Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Отчёт

по лабораторной работе №5

вариант 2658

Выполнил: Тимошкин Р. В., группа P3131

Преподаватель: Абузов Я. А.

Санкт-Петербург 2024

Текст задания

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

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.TreeSet
- 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 json
- Чтение данных из файла необходимо реализовать с помощью класса java.io.InputStreamReader
- Запись данных в файл необходимо реализовать с помощью класса java.io.PrintWriter
- Все классы в программе должны быть задокументированы в формате javadoc.
- Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутствие 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)
- add_if_max {element}: добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- add_if_min {element}: добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
- remove_lower {element}: удалить из коллекции все элементы, меньшие, чем заданный
- group_counting_by_type: сгруппировать элементы коллекции по значению поля type, вывести количество элементов в каждой группе
- count_greater_than_character {character}: вывести количество элементов, значение поля character которых больше заданного
- filter_starts_with_name {name}: вывести элементы, значение поля name которых начинается с заданной подстроки

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'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.
- 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:

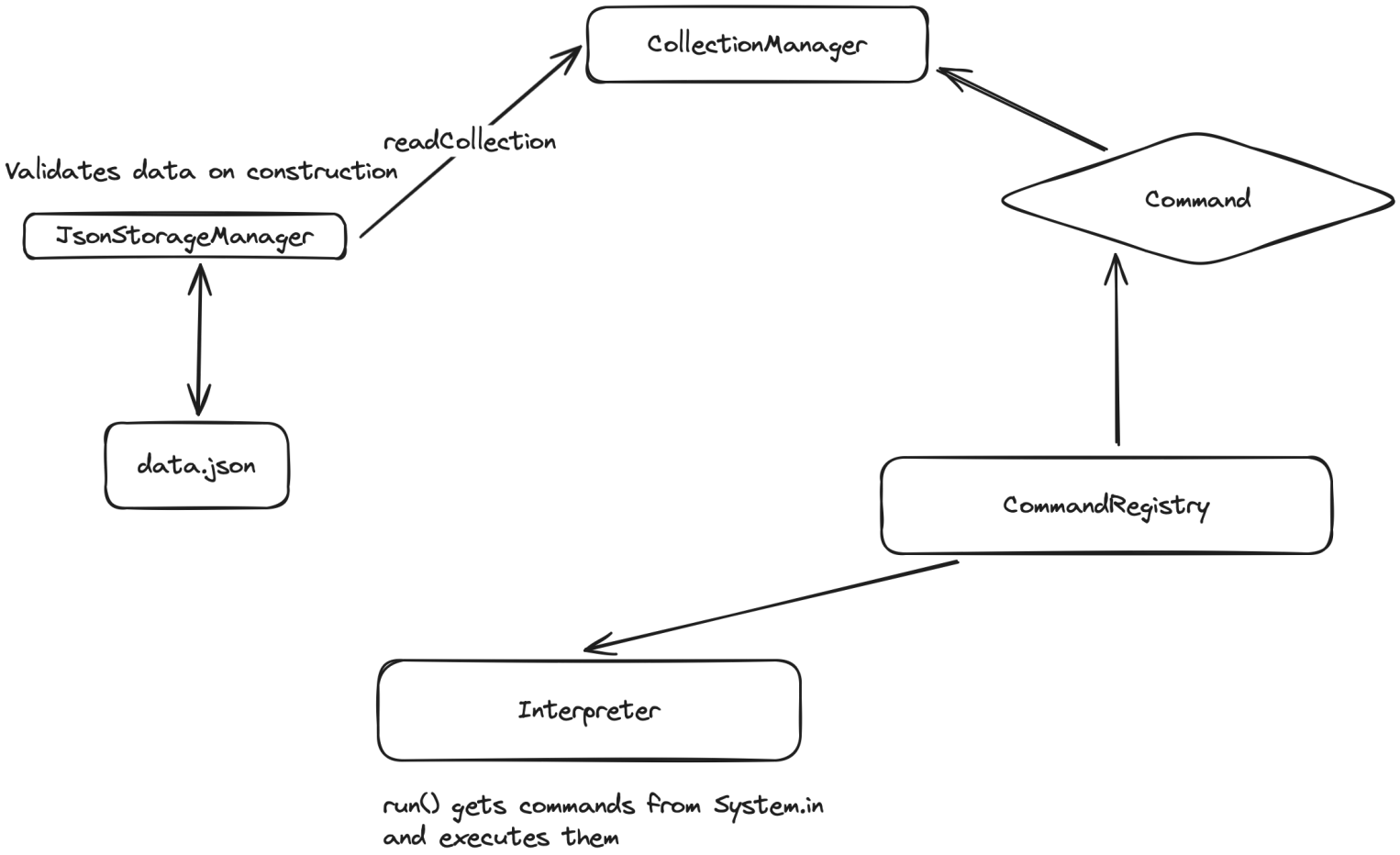
```
public class Dragon {
    private Integer id; //Поле не может быть null, Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
```

```
    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 Long age; //Значение поля должно быть больше 0, Поле не может быть null
    private Color color; //Поле может быть null
    private DragonType type; //Поле может быть null
    private DragonCharacter character; //Поле может быть null
    private DragonHead head;
}

public class Coordinates {
    private Float x; //Максимальное значение поля: 633, Поле не может быть null
    private float y; //Значение поля должно быть больше -408
}

public class DragonHead {
    private Double eyesCount; //Поле не может быть null
}

public enum Color {
    GREEN,
    ORANGE,
    BROWN
}

public enum DragonType {
    WATER,
    UNDERGROUND,
    AIR,
    FIRE
}

public enum DragonCharacter {
    EVIL,
    CHAOTIC,
    FICKLE
}
```

Исходный код программы

https://github.com/rmntim/ITMO/tree/main/Semester2/Programming/Labwork5

Архитектура программы



Вывод

Во время работы я взаимодействовал с коллекциями и Stream API. Узнал, как работают параметризованные типы и система ввода-вывода в Java. Во время написания лабораторной работы использовал паттерн Command и разобрался в архитектуре CLI приложений.