

Федеральное государственное
автономное учебное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной техники

Отчёт
по лабораторной работе №5
по дисциплине «Программирование»

Вариант 31187

Группа: Р3118

Студенты: Иванова Александра Юрьевна,
Кожухин Иван Алексеевич

Преподаватель: Кулинич Ярослав Вадимович

Санкт-Петербург
2023

Содержание

1	Задание	1
2	Выполнение работы	3
3	Вывод	4

1 Задание

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.LinkedList`
- 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.BufferedReader`
- 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, отсутствие прав доступа к файлу и т.п.).

Рис. 1: Текст задания варианта, ч. 1

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_first` : удалить первый элемент из коллекции
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `history` : вывести последние 9 команд (без их аргументов)
- `count_greater_than_status status` : вывести количество элементов, значение поля `status` которых больше заданного
- `filter_starts_with_name name` : вывести элементы, значение поля `name` которых начинается с заданной подстроки
- `print_descending` : вывести элементы коллекции в порядке убывания

Рис. 2: Текст задания варианта, ч. 2

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; 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 Long id; //Поле не может быть null, Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно
    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; //Поле может быть null, Значение поля должно быть больше 0
    private java.time.ZonedDateTime startDate; //Поле не может быть null
    private Position position; //Поле может быть null
    private Status status; //Поле не может быть null
    private Organization organization; //Поле может быть null
}
```

Рис. 3: Текст задания варианта, ч. 3

```
public class Coordinates {
    private int x;
    private double y;
}
public class Organization {
    private String fullName; //Длина строки не должна быть больше 1576, Значение этого поля должно быть уникальным, Поле не может быть null
    private double annualTurnover; //Значение поля должно быть больше 0
    private OrganizationType type; //Поле может быть null
}
public enum Position {
    HUMAN_RESOURCES,
    HEAD_OF_DIVISION,
    HEAD_OF_DEPARTMENT,
    BAKER;
}
public enum Status {
    FIRED,
    HIRED,
    RECOMMENDED_FOR_PROMOTION;
}
public enum OrganizationType {
    COMMERCIAL,
    PUBLIC,
    PRIVATE_LIMITED_COMPANY,
    OPEN_JOINT_STOCK_COMPANY;
}
```

Рис. 4: Текст задания варианта, ч. 4

2 Выполнение работы

Ссылка на репозиторий на GitHub

Ссылка диаграмму классов программы

(перейдя по ссылке, следует нажать «Открыть в приложении»)

3 Вывод

В ходе выполнения лабораторной работы были изучены: различные интерфейсы коллекций и их реализация на языке Java; элементы обобщённого программирования - обобщённые и параметризованные типы, wildcard-параметры, подстановки; классы-оболочки, их назначение, преимущества и недостатки; потоки ввода и вывода в Java, их виды и особенности; работа с файлами в Java; пакеты `java.io` и `java.nio`; утилита `javadoc` и особенности автоматического документирования кода в Java.