НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Программирование

Лабораторная работа № 5

Выполнил студент

Голоскок Д. С.

Группа № R3136

Преподаватель: Ладзин Артур Вячеславович

г. Санкт-Петербург

2021

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса LabWork, 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.LinkedList
* 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, 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)
* remove\_first : удалить первый элемент из коллекции
* 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\_any\_by\_personal\_qualities\_maximum personalQualitiesMaximum : удалить из коллекции один элемент, значение поля personalQualitiesMaximum которого эквивалентно заданному
* min\_by\_creation\_date : вывести любой объект из коллекции, значение поля creationDate которого является минимальным
* count\_by\_difficulty difficulty : вывести количество элементов, значение поля difficulty которых равно заданному

**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 LabWork {

private Long 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.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private int minimalPoint; //Значение поля должно быть больше 0

private Long personalQualitiesMaximum; //Поле может быть null, Значение поля должно быть больше 0

private Difficulty difficulty; //Поле может быть null

private Discipline discipline; //Поле может быть null

}

public class Coordinates {

private Long x; //Максимальное значение поля: 691, Поле не может быть null

private Long y; //Поле не может быть null

}

public class Discipline {

private String name; //Поле не может быть null, Строка не может быть пустой

private int practiceHours;

}

public enum Difficulty {

EASY,

IMPOSSIBLE,

INSANE,

TERRIBLE;

}

Выполнение работы

Helios: cd /home/s312458/ConsoleApp

Github: <https://github.com/BestAlice/ConsoleApp/tree/master>

Диаграмма классов на Github

Вывод:

В ходе работы я ближе познакомился с потоками данных, используя Scaner и InputStreamReader, а в случае чтения скриптов даже накладывал несколько потоков друг на друга. Также мне впервые пришлось самостоятельно разрабатывать довольно сложную для моего уровня систему, где важно всё правильно структурировать и реализовать. Полезным опытом стала работа с коллекцией и дженериками, без которых разработка ещё более сложных систем невозможна. Понял всю глубину человеческой глупости и при обработке возможных ошибок ввода.