Тестовое приложение по заполнению стакана жидкостью.

Все данные и операции выполняются через консоль/терминал.

На старте приложения пользователю даются выбор стакана:

1. Выбор типа стакана (Параллелепипед, Цилиндр).
2. Выбор размера стакана. Параметры одинаковые для всех видов стаканов - ширина и высота (поэтому считаем, что у стакана типа “Параллелепипед” основание квадратное). Единицы измерения размеров стакана - см.

Так же в системе есть список возможных жидкостей. У каждой есть своя плотность (кг/м3):

* + Вода - 1000
  + Бензин - 700
  + Масло - 900
  + Сливки - 940

Следовательно, жидкость с меньшей плотностью будет находиться вверху стакана, неважно, когда она была налита.

После этого появляется выбор:

1. Добавить жидкость
   1. Выбрать тип жидкости
   2. Выбрать объем (см3)
   3. Если указанный объем превышает свободное место в стакане, пользователю выводится сообщение, но жидкость добавляется в стакан, насколько это возможно (обрати внимание на плотность жидкости).
   4. Одна и та же жидкость может быть добавлена несколько раз
2. Вылить жидкость
   1. Выбрать объем (см3)
3. Показать текущую информацию о содержимом стакана
   1. Какие жидкости какого объема находятся в стакане (отсортированные от большей к меньшей плотности)
   2. Какая жидкость занимает больше всего объема
   3. Оставшееся свободное место в стакане
4. Изменить стакан
   1. Шаги идентичны тем, что предоставлялись на старте приложения
   2. Жидкость должна быть “перелита” в новый стакан
5. Сохранить текущий прогресс в файл/Загрузить прогресс из файла
   1. Формат данных - json
   2. Для сериализации/десериализации использовать библиотеку jackson
   3. Использовать методы ObjectMapper.readValue, ObjectMapper.writeValue
6. Тесты
   1. Unit-тесты (JUnit5)
   2. Желательно - не менее 5-ти классов, которые предназначены для работы со стаканами и жидкостями
7. Многопоточность\* (бонусная задача)
   1. Реализовать возможность выбора нескольких жидкостей при добавлении в стакан
   2. Объем указывается один на все выбранные жидкости

Примечания:

1. Для реализации не используем сторонние библиотеки (если обратное не указано выше).
2. Постараться применить пройденный материал, где возможно.
3. main() метод должен состоять из “одной строчки”, которая вызывает экземпляр класс для входа в интерактивный режим работы с консолью
4. Приложение должно учитывать некорректно введенные данные пользователем.
5. Использовать maven, реализованный проект залить в github.