

1. Рюкзак

Задание выполняется в “настоящей” консоли (тренажер не подойдет). Альтернативно, для любознательных*, можно сделать Android-приложение с кнопками и подходящим UI.

1.1 Создать класс Item (вещь). У Item есть поля:

- Название
- Вес
- Объем
- Тип вещи (еда, бумаги, одежда, инструмент). Int или enum.

1.2 Создать класс рюкзака. В классе есть поля:

- ArrayList<Item> items - содержимое рюкзака
- Предельная вместимость по объему (без учета навесных карманов) - выбрать по вкусу
- Предельная грузоподъемность (без учета навесных карманов) - выбрать по вкусу
- Количество навешенных в данный момент карманов. На рюкзак можно дополнительно прикрепить навесные карманы, каждый карман увеличивает грузоподъемность и объем рюкзака на некоторую величину (выбрать по вкусу).

Также в рюкзаке есть методы для добавления/удаления вещей/навесных карманов в инвентарь.

Существуют ограничения по загрузке рюкзака. Нельзя положить туда больше, чем позволяет объем/грузоподъемность, с учетом карманов. Кроме того, нельзя снимать карманы, не вынув достаточное количество вещей. Также, в рюкзак нельзя класть одномоментно более 2-х типов различных вещей. Необходимо убедиться, что при погрузке/разгрузке рюкзака ограничения всегда соблюдаются.

1.3 Сделать метод, вычисляющий статистику загрузки рюкзака:

- Количество вещей в рюкзаке
- Свободный объем (в единицах объема)
- Свободную грузоподъемность (в единицах массы)
- Процент загрузки по объему: а) общий б) с распределением по типам вещей (например, всего занято 70% рюкзака, при этом 30% занимает еда и 40% занимает одежда)
- Процент общей загрузки по массе (аналогично)

1.4 Подготовить “магазин” вещей. Наделать штук 10-20 вещей с разными параметрами массы/объема. Предусмотреть как легкие, но объемные вещи, так и маленькие, но тяжелые.

В методе main сделать интерактивное “меню” с пунктами (пример прилагается):

- [показ содержимого рюкзака]
- [показ статистики загрузки]
- 1. погрузить в рюкзак вещь из магазина
- 2. убрать вещь из рюкзака
- 3. навесить дополнительный карман
- 4. убрать дополнительный карман
- 0. выход из программы

1.5 Настроить классы рюкзака и вещей таким образом (т.е. подобрать методы и области видимости), чтобы его содержимое можно было свободно прочитать из метода `main`, но при этом **никакой код из метода `main` (даже написанный по ошибке)** не мог бы изменить содержимое рюкзака вопреки правилам.

* 1.6 Подумать, как изменится код рюкзака в случаях, если рюкзак большой (миллионы вещей), и при этом:

- Программист часто смотрит в рюкзак, но редко кладет/вынимает туда/оттуда вещи
- Программист редко смотрит в рюкзак, но часто кладет/вынимает туда/оттуда вещи

Реализовать разные варианты

2. Сортировка пузырьком.

Выполняется на тренажере. Ввести N точек. Отсортировать точки по удалению от точки $0,0$, вывести текстом отсортированный список. Алгоритм сортировки создать самостоятельно, запрещается использовать “стандартные” библиотеки. Использовать сортировку пузырьком, принцип сортировки спросить или посмотреть в гугле. На холсте отрисовать точки, пометить каждую точку номером в соответствии с сортировкой.

3. Сортировка относительная

То же, что и 2, но сортировать нужно не относительно точки $0,0$, а относительно произвольной точки, введенной пользователем. При отрисовке на холсте отрисовать и эту точку тоже, другим цветом.