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

Отчёт

По лабораторной работе №5

по дисциплине «Программирование»

Вариант: 6017

Работу выполнил:

Поленов Кирилл Александрович

Группа P3113

Работу принял:

Письмак Алексей Евгеньевич

г. Санкт-Петербург 2024

Оглавление

Задание

Лабораторная работа #5

Введите вариант: 6017

Внимание! У разных вариантов разный текст задания!

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Vehicle`, 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 `csv`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.BufferedReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.FileOutputStream`
- 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.).

- Данные должны храниться в файле в формате `csv`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.BufferedInputStream`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.BufferedOutputStream`

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
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `history` : вывести последние 5 команд (без их аргументов)
- `remove_any_by_engine_power enginePower` : удалить из коллекции один элемент, значение поля enginePower которого эквивалентно заданному
- `count_greater_than_distance_travelled distanceTravelled` : вывести количество элементов, значение поля distanceTravelled которых больше заданного
- `print_field_ascending_engine_power` : вывести значения поля enginePower всех элементов в порядке возрастания

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 Vehicle {
    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 float enginePower; //Значение поля должно быть больше 0
    private int capacity; //Значение поля должно быть больше 0
    private Float distanceTravelled; //Поле не может быть null, Значение поля должно быть больше 0
    private VehicleType type; //Поле не может быть null
}
public class Coordinates {
    private double x;
    private Integer y; //Поле не может быть null
}
public enum VehicleType {
    CAR,
    PLANE,
    HOVERBOARD;
}
```

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов разработанной программы.
3. Исходный код программы.
4. Выводы по работе.

Исходный код программы

Репозиторий на GitHub:

https://github.com/bilyardvmetro/ITMO-System-Application-Software/tree/main/Programming/2%20sem/Lab5/lab5

UML диаграмма реализованной объектной модели

Репозиторий на GitHub:

https://github.com/bilyardvmetro/ITMO-System-Application-Software/tree/main/Programming/2%20sem/Lab5

Выводы

В ходе данной лабораторной работы я:

- Познакомился со коллекциями в Java

- Познакомился с интерфейсами Comparator и Comparable

- Использовал паттерн проектирования “Command”

- Изучил символьные и байтовые потоки в Java