Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

**Факультет программной инженерии**

**Лабораторная работа №5**

**Курса «Программирование»**

Вариант 315710

Выполнил: **Клименко  
Кирилл Владимирович**Группа: **Р3114**

Преподаватель:

**Письмак Алексей Евгеньевич**

**г. Санкт-Петербург**

2021 г

**Задание:**

**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 xml
* Чтение данных из файла необходимо реализовать с помощью класса java.io.BufferedReader
* Запись данных в файл необходимо реализовать с помощью класса java.io.BufferedOutputStream
* 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 элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
* remove\_greater {element} : удалить из коллекции все элементы, превышающие заданный
* remove\_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный
* filter\_starts\_with\_name name : вывести элементы, значение поля name которых начинается с заданной подстроки
* print\_field\_descending\_passport\_i\_d : вывести значения поля passportID всех элементов в порядке убывания
* print\_field\_descending\_height : вывести значения поля height 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 Person {

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

private double height; //Значение поля должно быть больше 0

private String passportID; //Поле не может быть null

private Color hairColor; //Поле не может быть null

private Country nationality; //Поле может быть null

private Location location; //Поле не может быть null

}

public class Coordinates {

private Long x; //Максимальное значение поля: 51, Поле не может быть null

private double y;

}

public class Location {

private long x;

private float y;

private Long z; //Поле не может быть null

}

public enum Color {

RED,

ORANGE,

BROWN;

}

public enum Country {

RUSSIA,

VATICAN,

SOUTH\_KOREA;

}

**Исходный код:**  
<https://github.com/souress/Lab5>

**UML-диаграмма:**

(UML для удобства просмотра выложена на гитхаб)

**Вывод:**

Реализовав данное консольное приложение, изучил шаблон Command, освоил работу с файлами, научился сортировать коллекции с помощью переопределенного метода compareTo интерфейса Comparable.