

Задание

Введите вариант: 31222

Внимание! У разных вариантов разный текст задания!

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Flat`, 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: **аргумент командной строки**.
- 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.BufferedOutputStream`
- 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 {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
- `remove_lower {element}` : удалить из коллекции все элементы, меньшие, чем заданный
- `history` : вывести последние 11 команд (без их аргументов)
- `group_counting_by_area` : сгруппировать элементы коллекции по значению поля `area`, вывести количество элементов в каждой группе
- `filter_starts_with_name name` : вывести элементы, значение поля `name` которых начинается с заданной подстроки
- `print_unique_view` : вывести уникальные значения поля `view` всех элементов в коллекции

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 Flat {
    private Long id; //Поле не может быть null, 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.time.ZonedDateTime creationDate; //Поле не может быть null, Значение
этого поля должно генерироваться автоматически
        private double area; //Значение поля должно быть больше 0
        private int numberOfRooms; //Значение поля должно быть больше 0
        private long numberOfBathrooms; //Значение поля должно быть больше 0
        private View view; //Поле может быть null
        private Transport transport; //Поле не может быть null
        private House house; //Поле не может быть null
    }
    public class Coordinates {
        private int x; //Значение поля должно быть больше -232
        private float y; //Значение поля должно быть больше -150
    }
    public class House {
        private String name; //Поле может быть null
        private long year; //Значение поля должно быть больше 0
        private Integer numberOfLifts; //Значение поля должно быть больше 0
    }
    public enum View {
        STREET,
        YARD,
        BAD,
        NORMAL,
        GOOD;
    }
    public enum Transport {
        FEW,
        NONE,
        LITTLE,
        NORMAL,
        ENOUGH;
    }
}

```