Национальный исследовательский университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники



Лабораторная работа №5 предмет

«**Программирование**»

Вариант:3114005

Группа: P3114

Студент: Дау Конг Туан Ань

Преподаватель: Письмак Алексей Евгеньевич

г. Санкт-Петербург

Aпреляь, 2022

**1. 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.ArrayDequeue
* 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 xml
* 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
* remove\_greater {element} : удалить из коллекции все элементы, превышающие заданный
* group\_counting\_by\_capacity : сгруппировать элементы коллекции по значению поля capacity, вывести количество элементов в каждой группе
* count\_less\_than\_fuel\_type fuelType : вывести количество элементов, значение поля fuelType которых меньше заданного
* filter\_less\_than\_fuel\_consumption fuelConsumption : вывести элементы, значение поля fuelConsumption которых меньше заданного

**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 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.util.Date creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private Double enginePower; //Поле не может быть null, Значение поля должно быть больше 0

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

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

private FuelType fuelType; //Поле не может быть null

}

public class Coordinates {

private Integer x; //Максимальное значение поля: 717, Поле не может быть null

private double y; //Максимальное значение поля: 12

}

public enum FuelType {

ELECTRICITY,

MANPOWER,

ANTIMATTER;

}

**2. Диаграмма классов объектной модели.**

**3. Исходный код программы.**

**4. Результат работы программы.**

**https://github.com/andrey551/lab5\_pr**

**5. Выводы по работе.**

**Я научился работать с коллекциями, как читать/записывать файлы, как создавать простые программы**