Университет ИТМО

Факультет программной инженерии и компьютерной техники

Системное и прикладное программное обеспечение

Лабораторная работа №5

Вариант 41512

Выполнил: студент группы P3115

Киселев М.К.

Преподаватель: Горбунов М.В.

г. Санкт-Петербург,

2022

# Задание

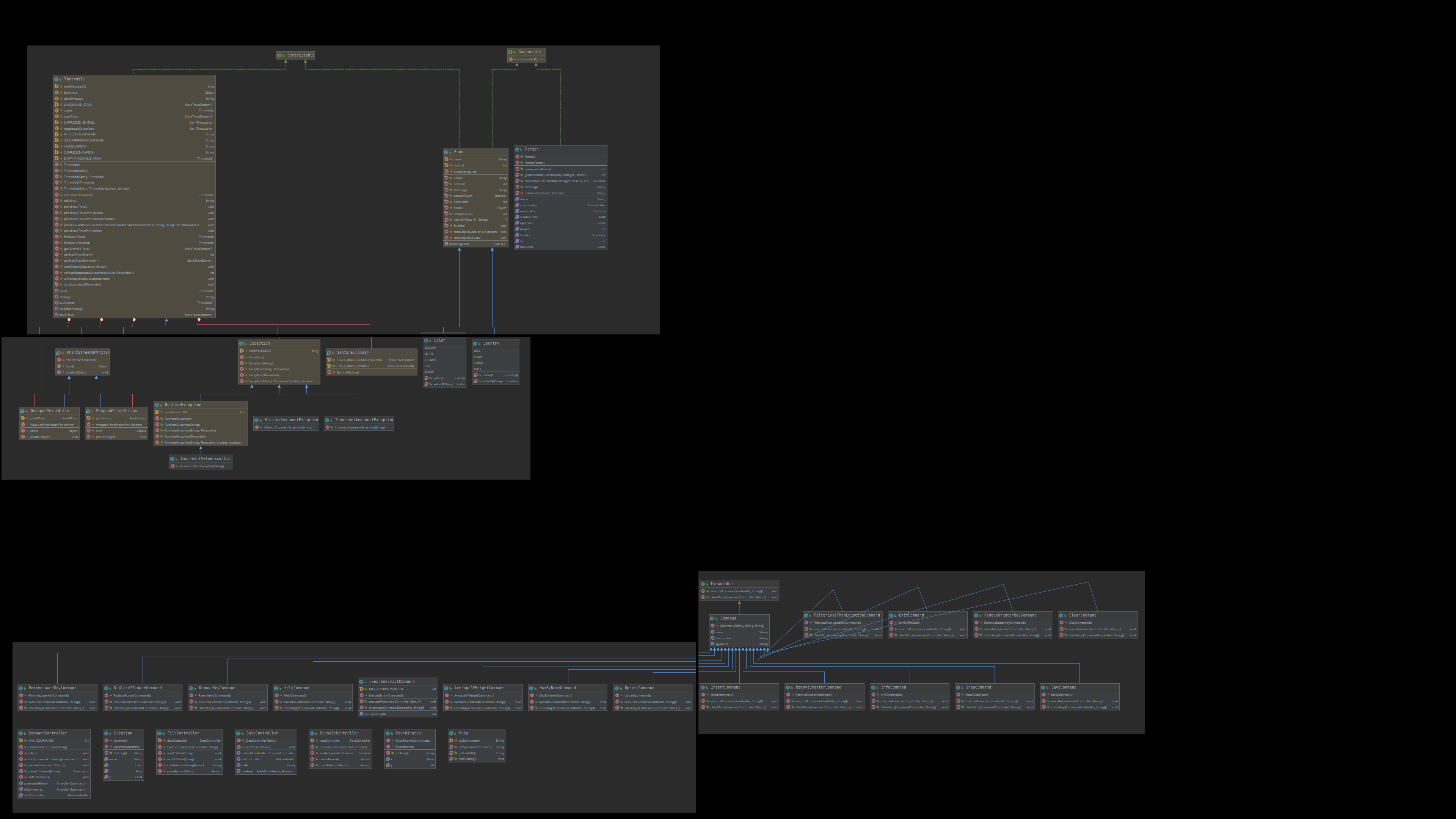
Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Person, описание которого приведено ниже.

|  |
| --- |
| **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.TreeMap * 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 csv * Чтение данных из файла необходимо реализовать с помощью класса java.io.InputStreamReader * Запись данных в файл необходимо реализовать с помощью класса java.io.FileOutputStream * 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 * replace\_if\_lowe null {element} : заменить значение по ключу, если новое значение меньше старого * remove\_greater\_key null : удалить из коллекции все элементы, ключ которых превышает заданный * average\_of\_height : вывести среднее значение поля height для всех элементов коллекции * max\_by\_name : вывести любой объект из коллекции, значение поля name которого является максимальным * filter\_less\_than\_location location : вывести элементы, значение поля location 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 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.util.Date creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически  private Integer height; //Поле может быть null, Значение поля должно быть больше 0  private Color eyeColor; //Поле не может быть null  private Color hairColor; //Поле может быть null  private Country nationality; //Поле может быть null  private Location location; //Поле может быть null  }  public class Coordinates {  private Integer x; //Максимальное значение поля: 749, Поле не может быть null  private long y; //Значение поля должно быть больше -599  }  public class Location {  private float x;  private Float y; //Поле не может быть null  private Long z; //Поле не может быть null  private String name; //Строка не может быть пустой, Поле может быть null  }  public enum Color {  YELLOW,  WHITE,  BROWN;  }  public enum Color {  RED,  BLACK,  WHITE;  }  public enum Country {  USA,  SPAIN,  CHINA,  ITALY;  } |

# Основные этапы решения

*Ссылка на github:*

[https://github.com/mikhail333056/prog\_lab5](https://github.com/Himezon/study/tree/main/Programming/Lab_5_NEW)

*Диаграмма классов (во избежание применения лупы, вы можете открыть диаграмму в формате png по ссылке на репозиторий лабораторной работы выше):*

# Вывод

Выполняя данную работу я на простом примере разобрался с применением коллекций, параметризированными типами, классами оболочками, потоками ввода-вывода в Java. Помимо этого, научился работе с файлами и утилитой javadoc