

Университет ИТМО
Факультет программной инженерии и компьютерной техники
Системное и прикладное программное обеспечение

Лабораторная работа №5 Вариант 41506

Выполнил: студент группы Р3115
Горинов Д.А.
Преподаватель: Горбунов М.В.

г. Санкт-Петербург,
2022

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.TreeMap`
- 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 `csv`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.InputStreamReader`
- 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, 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
- `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)
- `history` : вывести последние 12 команд (без их аргументов)

- `replace_if_lowe null {element}` : заменить значение по ключу, если новое значение меньше старого
- `remove_lower_key null` : удалить из коллекции все элементы, ключ которых меньше, чем заданный
- `filter_less_than_nationality nationality` : вывести элементы, значение поля `nationality` которых меньше заданного
- `filter_greater_than_height height` : вывести элементы, значение поля `height` которых больше заданного
- `print_field_descending_weight` : вывести значения поля `weight` 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 long 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 long height; //Значение поля должно быть больше 0
    private Float weight; //Поле может быть null, Значение поля должно быть больше
    0
    private Color hairColor; //Поле может быть null
    private Country nationality; //Поле может быть null
    private Location location; //Поле не может быть null
}
public class Coordinates {
```

```
    private Float x; //Значение поля должно быть больше -645, Поле не может быть
null
    private int y; //Максимальное значение поля: 727
}
public class Location {
    private int x;
    private double y;
    private Integer z; //Поле не может быть null
}
public enum Color {
    GREEN,
    RED,
    BLUE,
    YELLOW,
    BROWN;
}
public enum Country {
    USA,
    INDIA,
    ITALY,
    NORTH_KOREA;
}
```

Основные этапы решения

Ссылка на github:

https://github.com/Himezon/study/tree/main/Programming/Lab_5_NEW

Диаграмма классов (во избежание применения лупы, вы можете открыть диаграмму в формате png по ссылке на репозиторий лабораторной работы выше):



Вывод

Выполняя данную работу я на простом примере разобрался с применением коллекций, параметризованными типами, классами-оболочками, потоками ввода-вывода в Java. Помимо этого, научился работе с файлами и утилитой `javadoc`