 5 совершенно одинаковых двоечников, то все эти 5 двоечников продолжают учиться, а на курсе, соответственно, останется $N+2$ студента).
30. В базе данных недвижимости хранится информация о квартирах – район города (строка), количество комнат, общая площадь, площадь кухни, стоимость. Для каждого района города найти самую дешевую квартиру с указанным кол-вом комнат N (задается) и площадью не меньше S (также задается).
31. Входные данные соответствуют предыдущей задаче, однако теперь вам необходимо

написать программу, которая будет фильтровать объявления. В качестве параметров можно задать минимальное и максимальное значение для каждой характеристики квартиры (количество комнат, общая площадь, площадь кухни, стоимость). Учесть, что некоторые параметры фильтра могут быть опущены.

32. Входные данные соответствуют предыдущей задаче, однако теперь вам необходимо для каждого района и количества комнат рассчитать среднюю стоимость квадратного метра.

33.