Университет ИТМО

Факультет программной инженерии и компьютерной техники
Образовательная программа системное и прикладное программное обеспечение

Лабораторная работа №5

По дисциплине "Программирование"
Вариант 3801

Выполнил студент группы P3109
Евграфов Артём Андреевич
Проверила:
Мустафаева Айнур Вугар Кызы

Санкт-Петербург 2025

Содержание

1. Задание варианта 3801

- 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 `json`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.BufferedReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.PrintWriter`
- 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_head` : вывести первый элемент коллекции и удалить его
- `remove_lower {element}` : удалить из коллекции все элементы, меньшие, чем заданный
- `history` : вывести последние 8 команд (без их аргументов)
- `remove_all_by_group_admin groupAdmin` : удалить из коллекции все элементы, значение поля groupAdmin которого эквивалентно заданному
- `count_greater_than_group_admin groupAdmin` : вывести количество элементов, значение поля groupAdmin которых больше заданного
- `filter_greater_than_form_of_education formOfEducation` : вывести элементы, значение поля formOfEducation которых больше заданного

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 предложен
- 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 StudyGroup {
    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.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private long studentsCount; //Значение поля должно быть больше 0
    private int expelledStudents; //Значение поля должно быть больше 0
    private long shouldBeExpelled; //Значение поля должно быть больше 0
    private FormOfEducation formOfEducation; //Поле может быть null
    private Person groupAdmin; //Поле может быть null
}
public class Coordinates {
    private double x;
    private double y;
}
public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private java.time.LocalDateTime birthday; //Поле не может быть null
    private long height; //Значение поля должно быть больше 0
    private int weight; //Значение поля должно быть больше 0
    private String passportID; //Длина строки не должна быть больше 47, Строка не может быть пустой, Поле не может быть null
}
public enum FormOfEducation {
    DISTANCE_EDUCATION,
    FULL_TIME_EDUCATION,
    EVENING_CLASSES;
}
```

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов разработанной программы.
3. Исходный код программы.
4. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Коллекции. Сортировка элементов коллекции. Интерфейсы `java.util.Comparable` и `java.util.Comparator`.
2. Категории коллекций - списки, множества. Интерфейс `java.util.Map` и его реализации.
3. Параметризованные типы. Создание параметризуемых классов. Wildcard-параметры.
4. Классы-оболочки. Назначение, область применения, преимущества и недостатки. Автоупаковка и автораспаковка.
5. Потоки ввода-вывода в Java. Байтовые и символьные потоки. "Цепочки" потоков (Stream Chains).
6. Работа с файлами в Java. Класс `java.io.File`.
7. Пакет `java.nio` - назначение, основные классы и интерфейсы.
8. Утилита `javadoc`. Особенности автоматического документирования кода в Java.

2. Исходный код

Исходный код можно найти по ссылке на гитхабе

3. UML диаграмма

Диаграмму можно найти по ссылке на гитхабе

4. Вывод

В ходе выполнения лабораторной работы я:
Познакомился с коллекциями в Java
Познакомился с интерфейсами Comparator и Comparable
Использовал паттерн проектирования “Command”
Изучил символьные и байтовые потоки в Java