**Информатика**

**Лабораторная работа №5**

**Вариант 5114**

Выполнил:

Полуянов Игорь Андреевич

Группа Р3110

Преподаватель:

Мустафаева А.

## Задание

Внимание! У разных вариантов разный текст задания!

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Worker, 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.HashSet
* 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 json
* Чтение данных из файла необходимо реализовать с помощью класса java.util.Scanner
* Запись данных в файл необходимо реализовать с помощью класса java.io.FileWriter
* 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\_greater {element} : удалить из коллекции все элементы, превышающие заданный
* remove\_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный
* history : вывести последние 13 команд (без их аргументов)
* max\_by\_position : вывести любой объект из коллекции, значение поля position которого является максимальным
* filter\_by\_end\_date endDate : вывести элементы, значение поля endDate которых равно заданному
* print\_descending : вывести элементы коллекции 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 Worker {

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 long salary; //Значение поля должно быть больше 0

private java.util.Date startDate; //Поле не может быть null

private java.time.ZonedDateTime endDate; //Поле может быть null

private Position position; //Поле не может быть null

private Organization organization; //Поле может быть null

}

public class Coordinates {

private Integer x; //Максимальное значение поля: 443, Поле не может быть null

private float y;

}

public class Organization {

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

private Address postalAddress; //Поле не может быть null

}

public class Address {

private String zipCode; //Длина строки должна быть не меньше 9, Поле может быть null

private Location town; //Поле не может быть null

}

public class Location {

private Float x; //Поле не может быть null

private Double y; //Поле не может быть null

private Double z; //Поле не может быть null

}

public enum Position {

MANAGER,

ENGINEER,

MANAGER\_OF\_CLEANING;

}

## Диаграмма классов

Изображение выглядит как снимок экрана, текст, Графическое программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

## Исходный код

Ссылка: [ITMO/lab5 at main · PoluyanbIch/ITMO (github.com)](https://github.com/PoluyanbIch/ITMO/tree/main/lab2)

## Вывод