МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

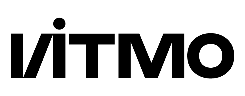
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И   
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ОТЧЁТ**  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине  
«ПРОГРАММИРОВАНИЕ»  
Вариант № 31149

**Выполнил:**Студент группы P3114  
Минкова Алина Андреевна  
**Преподаватель:**Инячина Диана Александровна



Санкт-Петербург, 2023

СОДЕРЖАНИЕ

[ЗАДАНИЕ 3](#_Toc132051902)

[ИСХОДНЫЙ КОД ПРОГРАММЫ 6](#_Toc132051903)

[ДОКУМЕНТАЦИЯ JAVADOC 6](#_Toc132051905)

[ДИАГРАММА КЛАССОВ РЕАЛИЗОВАННОЙ ОБЪЕКТНОЙ МОДЕЛИ 6](#_Toc132051906)

[ВЫВОД 6](#_Toc132051907)

ЗАДАНИЕ

До сброса вариантов, был следующий:

* 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.ArrayDequeue
* 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.io.InputStreamReader
* 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.).

**Команды:**

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 превышает значение наибольшего элемента этой коллекции  
head : вывести первый элемент коллекции  
remove\_greater {element} : удалить из коллекции все элементы, превышающие заданный  
remove\_lower\_id id : удалить из коллекции все элементы, id которых меньше, чем заданный  
filter\_by\_status status : вывести элементы, значение поля status которых равно заданному  
filter\_starts\_with\_name name : вывести элементы, значение поля name которых начинается с заданной подстроки

**Описание хранимых классов:**

public class Worker {

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 salary; //Значение поля должно быть больше 0

private java.time.LocalDate startDate; //Поле не может быть null

private java.time.LocalDate startDate; //Поле не может быть null

private Position position; //Поле не может быть null

private Status status; //Поле может быть null

private Person person; //Поле не может быть null

}

public class Coordinates {

private long x;

private float y; //Максимальное значение поля: 776

}

public class Person {

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

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 Location {

private Integer x; //Поле не может быть null

private Float y; //Поле не может быть null

private String name; //Строка не может быть пустой, Поле не может быть null

}

public enum Position {

HUMAN\_RESOURCES,

ENGINEER,

HEAD\_OF\_DIVISION,

DEVELOPER,

MANAGER\_OF\_CLEANING; }

public enum Status {

HIRED,

RECOMMENDED\_FOR\_PROMOTION,

REGULAR,

PROBATION; }

public enum Color {

GREEN,

BLACK,

YELLOW,

ORANGE; }

public enum Country {

GERMANY,

VATICAN,

ITALY,

THAILAND,

JAPAN; }

ИСХОДНЫЙ КОД ПРОГРАММЫ

<https://github.com/aulouu/lab5_proga>

ДОКУМЕНТАЦИЯ JAVADOC

<https://se.ifmo.ru/~s373317/>

ДИАГРАММА КЛАССОВ РЕАЛИЗОВАННОЙ ОБЪЕКТНОЙ МОДЕЛИ

На гите diagrams.png

<https://github.com/aulouu/lab5_proga/blob/master/diagrams.png>

ВЫВОД

Во время выполнения работы я ознакомился с синтаксисом языка Java, ещё лучше узнал преимущества ООП, познакомился с ним на практике, разобрался что такое Command pattern, разобрался с коллекциями Java и их обработкой, написал интерактивную программу, создал объектную модель по тексту. Полученные знания понадобятся в процессе дальнейшего обучения.