

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники

Вариант №31192
Лабораторная работа №5
по дисциплине
’Программирование’

Выполнил Студент группы Р3118:
Михеев Илья
Преподаватель:
Кулинич Ярослав Владимирович

г. Санкт-Петербург
2023г.

Содержание

1	Текст задания	3
2	Диаграмма классов	4
3	Исходный код программы	4
4	Вывод	4

1 Текст задания

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.TreeSet`
- 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.InputStreamReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.OutputStreamWriter`
- 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_max {element}`: добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- `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
- `filter_less_than_position position`: вывести элементы, значение поля `position` которых меньше заданного
- `print_descending`: вывести элементы коллекции в порядке убывания
- `print_unique_organization`: вывести уникальные значения поля `organization` всех элементов в коллекции

Описание хранимых в коллекции классов:

```
public class Worker {
    private Long 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.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Float salary; //Поле не может быть null, Значение поля должно быть больше 0
    private java.util.Date startDate; //Поле не может быть null
    private java.time.LocalDateTime endDate; //Поле может быть null
    private Position position; //Поле может быть null
    private Organization organization; //Поле не может быть null
}

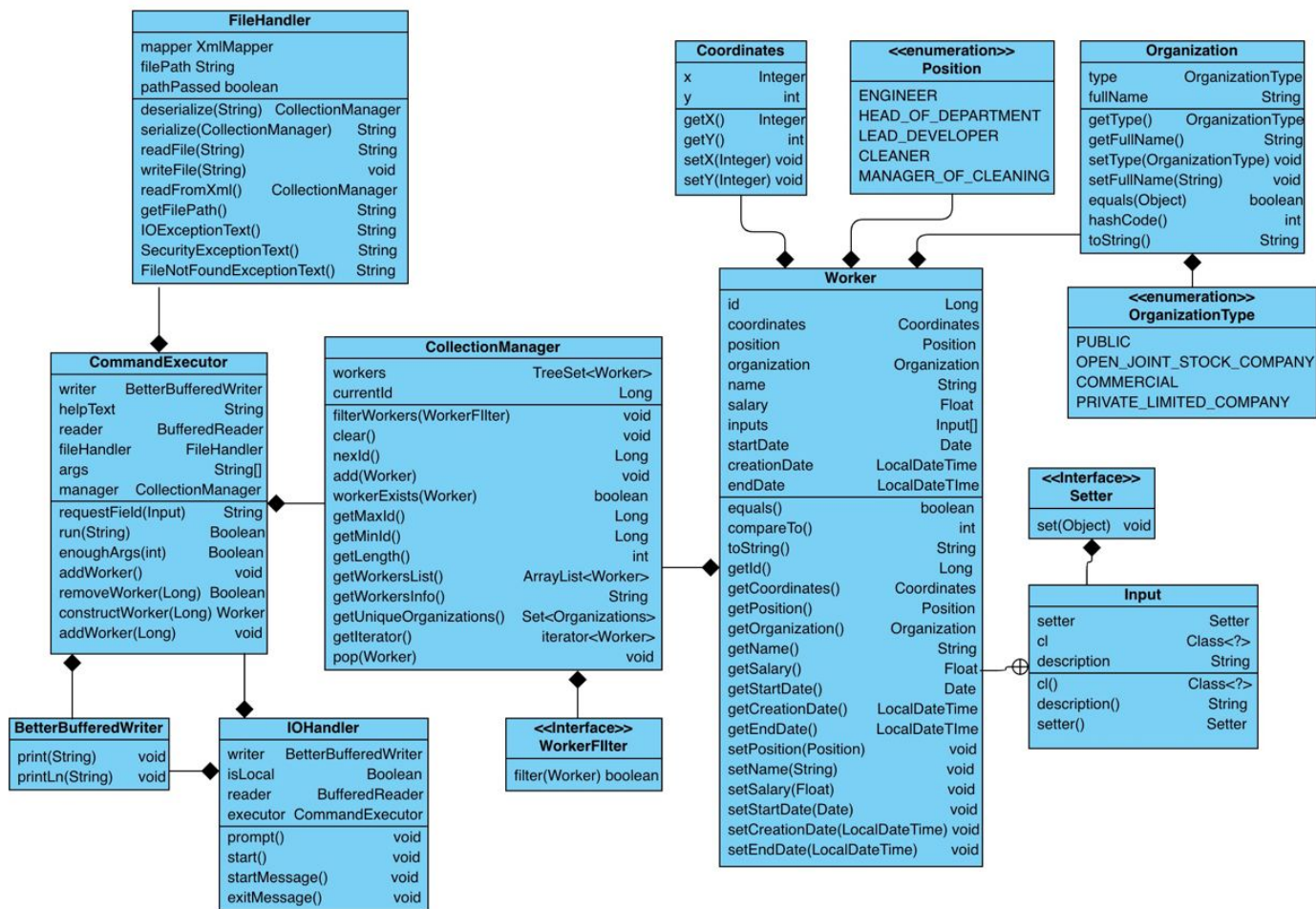
public class Coordinates {
    private Integer x; //Значение поля должно быть больше -720, Поле не может быть null
    private int y;
}

public class Organization {
    private String fullName; //Поле не может быть null
    private OrganizationType type; //Поле не может быть null
}

public enum Position {
    ENGINEER,
    HEAD_OF_DEPARTMENT,
    LEAD_DEVELOPER,
    CLEANER,
    MANAGER_OF_CLEANING;
}

public enum OrganizationType {
    COMMERCIAL,
    PUBLIC,
    PRIVATE_LIMITED_COMPANY,
    OPEN_JOINT_STOCK_COMPANY;
}
```

2 Диаграмма классов



3 Исходный код программы

<https://github.com/Ne0Ment/ITMO-proga/blob/main/lab5/src/>

4 Вывод

В ходе выполнения данной лабораторной работы я научился работ с исключениями Java, познакомился с внутренними и анонимными классами и применил их на практике.