

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Направление системного и прикладного программного обеспечения

Отсчет по лабораторной работе № 5
Дисциплина «Программирование»
Вариант № 4024

Выполнила:
Исаева Александра-Ирина Антоновна
Группа: Р3109
Преподаватели:
Гаврилов Антон Валерьевич
Мустафаева Айнур Вугар кызы

Содержание

Задание	3
UML диаграмма команд.....	5
Исходный код	5
Выводы.....	5

Задание

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса **HumanBeing**, 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.LinkedHashSet**
- 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.util.Scanner**
- 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)
- **add_if_max {element}** : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- **remove_lower {element}** : удалить из коллекции все элементы, меньшие, чем заданный
- **history** : вывести последние 8 команд (без их аргументов)
- **filter_greater_than_weapon_type weaponType** : вывести элементы, значение поля **weaponType** которых больше заданного
- **print_descending** : вывести элементы коллекции в порядке убывания
- **print_field_ascending_mood** : вывести значения поля **mood** всех элементов в порядке возрастания

Формат ввода команд:

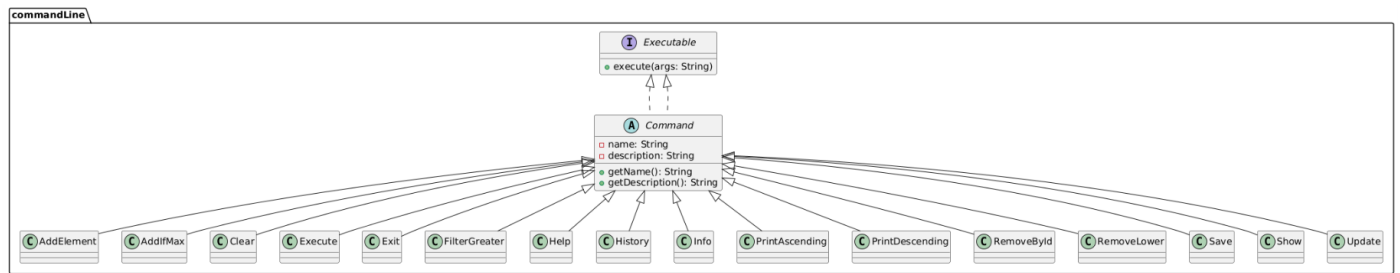
- Все аргументы команды, являющиеся стандартными типами данных (примитивные типы, классы-оболочки, **String**, классы для хранения дат), должны вводиться в той же строке, что и имя команды.

- 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'e; 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 HumanBeing {
    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 boolean realHero;
    private Boolean hasToothpick; //Поле не может быть null
    private float impactSpeed;
    private WeaponType weaponType; //Поле может быть null
    private Mood mood; //Поле не может быть null
    private Car car; //Поле может быть null
}
public class Coordinates {
    private Float x; //Поле не может быть null
    private float y;
}
public class Car {
    private String name; //Поле может быть null
}
public enum WeaponType {
    HAMMER,
    SHOTGUN,
    RIFLE,
    KNIFE,
    BAT;
}
public enum Mood {
    SADNESS,
    SORROW,
    CALM,
    RAGE,
    FRENZY;
}
```

UML диаграмма команд



Исходный код

https://github.com/isa-alex/Prog_lab5

Выводы

В процессе выполнения лабораторной работы я познакомилась с организацией коллекций, их сортировки, работы с файлами и работы с консолью. Узнала про различные потоки ввода-вывода, утилиту Javadoc, классы оболочки.