Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

**ЛАБОРАТОРНАЯ РАБОТА № 5**

**по дисциплине**

**“ПРОГРАММИРОВАНИЕ”**

**Вариант №** **1320**

Выполнил:

Вайнштейн Владислав Витальевич

Группа: Р3118

Преподаватель:

Кулинич Ярослав Вадимович

Г. Санкт-Петербург, 2023 г.

Задание

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.LinkedHashMap
* 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.util.Scanner
* 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.).

**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
* insert null {element} : добавить новый элемент с заданным ключом
* update id {element} : обновить значение элемента коллекции, id которого равен заданному
* remove\_key null : удалить элемент из коллекции по его ключу
* 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\_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный
* replace\_if\_greater null {element} : заменить значение по ключу, если новое значение больше старого
* remove\_lower\_key null : удалить из коллекции все элементы, ключ которых меньше, чем заданный
* average\_of\_engine\_power : вывести среднее значение поля enginePower для всех элементов коллекции
* min\_by\_coordinates : вывести любой объект из коллекции, значение поля coordinates которого является минимальным
* filter\_contains\_name name : вывести элементы, значение поля name которых содержит заданную подстроку

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

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

private VehicleType type; //Поле не может быть null

private FuelType fuelType; //Поле не может быть null

}

public class Coordinates {

private Float x; //Поле не может быть null

private Double y; //Значение поля должно быть больше -122, Поле не может быть null

}

public enum VehicleType {

CAR,

SUBMARINE,

SHIP,

CHOPPER,

SPACESHIP;

}

public enum FuelType {

GASOLINE,

ALCOHOL,

PLASMA;

}

**Отчёт по работе должен содержать:**

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

**Вопросы к защите лабораторной работы:**

1. Коллекции. Сортировка элементов коллекции. Интерфейсы java.util.Comparable и java.util.Comparator.
2. Категории коллекций - списки, множества. Интерфейс java.util.Map и его реализации.
3. Параметризованные типы. Создание параметризуемых классов. Wildcard-параметры.
4. Классы-оболочки. Назначение, область применения, преимущества и недостатки. Автоупаковка и автораспаковка.
5. Потоки ввода-вывода в Java. Байтовые и символьные потоки. "Цепочки" потоков (Stream Chains).
6. Работа с файлами в Java. Класс java.io.File.
7. Пакет java.nio - назначение, основные классы и интерфейсы.
8. Утилита javadoc. Особенности автоматического документирования кода в Java.

UML диаграмма

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описаниеИзображение выглядит как линия, диаграмма, График, Параллельный

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, документ, письмо

Автоматически созданное описание

Изображение выглядит как линия, Параллельный, снимок экрана, диаграмма

Автоматически созданное описание

Изображение выглядит как линия, снимок экрана, Параллельный, диаграмма

Автоматически созданное описание

Код программы:

Github:

[https://github.com/Vainslav/progLabs/tree/main/Lab5/](https://github.com/Vainslav/progLabs/tree/main/Lab4/src/main/java/org/example)src/main/java/lab/

Заключение

Узнал про java.collections, java.comparator и javadoc