

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `StudyGroup`, 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.Stack`
- 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 `csv`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.BufferedReader`
- 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
- `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)
- `add_if_max {element}` : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- `shuffle` : перемешать элементы коллекции в случайном порядке
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `remove_all_by_should_be_expelled shouldBeExpelled` : удалить из коллекции все элементы, значение поля `shouldBeExpelled` которого эквивалентно заданному
- `count_less_than_semester_enum semesterEnum` : вывести количество элементов, значение поля `semesterEnum` которых меньше заданного
- `print_field_descending_average_mark` : вывести значения поля `averageMark` всех элементов в порядке убывания

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 StudyGroup {
```

```

    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.util.Date creationDate; //Поле не может быть null, Значение
    этого поля должно генерироваться автоматически
    private Long studentsCount; //Значение поля должно быть больше 0, Поле
    может быть null
    private Integer shouldBeExpelled; //Значение поля должно быть больше 0,
    Поле не может быть null
    private Long averageMark; //Значение поля должно быть больше 0, Поле не
    может быть null
    private Semester semesterEnum; //Поле не может быть null
    private Person groupAdmin; //Поле может быть null
}
public class Coordinates {
    private Integer x; //Поле не может быть null
    private int y;
}
public class Person {
    private String name; //Поле не может быть null, Строка не может быть
    пустой
    private java.time.LocalDateTime birthday; //Поле не может быть null
    private Color eyeColor; //Поле может быть null
    private Color hairColor; //Поле не может быть null
    private Country nationality; //Поле может быть null
}
public enum Semester {
    FOURTH,
    FIFTH,
    SIXTH,
    SEVENTH,
    EIGHTH;
}
public enum Color {
    RED,
    BLACK,
    BLUE,
    YELLOW;
}
public enum Color {
    RED,
    BLUE,
    WHITE;
}
public enum Country {
    UNITED_KINGDOM,
    GERMANY,
    VATICAN,
    NORTH_KOREA,
    JAPAN;
}

```