

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5

по дисциплине

«Программирование»

Вариант №3113017

Выполнил:

Студент группы Р3113

Султанов А.Р.

Проверил:

Письмак А.Е.

г. Санкт-Петербург

2023г.

Оглавление

Оглавление	2
Задание	3
Исходный код	7
Заключение	8

Задание

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Person`, 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.LinkedHashSet`

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 `xml`

Чтение данных из файла необходимо реализовать с помощью класса `java.util.Scanner`

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, отсутствие 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_min {element} : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции

remove_greater {element} : удалить из коллекции все элементы, превышающие заданный

remove_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный

remove_all_by_location location : удалить из коллекции все элементы, значение поля location которого эквивалентно заданному

count_greater_than_location location : вывести количество элементов, значение поля location которых больше заданного

print_field_descending_nationality : вывести значения поля nationality 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'e; введена строка вместо числа; введенное 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 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.LocalDateTime creationDate; //Поле не может быть  
    null, Значение этого поля должно генерироваться автоматически
```

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

private String passportID; //Строка не может быть пустой, Длина строки не должна быть больше 25, Поле может быть null

private Color eyeColor; //Поле не может быть null

private Country nationality; //Поле не может быть null

private Location location; //Поле не может быть null

}

public class Coordinates {

private Float x; //Значение поля должно быть больше -527, Поле не может быть null

private int y; //Максимальное значение поля: 897

}

public class Location {

private long x;

private Integer y; //Поле не может быть null

private String name; //Поле не может быть null

}

public enum Color {

RED,

ORANGE,

WHITE,

BROWN;

}

public enum Country {

THAILAND,

SOUTH_KOREA,

NORTH_KOREA;

}

Исходный код

<https://github.com/sultanowskii/itmo-edu/blob/master/programming/lab5/>

Заключение

В рамках данной лабораторной работы необходимо изучить работу с файлами, различными структурами данных, изучить утилиту `javadoc` и формат документации, а также ознакомиться с методами сравнения/сортировки объектов в `java`.