

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №5

Вариант №3220

Группа: Р3132

Выполнил: Старикова Д. Д.

Преподаватель:
Усков Иван Владимирович

Санкт-Петербург
2022 г.

Задание:

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.Hashtable`
- 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.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
- `insert null {element}` : добавить новый элемент с заданным ключом
- `update id {element}` : обновить значение элемента коллекции, id которого равен заданному
- `remove_key null` : удалить элемент из коллекции по его ключу
- `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_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `replace_if_lowe null {element}` : заменить значение по ключу, если новое значение меньше старого
- `remove_lower_key null` : удалить из коллекции все элементы, ключ которых меньше, чем заданный
- `remove_any_by_weapon_type weaponType` : удалить из коллекции один элемент, значение поля `weaponType` которого эквивалентно заданному

- **average_of_health** : вывести среднее значение поля health для всех элементов коллекции
- **count_by_chapter chapter** : вывести количество элементов, значение поля chapter которых равно заданному

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 SpaceMarine {
    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 double health; //Значение поля должно быть больше 0
    private AstartesCategory category; //Поле не может быть null
    private Weapon weaponType; //Поле не может быть null
    private MeleeWeapon meleeWeapon; //Поле может быть null
    private Chapter chapter; //Поле не может быть null
}
public class Coordinates {
    private Double x; //Значение поля должно быть больше -685, Поле не может быть
null
    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 world; //Поле может быть null
}
public enum AstartesCategory {
    SCOUT,
    INCEPTOR,
    TACTICAL,
    CHAPLAIN;
}
public enum Weapon {
    HEAVY_BOLTGUN,
```

```
BOLT_RIFLE,  
PLASMA_GUN,  
INFERNO_PISTOL;  
}  
public enum MeleeWeapon {  
    CHAIN_AXE,  
    MANREAPER,  
    LIGHTING_CLAW,  
    POWER_BLADE,  
    POWER_FIST;  
}
```

Исходный код программы можно найти по ссылке <https://github.com/bhmaa/5-lab>

Выводы по работе:

Во время работы я познакомилась с основными реализациями коллекций в java, принципами их работы, их различиями между собой. Впервые осуществляла работу с файлами на этом языке, познакомилась с буферизованными потоками. Также для парсинга я использовала JAXB, который оказался очень простым и удобным в использовании.