Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №5

По дисциплине

«Программирование»

Вариант 41514

Выполнил: студент гр. P3115

Лазеев С.М.

Проверил:

Горбунов М.В.

Санкт-Петербург 2021

Задание:

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса HumanBeing, описание которого приведено в данном мне варианте работы.

Разработанная программа должна удовлетворять следующим требованиям:

Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.

Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.

Для хранения необходимо использовать коллекцию типа java.util.PriorityQueue

При запуске приложения коллекция должна автоматически заполняться значениями из файла.

Имя файла должно передаваться программе с помощью: переменная окружения.

Данные должны храниться в файле в формате csv

Чтение данных из файла необходимо реализовать с помощью класса java.util.Scanner

Запись данных в файл необходимо реализовать с помощью класса java.io.BufferedWriter

Все классы в программе должны быть задокументированы в формате javadoc.

Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутсвие прав доступа к файлу и т.п.).

В интерактивном режиме программа должна поддерживать выполнение следующих команд:

help : вывести справку по доступным командам

info : вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)

show : вывести в стандартный поток вывода все элементы коллекции в строковом представлении

add {element} : добавить новый элемент в коллекцию

update id {element} : обновить значение элемента коллекции, id которого равен заданному

remove_by_id id : удалить элемент из коллекции по его id

clear : очистить коллекцию

save : сохранить коллекцию в файл

execute_script file_name : считать и исполнить скрипт из указанного файла. В скрипте содержатся команды в таком же виде, в котором их вводит пользователь в интерактивном режиме.

exit : завершить программу (без сохранения в файл)

head : вывести первый элемент коллекции

add_if_min {element} : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции

history : вывести последние 10 команд (без их аргументов)

remove_any_by_mood mood : удалить из коллекции один элемент, значение поля mood которого эквивалентно заданному

filter_less_than_car car : вывести элементы, значение поля car которых меньше заданного

print_descending : вывести элементы коллекции в порядке убывания

Формат ввода команд:

Все аргументы команды, являющиеся стандартными типами данных (примитивные типы, классы-оболочки, String, классы для хранения дат), должны вводиться в той же строке, что и имя команды.

Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.

При вводе составных типов данных пользователю должно показываться приглашение к вводу, содержащее имя поля (например, "Введите дату рождения:")

Если поле является enum'ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).

При некорректном пользовательском вводе (введена строка, не являющаяся именем константы в enum'e; введена строка вместо числа; введенное число не входит в указанные границы и т.п.) должно быть показано сообщение об ошибке и предложено повторить ввод поля.

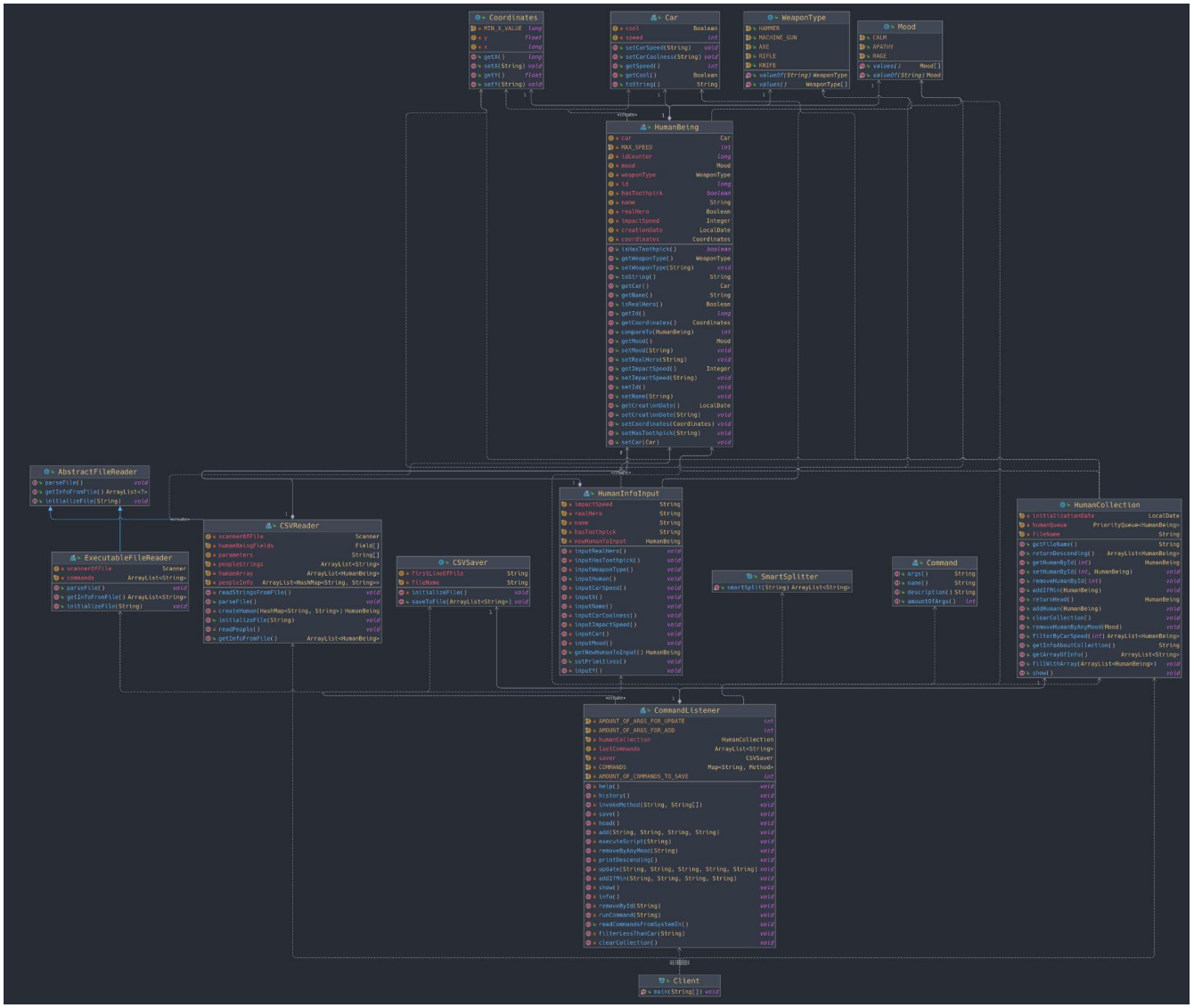
Для ввода значений null использовать пустую строку.

Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.

Описание хранимых в коллекции классов:

```
public class HumanBeing {
    private long id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.time.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Boolean realHero; //Поле не может быть null
    private boolean hasToothpick;
    private Integer impactSpeed; //Максимальное значение поля: 712, Поле может быть null
    private WeaponType weaponType; //Поле может быть null
    private Mood mood; //Поле может быть null
    private Car car; //Поле может быть null
}
public class Coordinates {
    private long x; //Значение поля должно быть больше -759
    private float y;
}
public class Car {
    private Boolean cool; //Поле может быть null
}
public enum WeaponType {
    HAMMER,
    AXE,
    RIFLE,
    KNIFE,
    MACHINE_GUN;
}
public enum Mood {
    APATHY,
    CALM,
    RAGE;
}
```

Диаграмма классов реализованной объектной модели:



Ссылка на репозиторий GitHub: https://github.com/k1b24/LaboratoryWork5

Выводы по работе: *Выполняя данную работу, я реализовал консольное приложение, позволяющее работать с коллекцией экземпляров класса в интерактивном режиме. Я узнал, как работать с файлами и потоками ввода вывода в Java, а также реализовывать консольные команды для работы с приложением в интерактивном режиме. Кроме того, я на практике применил свои знания о параметризованных типах и закрепил знания о принципах ООП и SOLID, полученные мною в первом семестре.*