

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
"Санкт-Петербургский национальный исследовательский университет ИТМО"

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Лабораторная работа
по дисциплине
"ПРОГРАММИРОВАНИЕ"
II часть

Вариант №22236587

Выполнил:
Студент группы Р3118
Павлов Александр Сергеевич
Преподаватель:
Сорокин Роман Борисович



Санкт-Петербург, 2022

Текст задания:

Лабораторная работа #5

Введите вариант: 2223658

Внимание! У разных вариантов разный текст задания!

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Dragon`, 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.LinkedHashMap`
- 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 `json`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.InputStreamReader`
- 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, 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
- `insert null {element}`: добавить новый элемент с заданным ключом
- `update id {element}`: обновить значение элемента коллекции, id которого равен заданному
- `remove_key null`: удалить элемент из коллекции по его ключу
- `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)
- `history`: вывести последние 14 команд (без их аргументов)
- `replace_if_greater null {element}`: заменить значение по ключу, если новое значение больше старого
- `replace_if_lowe null {element}`: заменить значение по ключу, если новое значение меньше старого
- `average_of_wingspan`: вывести среднее значение поля `wingspan` для всех элементов коллекции
- `count_less_than_color color`: вывести количество элементов, значение поля `color` которых меньше заданного
- `filter_less_than_weight weight`: вывести элементы, значение поля `weight` 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`'е; 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:

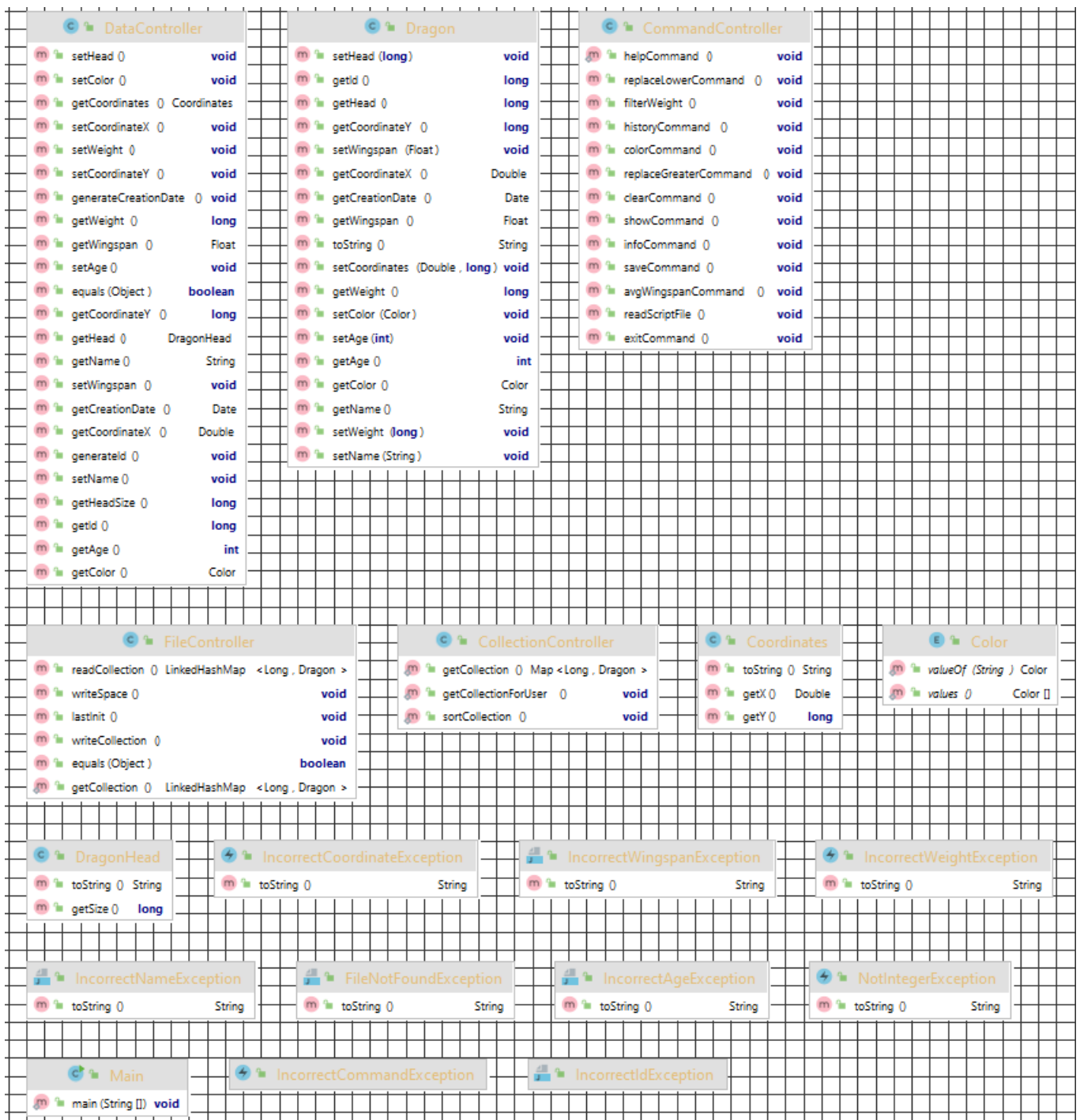
```
public class Dragon {
    private long id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.util.Date creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private int age; //Значение поля должно быть больше 0
    private Float wingspan; //Значение поля должно быть больше 0, Поле не может быть null
    private long weight; //Значение поля должно быть больше 0
    private Color color; //Поле не может быть null
    private DragonHead head;
}

public class Coordinates {
    private Double x; //Поле не может быть null
    private long y; //Значение поля должно быть больше -666
}

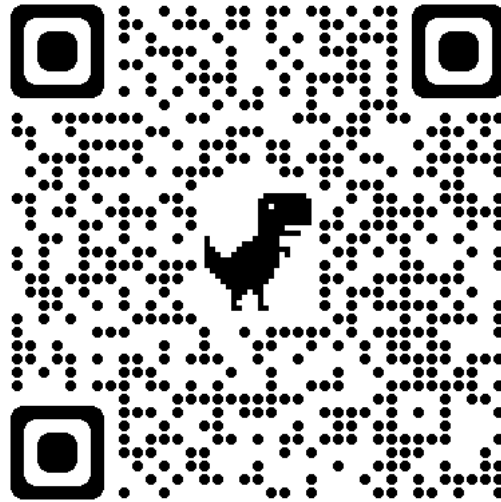
public class DragonHead {
    private long size;
}

public enum Color {
    RED,
    BLACK,
    ORANGE,
    BROWN;
}
```

Диаграмма классов:



Исходный код программы:
QR-код на GitHub:



https://github.com/ExcaliBBur/prog_labs/tree/main/lab5

Вывод:

В ходе выполнения данной лабораторной работы я улучшил навыки работы с ООП, узнал о работе с файлами и познакомился с коллекциями в Java, также познакомился с интерфейсами Comparable и Comparator.