Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники

Вариант №370331
Лабораторная работа №5
по дисциплине
Программирование

Выполнил Студент группы P3112
Кобелев Роман Павлович
Преподаватель: **Гаврилов Антон Валерьевич.**

г. Санкт-Петербург 2023г.

1 Задание

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

Разработанная программа должна удовлетворять следующим требованиям:

1. Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
2. Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.
3. Для хранения необходимо использовать коллекцию типа java.util.LinkedList
4. При запуске приложения коллекция должна автоматически заполняться значениями из файла.
5. Имя файла должно передаваться программе с помощью: аргумент командной строки.
6. Данные должны храниться в файле в формате xml
7. Чтение данных из файла необходимо реализовать с помощью класса java.io.InputReader
8. Запись данных в файл необходимо реализовать с помощью класса java.io.PrintWriter
9. Все классы в программе должны быть задокументированы в формате javadoc.
10. 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:

1. help : вывести справку по доступным командам
2. info : вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)
3. show : вывести в стандартный поток вывода все элементы коллекции в строковом представлении
4. add element : добавить новый элемент в коллекцию
5. update id element : обновить значение элемента коллекции, id которого равен заданному
6. remove_by_id id : удалить элемент из коллекции по его id
7. clear : очистить коллекцию
8. save : сохранить коллекцию в файл
9. 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.
10. exit : завершить программу (без сохранения в файл)
11. insert_at index element : добавить новый элемент в заданную позицию
12. remove_greater id : удалить из коллекции все элементы, у которого id больше заданного
13. remove_lower id : удалить из коллекции все элементы, id которых меньше, чем заданный
14. sum_of_impact_speed‘ : вывести сумму значений поля impactSpeed для всех элементов коллекции
15. count_greater_than_car car : вывести количество элементов, значение поля car которых больше заданного

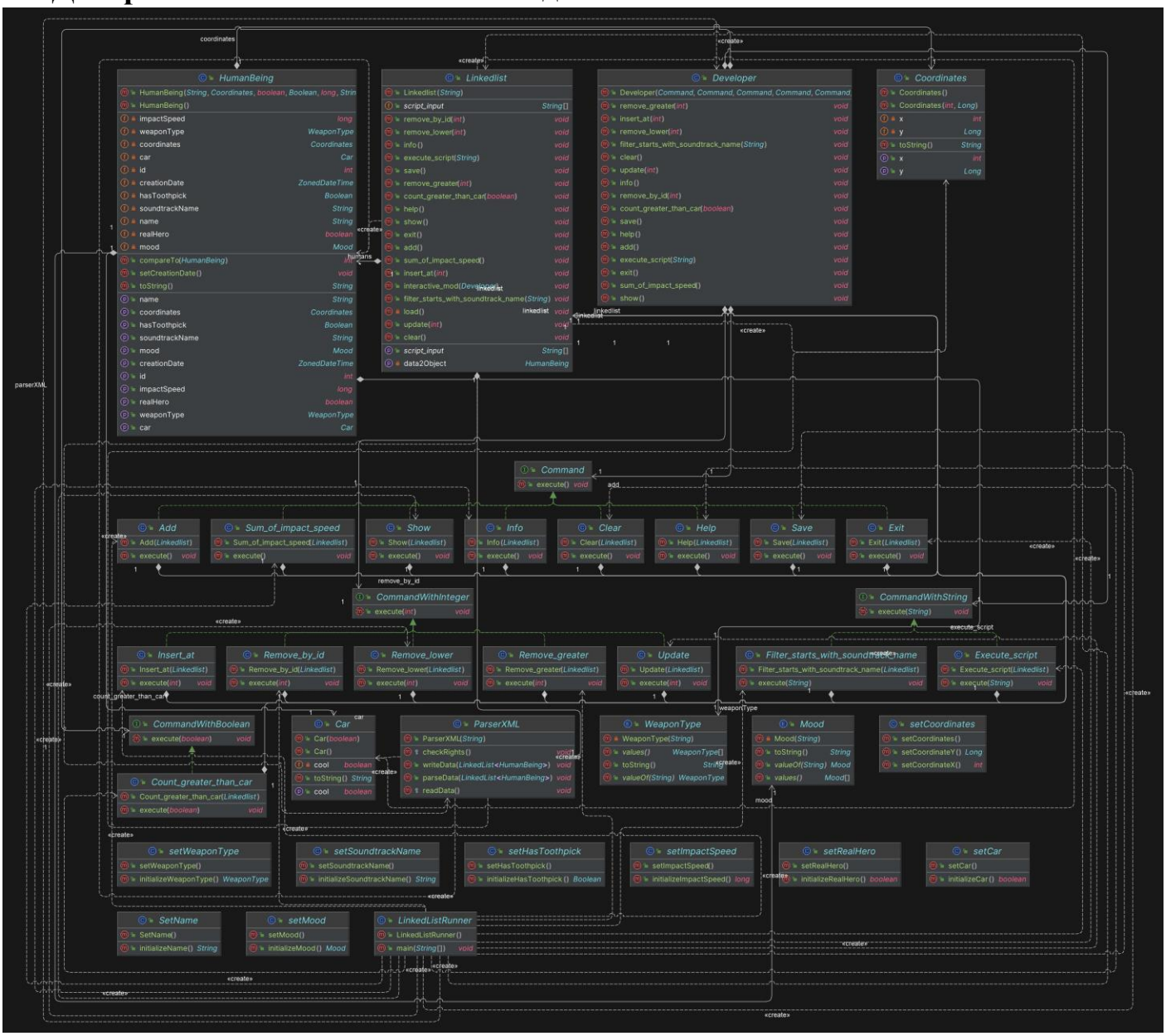
16. filter_starts_with_soundtrack_name soundtrackName : вывести элементы, значение поля name которых начинается с заданной подстроки

Формат ввода комманд:

1. 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.
2. Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.
3. При вводе составных типов данных пользователю должно показываться приглашение к вводу, содержащее имя поля (например, ”Введите дату рождения:”)
4. Если поле является enum’ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).
5. При некорректном пользовательском вводе (введена строка, не являющаяся именем константы в enum’е; введена строка вместо числа; введённое число не входит в указанные границы и т.п.) должно быть показано сообщение об ошибке и предложено повторить ввод поля.
6. Для ввода значений null использовать пустую строку.
7. 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 int 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.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически private boolean realHero; private Boolean hasToothpick; //Поле не может быть null private long impactSpeed; //Максимальное значение поля: 572 private String soundtrackName; //Поле не может быть null private WeaponType weaponType; //Поле может быть null private Mood mood; //Поле может быть null private Car car; //Поле не может быть null
}
public class Coordinates { private int x; private Long y; //Максимальное значение поля: 493, Поле не может быть null
} public class Car {
private boolean cool;
} public enum WeaponType {
SHOTGUN, RIFLE,
KNIFE,
MACHINE_GUN;
}
public enum Mood {
LONGING,
GLOOM,
FRENZY;
}

2 Диаграмма классов объектной модели



3 Исходный код программы

Код лежит на GitHub

4 Выводы

В данной лабораторной работе я изучил построение программы с помощью шаблона «Command». Ознакомился с библиотеками коллекций, побитового и посимвольного чтения, а также парсингом XML-фалйов. Научился пользоваться утилитой javadoc.