Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина:

Основы профессиональной деятельности

Лабораторная работа №3

"Управление коллекцией объектов в интерактивном режиме"

Вариант: 311694

Выполнил:

Кузнецов Максим Александрович

Группа: P3111

Преподаватель:

Горбунов М. В.

Санкт-петербург

2020 г.

Задание и условия:

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.PrintWriter`
- 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
- `add {element}` : добавить новый элемент в коллекцию
- `update id {element}` : 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_first` : удалить первый элемент из коллекции
- `add_if_min {element}` : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
- `history` : вывести последние 13 команд (без их аргументов)
- `filter_by_location location` : вывести элементы, значение поля location которых равно заданному
- `filter_contains_name name` : вывести элементы, значение поля name которых содержит заданную подстроку
- `filter_less_than_passport_i_d passportID` : вывести элементы, значение поля passportID 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.

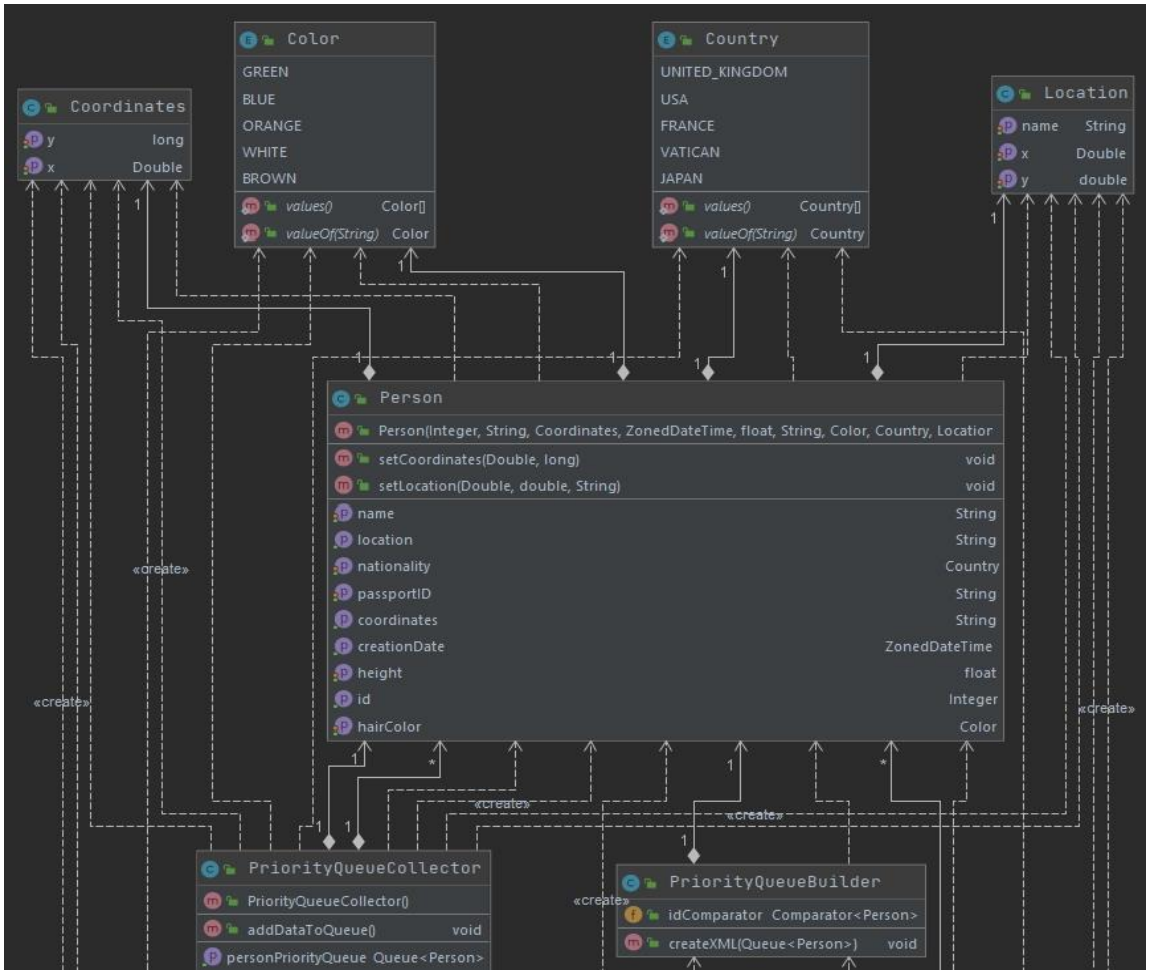
Описание хранимых в коллекции классов:

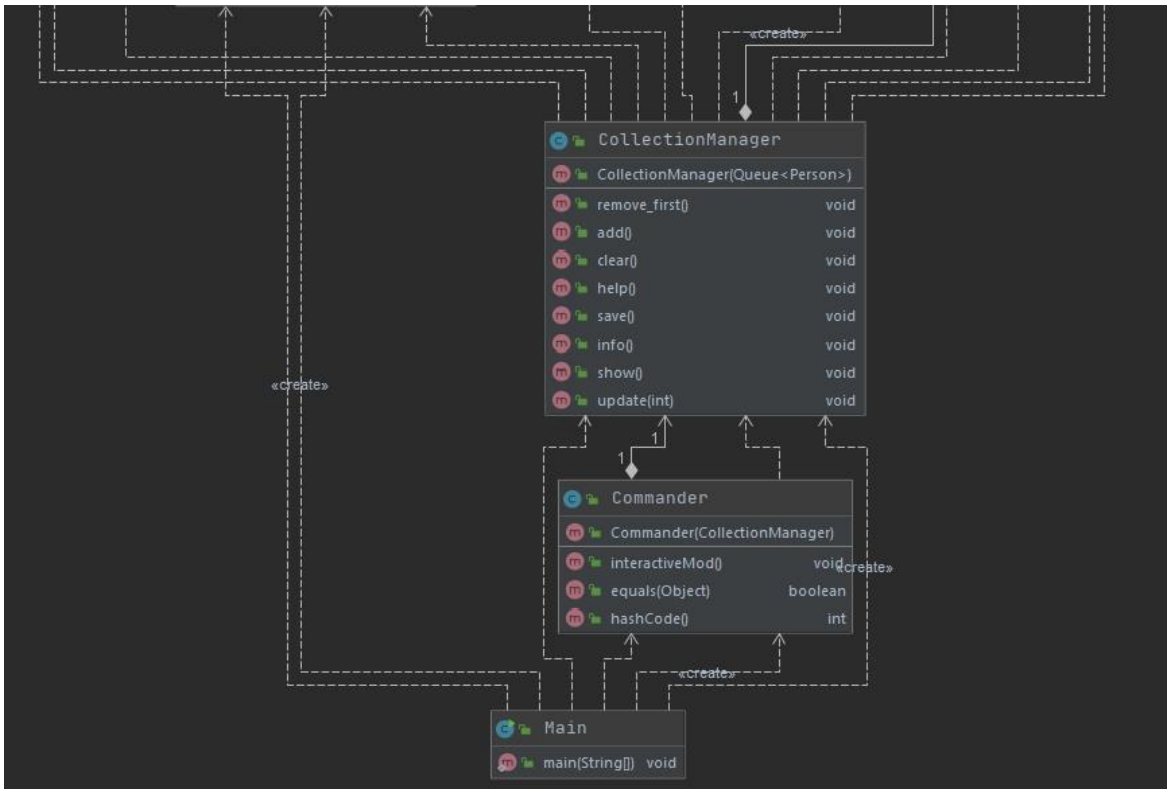
```
public class Person {
    private Integer id; //Поле не может быть null, Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерировать
ся автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.time.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private float height; //Значение поля должно быть больше 0
    private String passportID; //Значение этого поля должно быть уникальным, Поле не может быть null
    private Color hairColor; //Поле не может быть null
    private Country nationality; //Поле не может быть null
    private Location location; //Поле не может быть null
}
public class Coordinates {
    private Double x; //Максимальное значение поля: 203, Поле не может быть null
    private long y;
}
public class Location {
    private Double x; //Поле не может быть null
    private double y;
    private String name; //Строка не может быть пустой, Поле не может быть null
}
public enum Color {
    GREEN,
    BLUE,
    ORANGE,
    WHITE,
    BROWN;
}
public enum Country {
    UNITED_KINGDOM,
    USA,
    FRANCE,
    VATICAN,
    JAPAN;
}
```

Код программы:

Весь код программы находится на моем GitHub'е:
https://github.com/Icerzack/ITMOProgramming/tree/master/Lab5

Примерная UML-диаграмма классов программы:





Выводы по работе:

- Разобрался в работе с коллекциями, в моем случае это былА PrioirtyQueue, немного неподходящий вариант для поставленной задачи, поэтому много где пришлось разбираться с «нереализуемыми» проблемами.
- Научился делать взаимодействие с консолью, а конкретно организовал пользовательский ввод в программу, разобрался с возможными ошибками ввода, исключениями, неверными значениями и так далее.
- Поработал с файлами, а именно: их созданием, чтением из них, а также: парсингом XML-документов с помощью стандартных утилит Джавки