Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Программирование

**Лабораторная работа № 5**

Вариант № 3113014

Выполнил:

Сандов Кирилл Алекссевич

Группа:

P3113

Проверил:

преподаватель практики Письмак Алексей Евгеньевич

Санкт-Петербург

2023

# Оглавление

[Оглавление 2](#_Toc119848939)

[Задание 3](#_Toc119848940)

[Диаграмма классов модели 5](#_Toc119848941)

[Исходный код программы 6](#_Toc119848942)

[Результат работы программы 7](#_Toc119848943)

[Заключение 10](#_Toc119848944)

# 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.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 json
* Чтение данных из файла необходимо реализовать с помощью класса java.io.BufferedReader
* 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, отсутсвие 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\_last : удалить последний элемент из коллекции
* shuffle : перемешать элементы коллекции в случайном порядке
* remove\_all\_by\_health health : удалить из коллекции все элементы, значение поля health которого эквивалентно заданному
* filter\_starts\_with\_name name : вывести элементы, значение поля name которых начинается с заданной подстроки
* print\_field\_descending\_chapter : вывести значения поля chapter 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; 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.util.Date creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private Long health; //Поле не может быть null, Значение поля должно быть больше 0

private AstartesCategory category; //Поле не может быть null

private Weapon weaponType; //Поле может быть null

private MeleeWeapon meleeWeapon; //Поле может быть null

private Chapter chapter; //Поле не может быть null

}

public class Coordinates {

private double x;

private Float y; //Значение поля должно быть больше -273, 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 {

SUPPRESSOR,

TACTICAL,

TERMINATOR,

APOTHECARY;

}

public enum Weapon {

MELTAGUN,

GRENADE\_LAUNCHER,

MISSILE\_LAUNCHER;

}

public enum MeleeWeapon {

CHAIN\_SWORD,

CHAIN\_AXE,

POWER\_BLADE;

}

# Диаграмма классов модели

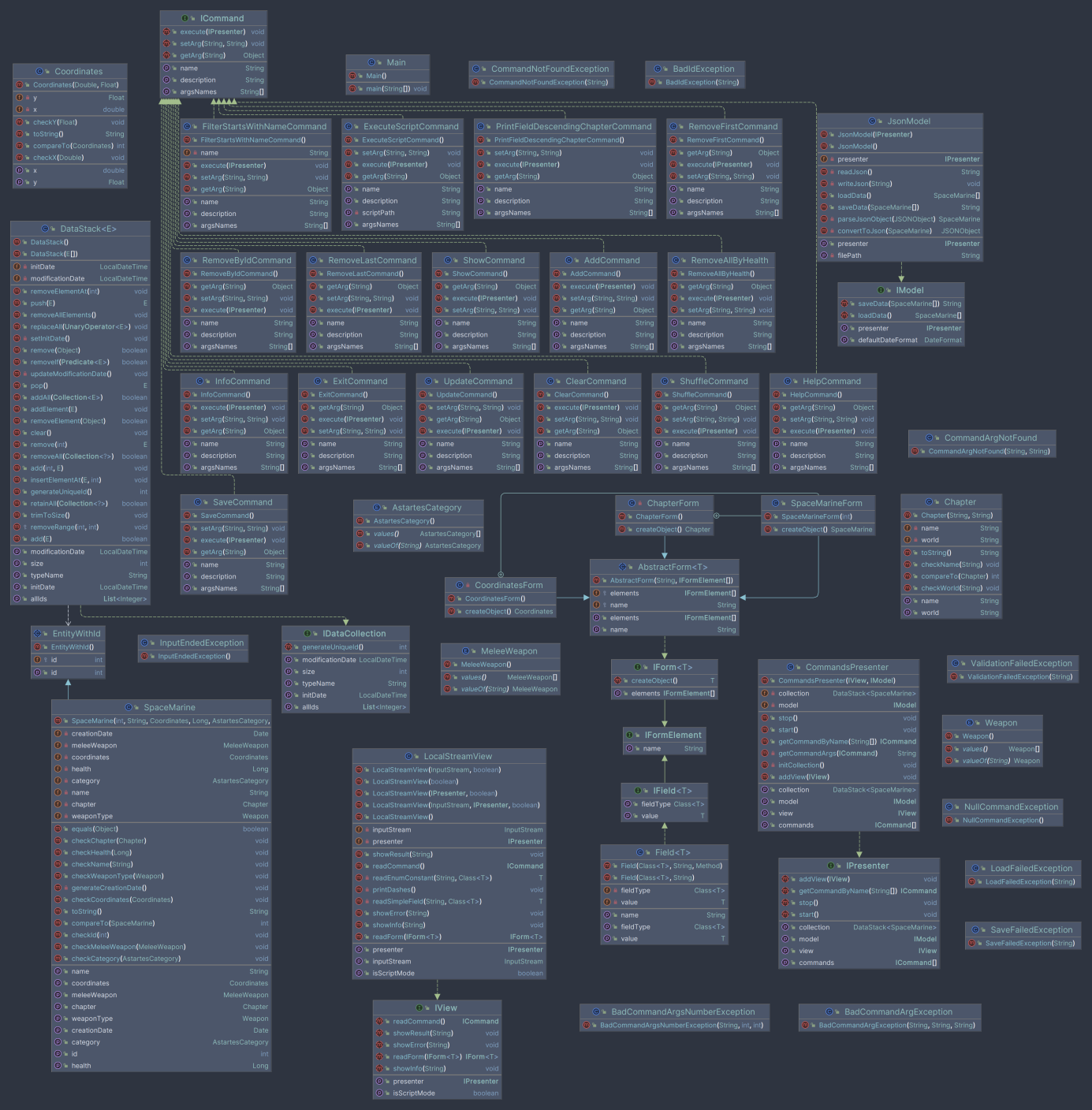


Рисунок 1 – «UML-диаграмма классов построенной объектной модели»

# Исходный код программы

Ссылка на GitHub-репозиторий с реализацией:

<https://github.com/amphyxs/vt-labas/tree/main/sem-2/prog/lab-5>

# **Заключение**

В результате выполнения данной лабораторной работы изучены методы работы с файла и потоками ввода-вывода. Изучены различные паттерны проектирования, например, такие как Command и Builder. Также получены знания о стандартных коллекциях в Java, изучена их сортировка. Применены параметры типов и подстановки. В итоге получилось полноценное консольное приложение.