Warehouse Management Project Documentation

By Szymon Janiak

Table of Contents

Introduction

The Warehouse Management project aims to provide an efficient way to manage products in a warehouse. The application allows users to add, sort, and clone products. It implements the Prototype design pattern to clone objects seamlessly.

Project Structure

src/

├── main/

│ ├── Main.java

├── dao/

│ ├── DatabaseConnector.java

│ ├── ProductDao.java

├── exceptions/

│ ├── ProductNotFoundException.java

├── model/

│ ├── Product.java

│ ├── Category.java

├── prototype/

│ ├── CloneableProduct.java

├── utils/

│ ├── ProductQuantityComparator.java

Main Classes and Interfaces

Main- The main class that launches the application. It initializes the database connection, adds products, sorts them by quantity, and clones a selected product.

public class Main { public static void main(String[] args) { // Logic to run the application} }

DatabaseConnector- A class responsible for managing the connection to the MySQL database. It provides the getConnection() method to establish a connection.

public class DatabaseConnector { // Method to get a connection to the database}

ProductDao - A Data Access Object (DAO) class that manages product operations in the database. It uses a HashMap to store products in memory and provides methods for adding, removing, and retrieving products.

public class ProductDao { // Methods for managing products in the database}

Product- A class that represents a product in the warehouse. It has properties like name, price, quantity, and category. The class implements the CloneableProduct interface to enable cloning.

public class Product implements CloneableProduct // Properties and methods of the Product class}

Category Enum- An enum representing product categories, such as ELECTRONICS, GROCERIES, TOYS i OTHERS.

public enum Category { ELECTRONICS, GROCERIES, TOYS, OTHERS }

CloneableProduct- An interface that defines the clone() method, allowing objects to be cloned.

public interface CloneableProduct { Product clone(); // Method for cloning objects}

ProductQuantityComparator- A comparator class that implements the Comparator interface and enables sorting products by their quantity in the warehouse.

public class ProductQuantityComparator implements Comparator<Product> { // Implementation of the compare method }

Database

The project requires a MySQL database named warehouse. The application checks if the database exists, and if not, it creates it automatically. The database contains a products table with columns corresponding to the properties of the products.

Prototype Pattern

The project employs the Prototype design pattern, allowing objects to be cloned without the need to create them from scratch. The Product class implements the CloneableProduct interface, making it easy to create copies of objects.

Technologies Used

-Java - Programming language used for the project.

-MySQL - Database management system.

-JDBC API for connecting to the database.

How to Run the Project

1.Ensure that MySQL is installed and running on your system.

2.Create a database named warehouse.

3.Configure the connection settings in the DatabaseConnector class.

4.Run the Main class to initialize the application and test its functionalities.