Санкт-Петербургский Национальный Исследовательский

Университет Информационных Технологий, Механики и Оптики

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

по дисциплине

“Программирование”

Вариант:54064

Выполнил

Студент группы P3116

Чэнь Хаолинь : 407960

**Текст задания**

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 **xml**
* Чтение данных из файла необходимо реализовать с помощью класса **java.io.FileReader**
* Запись данных в файл необходимо реализовать с помощью класса **java.io.BufferedOutputStream**
* 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  
**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)  
**remove\_greater {element}** : удалить из коллекции все элементы, превышающие заданный  
**remove\_lower {element}** : удалить из коллекции все элементы, меньшие, чем заданный  
**remove\_greater\_key null** : удалить из коллекции все элементы, ключ которых превышает заданный  
**min\_by\_creation\_date** : вывести любой объект из коллекции, значение поля creationDate которого является минимальным  
**filter\_contains\_name name** : вывести элементы, значение поля name которых содержит заданную подстроку  
**filter\_less\_than\_hair\_color hairColor** : вывести элементы, значение поля hairColor 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; 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 Person {

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 long height; //Значение поля должно быть больше 0

private java.time.LocalDateTime birthday; //Поле не может быть null

private Color eyeColor; //Поле не может быть null

private Color hairColor; //Поле не может быть null

private Location location; //Поле не может быть null

}

public class Coordinates {

private float x;

private long y; //Максимальное значение поля: 797

}

public class Location {

private Double x; //Поле не может быть null

private Integer y; //Поле не может быть null

private Integer z; //Поле не может быть null

}

public enum Color {

RED,

BLACK,

YELLOW,

BROWN;

}

public enum Color {

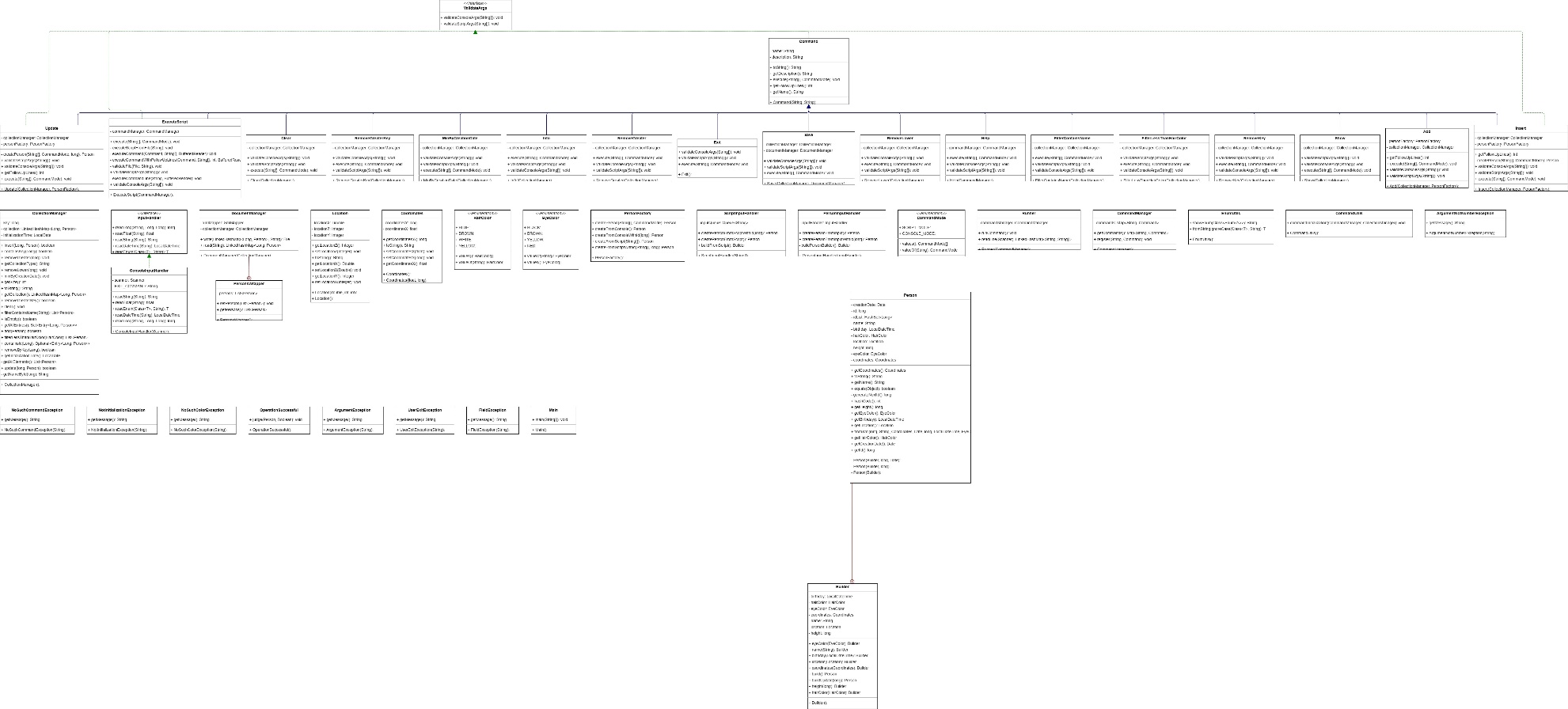
BLUE,

YELLOW,

WHITE,

BROWN;

}

**UML**

**Исходный код программы**

<https://github.com/special-meme/labs-utpwtb/tree/new_branch>

**Выводы по работе**

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.