Санкт-Петербургский Национальный Исследовательский Университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники

Вариант № 32218
Лабораторная работа №5
По дисциплине
Программирование

Выполнил студент группы P3121:
Фам Данг Чунг Нгиа

Преподаватель:
Лисицина Василиса Васильевна

Санкт-Петербург 2024 г.

Оглавление

1. 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.HashSet
- При запуске приложения коллекция должна автоматически заполняться значениями из файла.
- Имя файла должно передаваться программе с помощью: аргумент командной строки.
- Данные должны храниться в файле в формате json
- Чтение данных из файла необходимо реализовать с помощью класса java.io.BufferedInputStream
- 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, 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)
- add_if_max {element} : добавить новый елемент в коллекциюЮ если его зрначение превышает значение наибольшего элемента этой коллекции
- add_if_max {element} : добавить новый елемент в коллекциюЮ если его зрначение меньше, чем у наименьшего элемента этой коллекции
- history : вывести последние 13 команд (без их аргументов)
- filter_contains_name name : вывести элементы, значение поля name которых содержит заданную подстроку
- print_descending : вывести элементы коллекции в порядке убывания
- print_field_descending_speaking : вывести значения поля speaking всех элементов 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'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 Long id; //Значение поля 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.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Integer age; //Значение поля должно быть больше 0
    private Float weight; //Значение поля должно быть больше 0, Поле может быть null
    private boolean speaking;
    private Color color; //Поле может быть null
    private DragonHead head;
}
public class Coordinates {
    private Integer x; //Максимальное значение поля: 830, Поле не может быть null
    private Long y; //Поле не может быть null
}
public class DragonHead {
    private Long eyesCount; //Поле может быть null
}
public enum Color {
    GREEN,
    YELLOW,
    WHITE;
}
```

2. Исходный код программы.

Репозиторий: https://github.com/nghiaphamhb/Programming/tree/main/Lab%205/src

3. Диаграмма классов разработанной программы

Programming/Lab 5/docs/diagram.png at main · nghiaphamhb/Programming (github.com)

4. Вывод

Во время выполнения данной лабораторной работы я научился работать с различными структурами данных в Java и файлами, а также углубил свои знания о ООП в Java, изучил параметризованные типы, wildcard-параметры и утилиту javadoc.