

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

*Факультет программной инженерии и компьютерной техники*  
*Направление подготовки: 09.03.04 – Программная инженерия,*  
*Системное и прикладное программное обеспечение.*  
*Дисциплина «Программирование»*

**Отчет**  
**По лабораторной работе №5**  
**Программа на языке Java**

**Вариант №5991**

Выполнил:  
Молчанов Фёдор Денисович

Группа: P3113

Преподаватель: Письмак Алексей Евгеньевич

Г. Санкт-Петербург, 2024 г.

## Задание

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `MusicBand`, 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.PriorityQueue`

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 `xml`

Чтение данных из файла необходимо реализовать с помощью класса `java.io.InputStreamReader`

Запись данных в файл необходимо реализовать с помощью класса `java.io.BufferedWriter`

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 id : 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_lower {element} : удалить из коллекции все элементы, меньшие, чем
заданный
history : вывести последние 14 выполненных команд
min_by_number_of_participants : вывести объект из коллекции, значение поля
numberOfParticipants которого является минимальным
```

group\_counting\_by\_id id : вывести количество элементов, значение поля id которых меньше 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'e; введена строка вместо числа; введенное 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 MusicBand {
    private long 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.util.Date creationDate; //Поле не может быть null,
Значение этого поля должно генерироваться автоматически
    private int numberOfParticipants; //Поле не может быть null, Значение
поля должно быть больше 0
    private Long albumsCount; //Поле может быть null, Значение поля должно
быть больше 0
    private java.time.ZonedDateTime establishmentDate; //Поле может быть
null
    private MusicGenre genre; //Поле не может быть null
    private Label label; //Поле может быть null
}
public class Coordinates {
    private int x;
    private int y;
}
public class Label {
    private String name;
```

```
        private Long bands; //Поле не может быть null
    }
    public enum MusicGenre {
        PSYCHEDELIC_ROCK,
        PSYCHEDELIC_CLOUD_RAP,
        SOUL,
        BLUES,
        PUNK_ROCK;
    }
}
```

## ## Отчёт и вопросы

Отчёт по работе должен содержать:

- Текст задания.
- Диаграмма классов разработанной программы.
- Исходный код программы.
- Выводы по работе.

Вопросы к защите лабораторной работы:

- Коллекции. Сортировка элементов коллекции. Интерфейсы `java.util.Comparable` и `java.util.Comparator`.

- Категории коллекций - списки, множества. Интерфейс `java.util.Map` и его реализации.

- Параметризованные типы. Создание параметризуемых классов. Wildcard-параметры.

- Классы-оболочки. Назначение, область применения, преимущества и недостатки. Автоупаковка и автораспаковка.

- Потоки ввода-вывода в Java. Байтовые и символьные потоки. "Цепочки" потоков (Stream Chains).

- Работа с файлами в Java. Класс `java.io.File`.

- Пакет `java.nio` - назначение, основные классы и интерфейсы.

- Утилита `javadoc`. Особенности автоматического документирования кода в Java.

## Выполнение задания

Исходный код доступен по ссылке: <https://github.com/fe fumo/itmo/tree/main/Prog/lab5>

## Выводы

Во время выполнения данной лабораторной работы я научился работать с различными структурами данных в Java и файлами, а также углубил свои знания о ООП в Java, изучил параметризованные типы, wildcard-параметры и утилиту javadoc.