



Факультет Программной Инженерии и Компьютерной техники

Программирование

Лабораторная работа №5

Вариант: 31321

Выполнил: Малых Кирилл Романович

Группа: Р3132

Преподаватель: Данилов Павел Юрьевич

Санкт-Петербург, 2026г.

## **Содержание**

1. Условие поставленной задачи .....	3
3. Диаграмма классов разработанной программы .....	6
4. Код программы .....	7
5. Заключение.....	8
6. Список использованной литературы.....	9

## **1. Условие поставленной задачи**

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса LabWork, описание которого приведено ниже.

**Разработанная программа должна удовлетворять следующим требованиям:**

1. Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
2. Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.
3. Для хранения необходимо использовать коллекцию типа `java.util.LinkedHashMap`
4. При запуске приложения коллекция должна автоматически заполняться значениями из файла.
5. Имя файла должно передаваться программе с помощью: **переменная окружения**.
6. Данные должны храниться в файле в формате json
7. Чтение данных из файла необходимо реализовать с помощью класса `java.util.Scanner`
8. Запись данных в файл необходимо реализовать с помощью класса `java.io.OutputStreamWriter`
9. Все классы в программе должны быть задокументированы в формате javadoc.
10. 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:**

1. `help`: вывести справку по доступным командам
2. `info`: вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)
3. `show`: вывести в стандартный поток вывода все элементы коллекции в строковом представлении

4. `insert null {element}`: добавить новый элемент с заданным ключом
5. `update id {element}`: обновить значение элемента коллекции, `id` которого равен заданному
6. `remove_key null`: удалить элемент из коллекции по его ключу
7. `clear`: очистить коллекцию
8. `save`: сохранить коллекцию в файл
9. `execuse_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.
10. `exit`: завершить программу (без сохранения в файл)
11. `remove_lower {element}`: удалить из коллекции все элементы, меньшие, чем заданный
12. `history`: вывести последние 14 команд (без их аргументов)
13. `replace_if_greater null {element}`: заменить значение по ключу, если новое значение больше старого
14. `group_counting_by_minimal_point`: сгруппировать элементы коллекции по значению поля `minimalPoint`, вывести количество элементов в каждой группе
15. `count_less_than_difficulty difficulty`: вывести количество элементов, значение поля `difficulty` которых меньше заданного
16. `print_field_descending_minimal_point`: вывести значения поля `minimalPoint` всех элементов в порядке убывания

### **Формат ввода команд:**

1. 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.
2. Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.
3. При вводе составных типов данных пользователю должно показываться приглашение к вводу, содержащее имя поля (например, "Введите дату рождения:")
4. Если поле является enum'ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).

5. 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.
6. Для ввода значений null использовать пустую строку.
7. Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.

### **Описание хранимых в коллекции классов:**

[https://github.com/tetraminomusic/ITMO\\_2sem/blob/main/Прога/ЛБ5/Классы%20из%20коллекций.java](https://github.com/tetraminomusic/ITMO_2sem/blob/main/Прога/ЛБ5/Классы%20из%20коллекций.java)

### **3. Диаграмма классов разработанной программы**

Изображение диаграммы:

[https://github.com/tetraminomusic/ITMO\\_2sem/blob/main/Прога/ЛБ5/UML\\_lab5.png](https://github.com/tetraminomusic/ITMO_2sem/blob/main/Прога/ЛБ5/UML_lab5.png)

Исходный код диаграммы:

[https://github.com/tetraminomusic/ITMO\\_2sem/blob/main/Прога/ЛБ5/Lab5\\_bms/UML\\_lab5.puml](https://github.com/tetraminomusic/ITMO_2sem/blob/main/Прога/ЛБ5/Lab5_bms/UML_lab5.puml)

## **4. Код программы**

Исходный код программы:

[https://github.com/tetraminomusic/ITMO\\_2sem/tree/main/Прога/ЛБ5/Lab5\\_bms/src](https://github.com/tetraminomusic/ITMO_2sem/tree/main/Прога/ЛБ5/Lab5_bms/src)

Скомпилированный Jar-файл:

[https://github.com/tetraminomusic/ITMO\\_2sem/blob/main/Прога/ЛБ5/Lab5\\_bms.jar](https://github.com/tetraminomusic/ITMO_2sem/blob/main/Прога/ЛБ5/Lab5_bms.jar)

Удобный скрипт для запуска jar-архива с установкой переменной окружения и созданием файла json расширения:

[https://github.com/tetraminomusic/ITMO\\_2sem/blob/main/Прога/ЛБ5/five.sh](https://github.com/tetraminomusic/ITMO_2sem/blob/main/Прога/ЛБ5/five.sh)

## **5. Заключение**

В ходе данной лабораторной работы я ещё глубже окунулся в объектно-ориентированное программирование: я научился создать все более и более сложные по структуре программы. Также я стал более опытным при обработке данных, записываемых или получаемых в/из файлов благодаря Scanner в Java. Впервые познакомился с потоками stream, при помощи которых смог воспользоваться инструментами сортировки и фильтрации элементов коллекции по определённым атрибутам. Я более подробно изучил новые непримитивные типы данных: коллекции, классы-обертки. Также я в первый раз для себя открыл утилиту Javadoc, которая позволила мне автоматически составить документацию к программе.

## **6. Список использованной литературы**

- 1) Хидео Кодзима – “Хидео Кодзима. Гены гения” – 320 с.
- 2) Роберт Мартин – “Чистая архитектура. Искусство разработки программного обеспечения” – 352 с.
- 3) Грейди Буч, Джеймс Рамбо, Айвар Якобсон – “Язык UML. Руководство пользователя” – 496 с.
- 4) Мартин Фаулер – “Рефакторинг. Улучшение существующего кода” – 448 с.
- 5) Аллан Шаллоуэй, Джеймс Р. Тrott – “Паттерны проектирования. Новый подход к объектно-ориентированному анализу и проектированию” – 288 с.