Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Вариант №4828

Лабораторная работа №5

По дисциплине

Программирование

Выполнил студент группы P3106:

Некрутенко Максим

Преподаватель:

Вербовой Александр Александрович

**Санкт-Петербург 2025 г.**

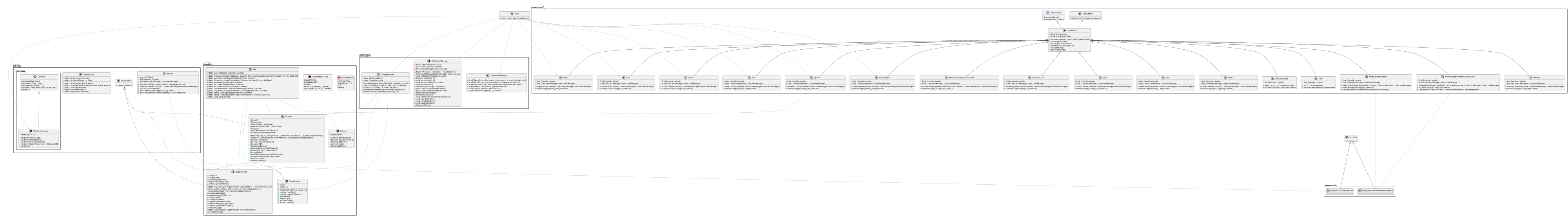
1. **Текст задания**

|  |
| --- |
| Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Product, 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: 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.BufferedOutputStream * 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 * sort : отсортировать коллекцию в естественном порядке * history : вывести последние 5 команд (без их аргументов) * remove\_any\_by\_manufacturer manufacturer : удалить из коллекции один элемент, значение поля manufacturer которого эквивалентно заданному * filter\_contains\_name name : вывести элементы, значение поля name которых содержит заданную подстроку * filter\_greater\_than\_unit\_of\_measure unitOfMeasure : вывести элементы, значение поля unitOfMeasure 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 Product {  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.time.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически  private int price; //Значение поля должно быть больше 0  private UnitOfMeasure unitOfMeasure; //Поле может быть null  private Organization manufacturer; //Поле не может быть null  }  public class Coordinates {  private long x;  private Integer y; //Поле не может быть null  }  public class Organization {  private Integer 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 int employeesCount; //Значение поля должно быть больше 0  private OrganizationType type; //Поле может быть null  private Address postalAddress; //Поле не может быть null  }  public class Address {  private String street; //Поле может быть null  }  public enum UnitOfMeasure {  CENTIMETERS,  SQUARE\_METERS,  PCS,  GRAMS;  }  public enum OrganizationType {  COMMERCIAL,  GOVERNMENT,  TRUST,  PRIVATE\_LIMITED\_COMPANY,  OPEN\_JOINT\_STOCK\_COMPANY;  } |

1. **Исходный код программы.**

Репозиторий: https://github.com/maksim-06/Study-in-ITMO/tree/proga

1. **Диаграмма классов реализованной объектной модели.**



1. **Вывод**

В процессе выполнения лабораторной работы я научился работать с различными структурами данных в Java и файлами, а также углубил свои знания в ООП в Java. Более глубоко понял интерфейсы java.util.Comparable и java.util.Comparator,пакет java.nio. Также изучил параметризованные типы, wildcard-параметры и утилиту javadoc.