Факультет программной инженерной и компьютерной техники

Программирование

Лабораторная работа №5

Вариант 1114

Выполнил: Альхимович Арсений Дмитриевич

P3110

Санкт-Петербург, 2023

Условие

Лабораторная работа #5

Введите вариант: 1114

Внимание! У разных вариантов разный текст задания!

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `SpaceMarine`, 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.Vector`
- 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.util.Scanner`
- 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, 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
- `sort` : отсортировать коллекцию в естественном порядке
- `history` : вывести последние 6 команд (без их аргументов)
- `group_counting_by_chapter` : сгруппировать элементы коллекции по значению поля chapter, вывести количество элементов в каждой группе
- `count_less_than_chapter chapter` : вывести количество элементов, значение поля chapter которых меньше заданного
- `count_greater_than_weapon_type weaponType` : вывести количество элементов, значение поля weaponType которых больше заданного

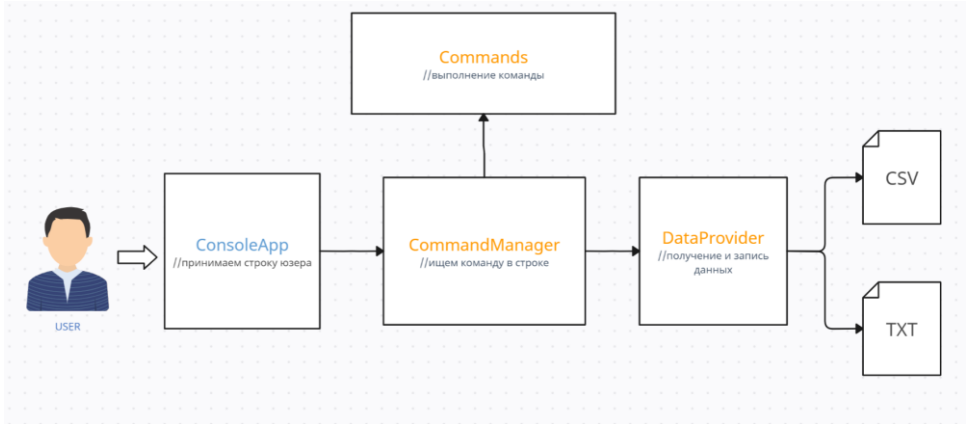
Формат ввода команд:

- 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 SpaceMarine {
    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.time.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Long health; //Поле может быть null, Значение поля должно быть больше 0
    private Long heartCount; //Поле не может быть null, Значение поля должно быть больше 0, Максимальное значение поля: 3
    private Weapon weaponType; //Поле может быть null
    private MeleeWeapon meleeWeapon; //Поле не может быть null
    private Chapter chapter; //Поле может быть null
}
public class Coordinates {
    private int x;
    private Long y; //Поле не может быть null
}
public class Chapter {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private String parentLegion;
    private Integer marinesCount; //Поле может быть null, Значение поля должно быть больше 0, Максимальное значение поля: 1000
}
public enum Weapon {
    PLASMA_GUN,
    GRENADE_LAUNCHER,
    INFERNO_PISTOL,
    MULTI_MELTA;
}
public enum MeleeWeapon {
    POWER_SWORD,
    MANREAPER,
    POWER_FIST;
}
```

Диаграмма классов



Исходный код программы

https://github.com/senya-2011/itmo-proga/tree/main/lab5

Вывод

Составление UML диаграммы классов перед началом самого процесса писания кода очень упрощает и помогает в последующих действиях. Удобно сохранять данные в файл, тем самым сохраняя данные даже после перезапуска программы. Мапа удобна при группировке, тк элементы имеют уникальный ключ. Столкнулся с ошибкой кодировки при сборке, важно смотреть, что проект на правильной.