**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG THÀNH PHỐ HỒ CHÍ MINH**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

**ĐỒ ÁN**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI QUẢN LÝ KHO QUẦN ÁO**

NHÓM 1

LÊ NGUYỄN HÀ MY

ĐẶNG TRẦN MINH QUÂN

VÕ TẤN KIỆT

LỚP : CNTT5

GIÁO VIÊN HƯỚNG DẪN : PHẠM TRỌNG HUYNH

**MỤC LỤC**

[**I. Sơ lược về dự án 2**](#_lqnobd10axgm)

[**II. Dùng maven để quản lý phụ thuộc của dự án 4**](#_2tmxi95e32oe)

[**III. Xác định các lớp của bài toán 6**](#_nuq8t9p5qatp)

[1. Class ChungTu 6](#_rbt4dar1jvfo)

[2. Class HangHoa 6](#_19z4wb8xulzw)

[3. Class NhaCC 6](#_e5lijm7iam1)

[4. Class XuatXu 7](#_vql3uqgse2j7)

[5. Class DeleteChungTu 7](#_ap8itwgzhuef)

[6. Class DeleteHangHoa 7](#_mi3hu9gitwkc)

[7. Class DeleteNhaCC 7](#_u5tdhkmkeras)

[8. Class DeleteXuatXu 8](#_hjazv4kl0e87)

[9. Class InsertChungTu 8](#_hqexoxaq1k5v)

[11. Class InsertNhaCC 8](#_iphukl5bmorp)

[12. Class InsertXuatXu 9](#_b9avfk1b619y)

[13. Class ReadChungTu 9](#_2wgfqmcef68y)

[14. Class ReadHangHoa 9](#_s23i91y1qhtg)

[15. Class ReadNhaCC 10](#_6axpszmd5v91)

[16. Class ReadXuatXu 10](#_qwl4ndwnlvub)

[17. Class UpdateChungTu 10](#_qewfvpo6otlg)

[18. Class UpdateHangHoa 10](#_n58q53zchhjx)

[19. Class UpdateNhaCC 11](#_9epi18fn7dqy)

[20. Class UpdateXuatXu 11](#_4szibk8nr5y1)

[21. Class Connecting 11](#_s2dts43zwjiw)

[22. Class ClassName 12](#_slecxeiiy0o6)

[23. Class CreateTable 12](#_r99n9g2p8cpu)

[24. Class DataFormat 12](#_y8eep819akat)

[25. Class ManipulationFile 13](#_k33tokeziszm)

[26. Class UserInput 13](#_xv333qxv4ztk)

[27. Class ClearConsole 13](#_hmywvpk05z5j)

[28. Class MenuChungTu 13](#_ghbyu48wx7v7)

[29. Class MenuHangHoa 13](#_b8blizf3b0ws)

[30. Class MenuNhaCC 13](#_2im0s1pvmbuh)

[31. Class MenuXuatXu 14](#_jep19irfeta3)

[32. Class MenuMain 14](#_dbxnzfs9snk5)

[33. Class ShowChungTu 14](#_59evilqcdyfh)

[34. Class ShowHangHoa 14](#_ve2x76nnz02y)

[35. Class ShowNhaCC 14](#_4f726qaiwpep)

[36. Class ShowXuatXu 14](#_q5twimpt2hzv)

[**IV. Cài đặt bài toán 14**](#_dkz98wtnr9ri)

[1. Class ChungTu 14](#_zf09cjvlwukj)

[2. Class HangHoa 15](#_jzilxborzkty)

[3. Class NhaCC 16](#_1nkd64xvv8mp)

[4. Class XuatXu 17](#_lrqvong6s7vz)

[5. Class DeleteChungTu 17](#_afd2grdtntdz)

[6. Class DeleteHangHoa 19](#_tk1uvcu8pfkv)

[7. Class DeleteNhaCC 21](#_qciykd262big)

[8. Class DeleteXuatXu 22](#_vpy6un3i5hgy)

[9. Class InsertChungTu 24](#_7w3xenyoh1a9)

[10. Class InsertHangHoa 25](#_z8sdcivw3k12)

[11. Class InsertNhaCC 27](#_fe1ylo2tgipv)

[12. Class InsertXuatXu 29](#_ous2zvf758xy)

[13. Class ReadChungTu 30](#_vzzuj64vue6q)

[14. Class ReadHangHoa 33](#_tyua8dfk5xur)

[15. Class ReadNhaCC 37](#_1oqkime8zjh1)

[16. Class ReadXuatXu 40](#_z2o8fo79vtpf)

[17. Class UpdateChungTu 44](#_pqkzehah5van)

[18. Class UpdateHangHoa 45](#_w5rg7yshqt11)

[19. Class UpdateNhaCC 47](#_f472m6bj0lli)

[20. Class UpdateXuatXu 49](#_1yvzez3tfku)

[21. Class Connecting 51](#_k313jweteuwe)

[22. Class ClassName 52](#_1nihcjs6e1sr)

[23. Class CreateTable 53](#_po5swpuk0ca)

[24. Class DataFormat 55](#_82r3sw78zzmn)

[25. Class ManipulationFile 56](#_r6z341k4o4gz)

[26. Class UserInput 59](#_4r2pu8zceaub)

[27. Class ClearConsole 60](#_yjikzl7c2mqp)

[28. Class MenuChungTu 60](#_1yna40cleb2f)

[29. Class MenuHangHoa 70](#_jc12ijfmwbqm)

[30. Class MenuNhaCC 79](#_5xcnkbynevdc)

[31. Class MenuXuatXu 89](#_u60lhbptceyz)

[32. Class MenuMain 98](#_bn09zboc3kqr)

[33. Class ShowChungTu 100](#_bnllgdn0mc2u)

[34. Class ShowHangHoa 101](#_z2sccr3qnh4c)

[35. Class ShowNhaCC 101](#_fail3n2ux0ep)

[36. Class ShowXuatXu 102](#_ar3uu5lsclxx)

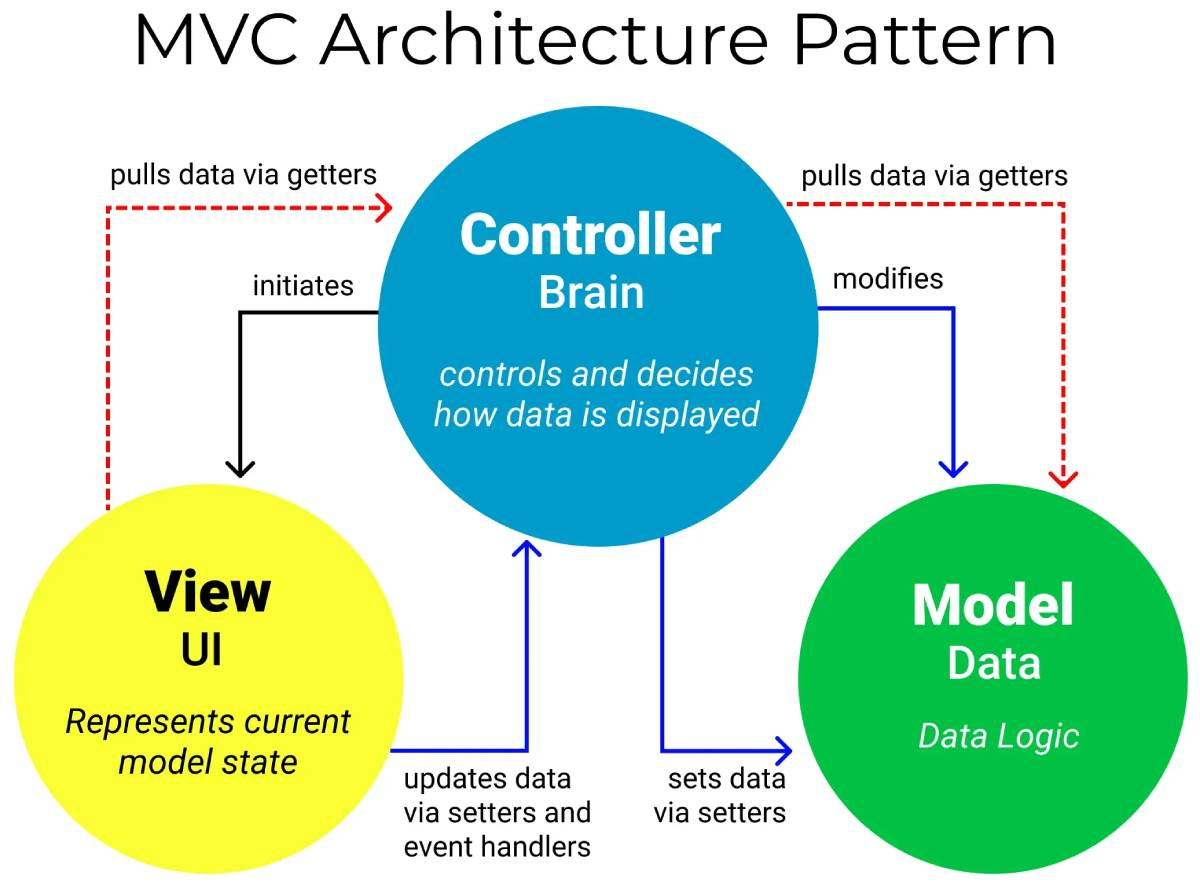
[37. Class App 102](#_7bncyi8igh89)

# Sơ lược về dự án

**Mục tiêu**: tạo một chương trình java có chức năng quản lý kho quần áo

**Công cụ sử dụng**: Maven, Microsoft SQL Server

Dự án được thiết kế theo mô hình MVC



Cấu trúc thư mục dự án bao gồm:

App.java: lớp này chứa hàm main dùng để chạy chương trình

**Thư mục Controller**: Chứa các lớp có chức năng thêm xóa sửa đọc file của các bảng bao gồm

* DeleteHangHoa.java
* InsertHangHoa.java
* ReadHangHoa.java
* UpdateHangHoa.java
* DeleteChungTu.java
* InsertChungTu.java
* ReadChungTu.java
* UpdateChungTu.java
* DeleteNhaCC.java
* InsertNhaCC.java
* ReadNhaCC.java
* Update.NhaCC.java
* DeleteXuatXu.java
* InsertXuatXu.java
* ReadXuatXu.java
* UpdateXuatXu.java

**Thư mục DatabaseConnection**: Thư mục chứa các lớp dùng để thiết lập kết nối đến database bao gồm

* Connecting.java

**Thư mục Helper:** Chứa các lớp hỗ trợ trong việc xây dựng các chức năng của dự án ví dụ như nhập dữ liệu người dùng, chuẩn hóa chuỗi, định dạng đầu ra ….

* ClassName.java
* CreateTable.java
* DataFormat.java
* ManipulationFile.java
* UserInput.java

**Thư mục Model:** nơi này chứa các lớp mô hình dùng để mô hình hóa dữ liệu trong database bao gồm các lớp

* ChungTu.java
* HangHoa.java
* NhaCC.java
* XuatXu.java

**Thư mục Stored:** nơi này lưu trữ các file txt chứa dữ liệu mà người dùng yêu cầu chương trình tạo ra

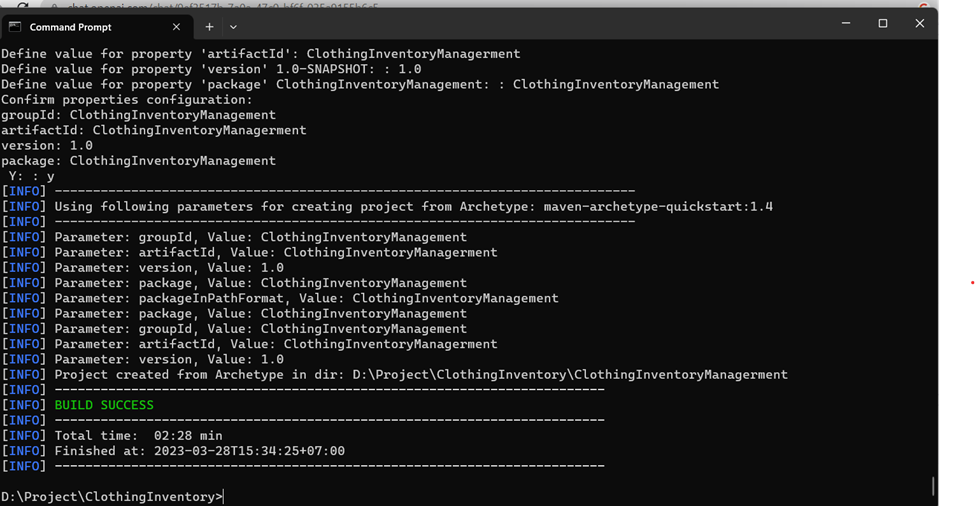
**Thư mục View:** nơi này chứa các class có chức năng tạo ra giao diện người dùng như menu, bảng dữ liệu

* ClearConsole.java
* MenuChungTu.java
* MenuHangHoa.java
* MenuNhaCC.java
* MenuXuatXu.java
* ShowChungTu.java
* ShowHangHoa.java
* ShowNhaCC.java
* ShowXuatXu.java

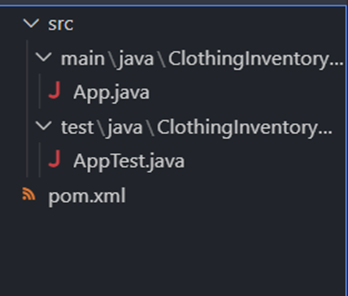
Ngoài ra dự án này có sử dụng thêm thư viện bên ngoài dùng để hỗ trợ việc tạo bảng (thư viện asciitable link github: <https://github.com/vdmeer/asciitable>) các thư viện bên ngoài được cài đặt bằng công cụ quản lý dự án MAVEN.

# Dùng maven để quản lý phụ thuộc của dự án

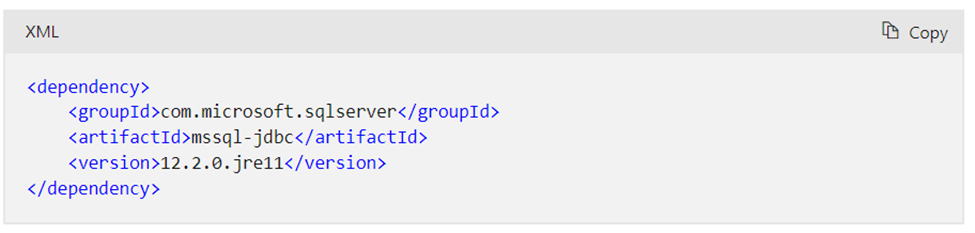
Do dự án có sử dụng database nên phải sử dụng driver cơ sở dữ liệu để thực hiện các truy vấn trên cơ sở dữ liệu. Ở đây nhóm chúng em sử dụng công cụ quản lý dự án Maven.



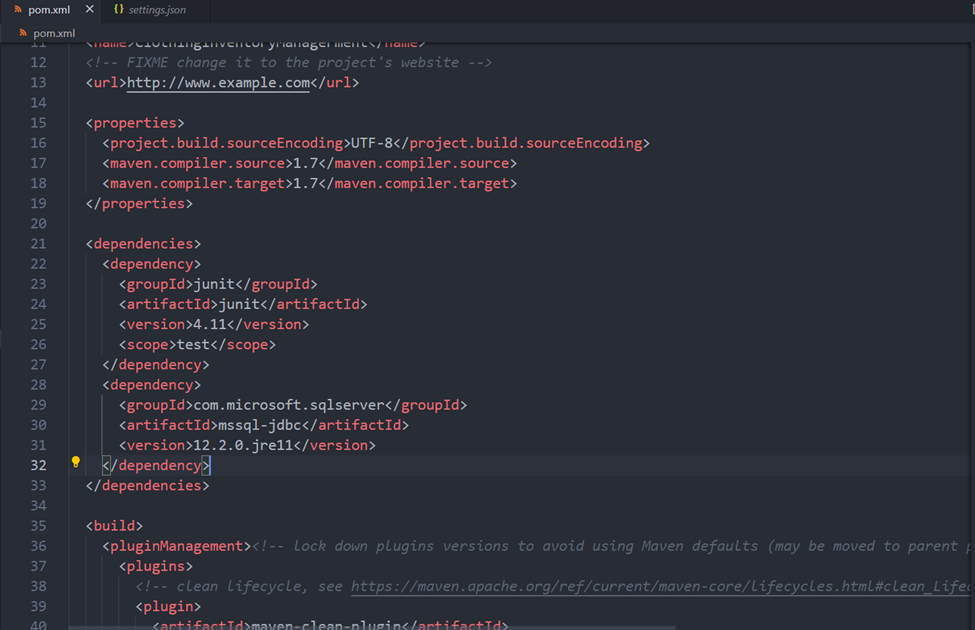
Sau khi config xong cái thuộc tính như tên dự án, packet chúng ta có một cấu trúc thư mục dự án như sau.



Trong đó có file pom.xml nơi mà chúng ta cài các driver cơ sở dữ liệu.



Chúng ta copy đoạn mã này vào thẻ dependencies trong file pom.xml



Sau đó gõ lệnh

mvn clean install

# Xác định các lớp của bài toán

## Class ChungTu

Thuộc tính: int soLuong, donGia, String maHang, maNCC, maXX

Constructor: hàm tạo có đối số và hàm tạo không có đối số

## Class HangHoa

Thuộc tính: String maHang, tenHang, mauSac, kichCo, xuatXu, int donGia

Constructor: Hàm tạo có đối số và hàm tạo không có đối số

## Class NhaCC

Thuộc tính: String maNCC, tenNCC, std, email

Constructor: Hàm tạo có đối số và hàm tạo không có đối số

## Class XuatXu

Thuộc tính: String ten, maXX

Constructor: Hàm tạo có đối sô và hàm tạo không có đối số

## Class DeleteChungTu

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức delete(ChungTu data) thực hiện công việc xóa dữ liệu trên database

delete(ArrayList<ChungTu> list) là đa hình của phương thức delete bên trên thực hiện việc xóa nhiều dữ liệu

## Class DeleteHangHoa

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức delete(String id) thực hiện công việc xóa dữ liệu trên database

delete(ArrayList<String> list) thực hiện việc xóa nhiều dữ liệu

## Class DeleteNhaCC

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức delete(String id) thực hiện công việc xóa dữ liệu trên database

delete(ArrayList<String> list) thực hiện việc xóa nhiều dữ liệu

## Class DeleteXuatXu

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức delete(String id) thực hiện công việc xóa dữ liệu trên database

delete(ArrayList<String> list) thực hiện việc xóa nhiều dữ liệu

## Class InsertChungTu

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: insert(Chungtu data) thực hiện việc nhập dữ liệu vào database

insert(ArrayList<ChungTu> list) thực hiện việc nhập nhiều dữ liệu

1. **Class InsertHangHoa**

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: insert(HangHoa data) thực hiện việc nhập dữ liệu vào database

insert(ArrayList<HangHoa> list) thực hiện việc nhập nhiều dữ liệu

## Class InsertNhaCC

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: insert(NhaCC data) thực hiện việc nhập dữ liệu vào database

insert(ArrayList<NhaCC> list) thực hiện việc nhập nhiều dữ liệu

## Class InsertXuatXu

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: insert(XuatXu data) thực hiện việc nhập dữ liệu vào database

insert(ArrayList<XuatXu> list) thực hiện việc nhập nhiều dữ liệu

## Class ReadChungTu

Phương thức tĩnh: readFromDataBase(Connection connection, String extend) thực hiện nhiệm vụ đọc dữ liệu từ database và trả về cho người dùng

readFromFile(String fileName) thực hiện nhiệm vụ đọc dữ liệu từ file theo định dạng đã được quy định trước

readFromKeyboad() thực hiện nhiệm vụ đọc dữ liệu người dùng từ bàn phím

## Class ReadHangHoa

Phương thức tĩnh: readFromDataBase(Connection connection, String extend) thực hiện nhiệm vụ đọc dữ liệu từ database và trả về cho người dùng

readFromFile(String fileName) thực hiện nhiệm vụ đọc dữ liệu từ file theo định dạng đã được quy định trước

readFromKeyboad() thực hiện nhiệm vụ đọc dữ liệu người dùng từ bàn phím

## Class ReadNhaCC

Phương thức tĩnh: readFromDataBase(Connection connection, String extend) thực hiện nhiệm vụ đọc dữ liệu từ database và trả về cho người dùng

readFromFile(String fileName) thực hiện nhiệm vụ đọc dữ liệu từ file theo định dạng đã được quy định trước

readFromKeyboad() thực hiện nhiệm vụ đọc dữ liệu người dùng từ bàn phím

## Class ReadXuatXu

Phương thức tĩnh: readFromDataBase(Connection connection, String extend) thực hiện nhiệm vụ đọc dữ liệu từ database và trả về cho người dùng

readFromFile(String fileName) thực hiện nhiệm vụ đọc dữ liệu từ file theo định dạng đã được quy định trước

readFromKeyboad() thực hiện nhiệm vụ đọc dữ liệu người dùng từ bàn phím

## Class UpdateChungTu

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: update(Chung tu data) thực hiện việc cập nhật dữ liệu lên database

update(ArrayList<ChungTu> list) thực hiên việc cập nhật nhiều dữ liệu lên database

## Class UpdateHangHoa

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: update(HangHoa data) thực hiện việc cập nhật dữ liệu lên database

update(ArrayList<HangHoa> list) thực hiên việc cập nhật nhiều dữ liệu lên database

## Class UpdateNhaCC

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: update(NhaCC data) thực hiện việc cập nhật dữ liệu lên database

update(ArrayList<NhaCC> list) thực hiên việc cập nhật nhiều dữ liệu lên database

## Class UpdateXuatXu

Thuộc tính: Connection connection; có setter để thay đổi kết nối nếu muốn

Constructor: Hàm tạo có tham số

Phương thức: update(Chung tu data) thực hiện việc cập nhật dữ liệu lên database

update(ArrayList<ChungTu> list) thực hiên việc cập nhật nhiều dữ liệu lên database

## Class Connecting

Thuộc tính: Connection connection có getter dùng để lấy connection nếu kết nối thất bại thì connection là null. Nếu kết nối thành công in ra thông báo “Kết nối thành công”

Constructor: Phương thức khởi tạo có đối số đồng thời thực hiện thao tác kết nối vào cơ sở dữ liệu

## Class ClassName

Mô tả: lớp này là lớp chức năng cung cấp các công cụ dùng để lấy tên class từ một generic cũng như tên các thuộc tính của nó tạo thuận lợi cho việc xây dựng các class chức năng khác.

Phương thức tĩnh: getNameWithFileName(String fileName) phương thức này dùng để lấy tên class từ tên file để tạo thuận lợi cho việc đọc dữ liệu từ file

Ví dụ: file ChungTu23551235.txt thì ta biết rằng file này chứa dữ liệu của bảng ChungTu nên ta có thể dùng chính tên lớp đó tạo ra 1 instance và chỉnh sửa các thuộc tính của instance đó từ đó chúng ta có thể dễ dàng xây dựng các class chức năng hữu ích như đọc file. Việc sử dụng các tên class sẽ thực sự hữu dụng khi xây dựng các class trong Controller sử dụng nhiều mô hình dữ liệu khác nhau.

getClassPropertyName(Object obj) phương thức này dùng để lấy tên các thuộc tính của một class với obj là mẫu dùng để lấy tên thuộc tính

Ví dụ ChungTu sample = new ChungTu(); là một mẫu dùng để lấy tên thuộc tính của class Chung tu

Việc lấy tên các thuộc tính sẽ thực sự hữu dụng khi xây dựng các chức năng như tạo bảng, tạo menu yêu cầu người dùng nhập thuộc tính của bảng .v.v

getNameWithClass(Object obj) phương thức này dùng để lấy tên class từ mẫu (obj) công dụng đã giải thích ở bên trên

## Class CreateTable

Phương thức tĩnh: createTable(ArrayList<T> array) với T là kiểu generic ở đây là class model không biết trước như là NhaCC, HangHoa, ChungTu,... Có chức năng tạo một bảng để trực quan hóa dữ liệu

## Class DataFormat

Phương thức tĩnh: Format(ArrayList<T> array) có chức năng định dạng dữ liệu đầu ra (mỗi trường dữ liệu cách bởi 1 tab, mỗi instance cách bởi một kí tự xuống dòng)

NormalizeString(String input) dùng để chuẩn hóa dữ liệu ở đây là bỏ các dấu cách thừa ở cuối mỗi chuỗi

## Class ManipulationFile

Mô tả: Class này có chức năng tạo và đọc file

Phương thức tĩnh: reader(String fileName) thực hiện việc đọc file với tên file có thể được người dùng nhập

writer(String fileName, String data) thực hiện việc đọc file với data là chuỗi dữ liệu (chuỗi dữ liệu này nên được định dạng trước dùng phương thức định dạng như đã mô tả ở trên)

## Class UserInput

Phương thức tĩnh: InputIntoInstance(T sample) có chức năng yêu cầu người dùng nhập dữ liệu và trả về một instance của một class model cụ thể Ví dụ: ChungTu, HangHoa, …

## Class ClearConsole

Phương thức tĩnh: clear() in ra các kí tự xuống hàng để trôi đi (ẩn đi) các giao diện đã sử dụng trong quá trình chạy chương trình. Nhằm mục đích tạo ra trải nghiệm người dùng tốt hơn

## Class MenuChungTu

Phương thức tĩnh generate(Connection connection) tạo ra menu thao tác trên bảng ChungTu

## Class MenuHangHoa

Phương thức tĩnh generate(Connection connection) tạo ra menu thao tác trên bảng HangHoa

## Class MenuNhaCC

Phương thức tĩnh generate(Connection connection) tạo ra menu thao tác trên bảng NhaCC

## Class MenuXuatXu

Phương thức tĩnh generate(Connection connection) tạo ra menu thao tác trên bảng XuatXu

## Class MenuMain

Phương thức tĩnh generate(Connection connection) tạo ra menu chính

## Class ShowChungTu

Phương thức tĩnh: view(ArrayList<ChungTu> data) tạo bảng cho model Chung Tu

## Class ShowHangHoa

Phương thức tĩnh: view(ArrayList<HangHoa> data) tạo bảng cho model HangHoa

## Class ShowNhaCC

Phương thức tĩnh: view(ArrayList<NhaCC> data) tạo bảng cho model NhaCC

## Class ShowXuatXu

Phương thức tĩnh: view(ArrayList<XuatXu> data) tạo bảng cho model XuatXu

# IV. Cài đặt bài toán

**link github**

**https://github.com/Votankiet/CodeDoAnQL\_KhoQuanAo\_Nhom1.git**

**1. Class ChungTu**

package ClothingInventoryManagement.Model;

public class ChungTu {

public int soLuong;

public int donGia;

public String maHang;

public String maNCC;

public String maXX;

public ChungTu(int soLuong, int donGia, String maHang, String maNCC, String maXX) {

this.soLuong = soLuong;

this.donGia = donGia;

this.maHang = maHang;

this.maNCC = maNCC;

this.maXX = maXX;

}

public ChungTu() {

}

}

## 2. Class HangHoa

package ClothingInventoryManagement.Model;

public class HangHoa {

public String maHang;

public String tenHang;

public String mauSac;

public String kichCo;

public int donGia;

public String xuatXu;

public HangHoa(String maHang, String tenHang, String mauSac, String kichCo, int donGia, String xuatXu) {

this.maHang = maHang;

this.tenHang = tenHang;

this.mauSac = mauSac;

this.kichCo = kichCo;

this.donGia = donGia;

this.xuatXu = xuatXu;

}

public HangHoa() {

}

}

## 3. Class NhaCC

package ClothingInventoryManagement.Model;

public class NhaCC {

public String maNCC;

public String tenNCC;

public String diaChi;

public String std;

public String email;

public NhaCC(String maNCC, String tenNCC, String diaChi, String std, String email) {

this.maNCC = maNCC;

this.tenNCC = tenNCC;

this.diaChi = diaChi;

this.std = std;

this.email = email;

}

public NhaCC() {

}

}

## 4. Class XuatXu

package ClothingInventoryManagement.Model;

public class XuatXu {

public String ten;

public String maXX;

public XuatXu(String ten, String maXX) {

this.ten = ten;

this.maXX = maXX;

}

public XuatXu() {

}

}

## 5. Class DeleteChungTu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.ChungTu;

public class DeleteChungTu {

private Connection connection;

public DeleteChungTu(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void delete(ChungTu data) {

String sql = "DELETE FROM CHUNGTU WHERE MAHANG=? AND MANCC=? AND MAXX=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.maHang);

statement.setString(2, data.maNCC);

statement.setString(3, data.maXX);

int rowsDeleted = statement.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Xóa dữ liệu thành công");

}

else {

System.out.println("Xóa dữ liêu thất bại vui lòng thử kiểm tra các mã bạn nhập có trùng khớp");

}

} catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình xóa dữ liêu: " + e);

}

}

public void delete(ArrayList<ChungTu> dataList) {

for(ChungTu data : dataList) {

delete(data);

}

}

}

## 6. Class DeleteHangHoa

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

public class DeleteHangHoa {

private Connection connection;

public DeleteHangHoa(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void delete(String id) {

String sql = "DELETE FROM HANGHOA WHERE MAHANG=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, id);

int rowsDeleted = statement.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Xóa thành công");

}

else {

System.out.println("Không xóa được có thể do bạn nhập sai MAHANG");

}

}

catch(Exception e) {

System.out.println("Có lỗi xãy ra: " + e);

}

}

public void delete(ArrayList<String> list) {

for(String id : list) {

delete(id);

}

}

}

## 7. Class DeleteNhaCC

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

public class DeleteNhaCC {

private Connection connection;

public DeleteNhaCC(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void delete(String id) {

String sql = "DELETE FROM NHACC WHERE MANCC=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, id);

int rowsDeleted = statement.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Xóa thành công");

}

else {

System.out.println("Không xóa được có thể do bạn nhập sai MANCC");

}

}

catch(Exception e) {

System.out.println("Có lỗi xãy ra: " + e);

}

}

public void delete(ArrayList<String> list) {

for(String id : list) {

delete(id);

}

}

}

## 8. Class DeleteXuatXu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

public class DeleteXuatXu {

private Connection connection;

public DeleteXuatXu(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void delete(String id) {

String sql = "DELETE FROM XUATXU WHERE MAXX=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, id);

int rowsDeleted = statement.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Xóa thành công");

}

else {

System.out.println("Không xóa được có thể do bạn nhập sai MAXX");

}

}

catch(Exception e) {

System.out.println("Có lỗi xãy ra: " + e);

}

}

public void delete(ArrayList<String> list) {

for(String id : list) {

delete(id);

}

}

}

## 9. Class InsertChungTu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.ChungTu;

public class InsertChungTu {

private Connection connection;

public InsertChungTu(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void insert(ChungTu data) {

String sql = "INSERT INTO CHUNGTU (SOLUONG, DONGIA, MAHANG, MANCC, MAXX) VALUES (?, ?, ?, ?, ?)";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setInt(1, data.soLuong);

statement.setInt(2, data.donGia);

statement.setString(3, data.maHang);

statement.setString(4, data.maNCC);

statement.setString(5, data.maXX);

int rowsInserted = statement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("Thêm dữ liệu thành công");

}

else {

System.out.println("Thêm dữ liêu thất bại");

}

} catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình thêm dữ liêu: " + e);

}

}

public void insert(ArrayList<ChungTu> dataList) {

for(ChungTu data : dataList) {

insert(data);

}

}

}

## **10. Class InsertHangHoa**

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.HangHoa;

public class InsertHangHoa {

private Connection connection;

public InsertHangHoa(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void insert(HangHoa data) {

String sql = "INSERT INTO HANGHOA (MAHANG, TENHANG, MAUSAC, KICHCO, DONGIA, XUATXU) VALUES (?, ?, ?, ?, ?, ?);";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.maHang);

statement.setString(2, data.tenHang);

statement.setString(3, data.mauSac);

statement.setString(4, data.kichCo);

statement.setInt(5, data.donGia);

statement.setString(6, data.xuatXu);

int rowsInserted = statement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("Thêm dữ liệu thành công");

}

} catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình thêm dữ liêu: " + e);

}

}

public void insert(ArrayList<HangHoa> dataList) {

for(HangHoa data : dataList) {

this.insert(data);

}

}

}

## 11. Class InsertNhaCC

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.NhaCC;

public class InsertNhaCC {

private Connection connection;

public InsertNhaCC(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void insert(NhaCC data) {

String sql = "INSERT INTO NHACC (MANCC, TENNCC, DIACHI, SDT, EMAIL) VALUES (?, ?, ?, ?, ?)";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.maNCC);

statement.setString(2, data.tenNCC);

statement.setString(3, data.diaChi);

statement.setString(4, data.std);

statement.setString(5, data.email);

int rowsInserted = statement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("Thêm dữ liệu thành công");

}

} catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình thêm dữ liêu: " + e);

}

}

public void insert(ArrayList<NhaCC> dataList) {

for(NhaCC data : dataList) {

this.insert(data);

}

}

}

## 12. Class InsertXuatXu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.XuatXu;

public class InsertXuatXu {

private Connection connection;

public InsertXuatXu(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void insert(XuatXu data) {

String sql = "INSERT INTO XUATXU (MAXX, TEN) VALUES (?, ?)";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.maXX);

statement.setString(2, data.ten);

int rowsInserted = statement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("Thêm dữ liệu thành công");

}

} catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình thêm dữ liêu: " + e);

}

}

public void insert(ArrayList<XuatXu> dataList) {

for(XuatXu data : dataList) {

this.insert(data);

}

}

}

## 13. Class ReadChungTu

package ClothingInventoryManagement.Controler;

import java.io.Console;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Helper.UserInput;

import ClothingInventoryManagement.Model.ChungTu;

public class ReadChungTu {

public static ArrayList<ChungTu> readFromDataBase(Connection connection, String extend) {

ArrayList<ChungTu> data = new ArrayList<ChungTu>();

try {

Statement stat = connection.createStatement();

String query = "Select \* from CHUNGTU " + extend;

ResultSet res = stat.executeQuery(query);

while(res.next()) {

data.add(new ChungTu(res.getInt(2), res.getInt(3), res.getString(4), res.getString(5), res.getString(6)));

}

System.out.println("Tải dữ liệu về thành công");

res.close();

}

catch (Exception e){

System.out.println(e);

}

return data;

}

public static ArrayList<ChungTu> readFromFile(String fileName) {

ArrayList<String> listData = new ArrayList<String>();

ArrayList<ChungTu> res = new ArrayList<ChungTu>();

ChungTu atho = new ChungTu();

if(!ClassName.getNameWithFileName(fileName).equalsIgnoreCase(ClassName.getNameWithClass(atho))) {

System.out.println("Nhập tên file không đúng định dạng");

return res;

}

listData = ManipulationFile.reader(fileName);

for(String data : listData) {

String[] parts = data.split("\t");

if(parts.length == 0) {

System.out.println("Có lỗi xảy ra");

break;

}

try {

ChungTu temp = new ChungTu(Integer.parseInt(parts[0]), Integer.parseInt(parts[1]), parts[2], parts[3], parts[4]);

res.add(temp);

}

catch(Exception e) {

}

}

return res;

}

public static ArrayList<ChungTu> readFromKeyboad() {

ArrayList<ChungTu> res = new ArrayList<ChungTu>();

Console scr = System.console();

System.out.println("Nhập số lượng bản ghi");

int numberEntry = Integer.parseInt(scr.readLine());

for(int i = 0; i < numberEntry; i ++) {

System.out.println("Bản ghi " + (i + 1));

ChungTu instance = new ChungTu();

try {

UserInput.InputIntoInstance(instance);

} catch (Exception e) {

System.out.println("Có lỗi sảy ra và hệ thống đả hủy bản ghi " + (i + 1));

continue;

}

res.add(instance);

}

return res;

}

}

## 14. Class ReadHangHoa

package ClothingInventoryManagement.Controler;

import java.io.Console;

import java.util.ArrayList;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Helper.UserInput;

import ClothingInventoryManagement.Model.HangHoa;

public class ReadHangHoa {

public static ArrayList<HangHoa> readFromDataBase(Connection connection, String extend) {

ArrayList<HangHoa> data = new ArrayList<HangHoa>();

try {

Statement stat = connection.createStatement();

String query = "Select \* from HANGHOA " + extend;

ResultSet res = stat.executeQuery(query);

while(res.next()) {

data.add(new HangHoa(res.getString(1), res.getString(2), res.getString(3), res.getString(4), res.getInt(5), res.getString(6)));

}

System.out.println("Tải dữ liệu về thành công");

res.close();

}

catch (Exception e){

System.out.println(e);

}

return data;

}

public static ArrayList<HangHoa> readFromFile(String fileName) {

ArrayList<String> listData = new ArrayList<String>();

ArrayList<HangHoa> res = new ArrayList<HangHoa>();

HangHoa atho = new HangHoa();

if(!ClassName.getNameWithFileName(fileName).equalsIgnoreCase(ClassName.getNameWithClass(atho))) {

System.out.println("Nhập tên file không đúng định dạng");

return res;

}

listData = ManipulationFile.reader(fileName);

for(String data : listData) {

String[] parts = data.split("\t");

if(parts.length == 0) {

System.out.println("Có lỗi xảy ra");

break;

}

try {

HangHoa temp = new HangHoa(parts[0], parts[1], parts[2], parts[3], Integer.parseInt(parts[4]), parts[5]);

res.add(temp);

}

catch(Exception e) {

}

}

return res;

}

public static ArrayList<HangHoa> readFromKeyboad() {

ArrayList<HangHoa> res = new ArrayList<HangHoa>();

Console scr = System.console();

System.out.println("Nhập số lượng bản ghi");

int numberEntry = Integer.parseInt(scr.readLine());

for(int i = 0; i < numberEntry; i ++) {

System.out.println("Bản ghi " + (i + 1));

HangHoa instance = new HangHoa();

try {

UserInput.InputIntoInstance(instance);

} catch (Exception e) {

System.out.println("Có lỗi sảy ra và hệ thống đả hủy bản ghi " + (i + 1));

continue;

}

res.add(instance);

}

return res;

}

}

## 15. Class ReadNhaCC

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.io.Console;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Helper.UserInput;

import ClothingInventoryManagement.Model.NhaCC;

public class ReadNhaCC {

public static ArrayList<NhaCC> readFromDataBase(Connection connection, String extend) {

ArrayList<NhaCC> data = new ArrayList<NhaCC>();

try {

Statement stat = connection.createStatement();

String query = "Select \* from NHACC " + extend;

ResultSet res = stat.executeQuery(query);

while(res.next()) {

data.add(new NhaCC(res.getString(1), res.getString(2), res.getString(3), res.getString(4), res.getString(5)));

}

System.out.println("Tải dữ liệu về thành công");

res.close();

}

catch (Exception e){

System.out.println(e);

}

return data;

}

public static ArrayList<NhaCC> readFromFile(String fileName) {

ArrayList<String> listData = new ArrayList<String>();

ArrayList<NhaCC> res = new ArrayList<NhaCC>();

NhaCC atho = new NhaCC();

if(!ClassName.getNameWithFileName(fileName).equalsIgnoreCase(ClassName.getNameWithClass(atho))) {

System.out.println("Nhập tên file không đúng định dạng");

return res;

}

listData = ManipulationFile.reader(fileName);

for(String data : listData) {

String[] parts = data.split("\t");

if(parts.length == 0) {

System.out.println("Có lỗi xảy ra");

break;

}

try {

NhaCC temp = new NhaCC(parts[0], parts[1], parts[2], parts[3], parts[4]);

res.add(temp);

}

catch(Exception e) {

}

}

return res;

}

public static ArrayList<NhaCC> readFromKeyboad() {

ArrayList<NhaCC> res = new ArrayList<NhaCC>();

Console scr = System.console();

System.out.println("Nhập số lượng bản ghi");

int numberEntry = Integer.parseInt(scr.readLine());

for(int i = 0; i < numberEntry; i ++) {

System.out.println("Bản ghi " + (i + 1));

NhaCC instance = new NhaCC();

try {

UserInput.InputIntoInstance(instance);

} catch (Exception e) {

System.out.println("Có lỗi sảy ra và hệ thống đả hủy bản ghi " + (i + 1));

continue;

}

res.add(instance);

}

return res;

}

}

## 16. Class ReadXuatXu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.io.Console;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Helper.UserInput;

import ClothingInventoryManagement.Model.XuatXu;

public class ReadXuatXu {

public static ArrayList<XuatXu> readFromDataBase(Connection connection, String extend) {

ArrayList<XuatXu> data = new ArrayList<XuatXu>();

try {

Statement stat = connection.createStatement();

String query = "Select \* from XuatXu " + extend;

ResultSet res = stat.executeQuery(query);

while(res.next()) {

data.add(new XuatXu(res.getString(1), res.getString(2)));

}

System.out.println("Tải dữ liệu về thành công");

res.close();

}

catch (Exception e){

System.out.println(e);

}

return data;

}

public static ArrayList<XuatXu> readFromFile(String fileName) {

ArrayList<String> listData = new ArrayList<String>();

ArrayList<XuatXu> res = new ArrayList<XuatXu>();

XuatXu atho = new XuatXu();

if(!ClassName.getNameWithFileName(fileName).equalsIgnoreCase(ClassName.getNameWithClass(atho))) {

System.out.println("Nhập tên file không đúng định dạng");

return res;

}

listData = ManipulationFile.reader(fileName);

for(String data : listData) {

String[] parts = data.split("\t");

if(parts.length == 0) {

System.out.println("Có lỗi xảy ra");

break;

}

try {

XuatXu temp = new XuatXu(parts[0], parts[1]);

res.add(temp);

}

catch(Exception e) {

}

}

return res;

}

public static ArrayList<XuatXu> readFromKeyboad() {

ArrayList<XuatXu> res = new ArrayList<XuatXu>();

Console scr = System.console();

System.out.println("Nhập số lượng bản ghi");

int numberEntry = Integer.parseInt(scr.readLine());

for(int i = 0; i < numberEntry; i ++) {

System.out.println("Bản ghi " + (i + 1));

XuatXu instance = new XuatXu();

try {

UserInput.InputIntoInstance(instance);

} catch (Exception e) {

System.out.println("Có lỗi sảy ra và hệ thống đả hủy bản ghi " + (i + 1));

continue;

}

res.add(instance);

}

return res;

}

}

## 17. Class UpdateChungTu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.ChungTu;

public class UpdateChungTu {

private Connection connection;

public UpdateChungTu(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void update(ChungTu data) {

String sql = "UPDATE CHUNGTU SET SOLUONG=?, DONGIA=? WHERE MAHANG=? AND MANCC=? AND MAXX=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setInt(1, data.soLuong);

statement.setInt(2, data.donGia);

statement.setString(3, data.maHang);

statement.setString(4, data.maNCC);

statement.setString(5, data.maXX);

int rowsUpdated = statement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Cập nhật dữ liệu thành công");

}

else {

System.out.println("Cập nhật dữ liêu thất bại vui lòng thử kiểm tra các mã bạn nhập có trùng khớp");

}

} catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình cập nhật dữ liêu: " + e);

}

}

public void update(ArrayList<ChungTu> dataList) {

for(ChungTu data : dataList) {

update(data);

}

}

}

## 18. Class UpdateHangHoa

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.HangHoa;

public class UpdateHangHoa {

private Connection connection;

public UpdateHangHoa(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void update(HangHoa data) {

String sql = "UPDATE HANGHOA SET TENHANG=?, MAUSAC=?, KICHCO=?, DONGIA=?, XUATXU=? WHERE MAHANG=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.tenHang);

statement.setString(2, data.mauSac);

statement.setString(3, data.kichCo);

statement.setInt(4, data.donGia);

statement.setString(5, data.xuatXu);

statement.setString(6, data.maHang);

int rowsUpdated = statement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Cập nhật thành công");

}

else {

System.out.println("Không cập nhật được có thể do bạn nhập sai MAHANG");

}

}

catch(Exception e) {

System.out.println("Có lỗi xảy ra: " + e);

}

}

public void update(ArrayList<HangHoa> list) {

for(HangHoa data : list) {

update(data);

}

}

}

## 19. Class UpdateNhaCC

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.NhaCC;

public class UpdateNhaCC {

private Connection connection;

public UpdateNhaCC(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void update(NhaCC data) {

String sql = "UPDATE NHACC SET TENNCC=? DIACHI=? SDT=? EMAIL=? WHERE MANCC=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.tenNCC);

statement.setString(2, data.diaChi);

statement.setString(3, data.std);

statement.setString(4, data.email);

statement.setString(5, data.maNCC);

int rowsUpdated = statement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Cập nhật thành công");

}

else {

System.out.println("Không cập nhật được có thể do bạn nhập sai MANCC");

}

}

catch(Exception e) {

System.out.println("Có lỗi xảy ra: " + e);

}

}

public void update(ArrayList<NhaCC> list) {

for(NhaCC data : list) {

update(data);

}

}

}

## 20. Class UpdateXuatXu

package ClothingInventoryManagement.Controler;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import ClothingInventoryManagement.Model.XuatXu;

public class UpdateXuatXu {

private Connection connection;

public UpdateXuatXu(Connection connection) {

this.connection = connection;

}

public void setConnection(Connection connection) {

this.connection = connection;

}

public void update(XuatXu data) {

String sql = "UPDATE XUATXU SET TEN=? WHERE MAXX=?";

try {

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, data.ten);

statement.setString(2, data.maXX);

int rowsUpdated = statement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Cập nhật thành công");

}

else {

System.out.println("Không cập nhật được có thể do bạn nhập sai MAXX");

}

}

catch(Exception e) {

System.out.println("Có lỗi xảy ra: " + e);

}

}

public void update(ArrayList<XuatXu> list) {

for(XuatXu data : list) {

update(data);

}

}

}

## 21. Class Connecting

package ClothingInventoryManagement.DatabaseConnection;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class Connecting {

private Connection connection;

public Connecting(String url, String user, String password) {

try {

DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());

connection = DriverManager.getConnection(url, user, password);

}

catch (SQLException e) {

System.out.println("Lỗi kết nối đến cơ sở dữ liệu");

e.printStackTrace();

}

}

public Connection getConnection() {

if(connection != null) {

System.out.println("Kết nối thành công");

}

return connection;

}

}

## 22. Class ClassName

package ClothingInventoryManagement.Helper;

import java.util.ArrayList;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.lang.reflect.Field;

public class ClassName {

public static String getNameWithFileName(String fileName) {

Pattern pattern = Pattern.compile("(\\D+)(\\d+)"); // tìm các phần tử không phải số (\D+) và phần tử số (\d+)

Matcher matcher = pattern.matcher(fileName);

String className = "";

if (matcher.find()) {

className = matcher.group(1); // lấy phần tử không phải số

}

return className;

}

public static ArrayList<String> getClassPropertyName(Object obj) {

Class<?> tObj = obj.getClass();

Field[] fields = tObj.getDeclaredFields();

ArrayList<String> res = new ArrayList<String>();

for(Field field : fields) {

res.add(field.getName());

}

return res;

}

public static String getNameWithClass(Object obj) {

return obj.getClass().getSimpleName();

}

}

## 23. Class CreateTable

package ClothingInventoryManagement.Helper;

import java.util.ArrayList;

import de.vandermeer.asciitable.AsciiTable;

import java.lang.reflect.Field;

public class CreateTable {

public static <T> String createTable(ArrayList<T> array) {

AsciiTable at = new AsciiTable();

if(array.size() == 0) return "";

Class<?> tObj = array.get(0).getClass();

Field[] fields = tObj.getDeclaredFields();

Object[] titles;

ArrayList<Object> temp = new ArrayList<Object>();

for(Field field : fields) {

temp.add(field.getName());

}

titles = temp.toArray();

at.addRule();

at.addRow(titles);

at.addRule();

for(T obj : array) {

Object[] row;

ArrayList<Object> tempR = new ArrayList<Object>();

try {

for(Field field : fields) {

tempR.add(field.get(obj));

}

}

catch(Exception e) {

System.out.println("Có lỗi xảy ra");

}

row = tempR.toArray();

at.addRow(row);

at.addRule();

}

return at.render();

}

}

## 24. Class DataFormat

package ClothingInventoryManagement.Helper;

import java.util.ArrayList;

import java.lang.reflect.Field;

public class DataFormat {

public static <T> String Format(ArrayList<T> array) {

String res = "";

if(array.size() == 0) return res;

Class<?> tObj = array.get(0).getClass();

Field[] fields = tObj.getDeclaredFields();

for(T obj : array) {

String tempR = "";

try {

for(Field field : fields) {

tempR += (NormalizeString(field.get(obj).toString()) + "\t");

}

tempR = (tempR.substring(0, tempR.length() - 1) + "\n");

}

catch(Exception e) {

System.out.println("Có lỗi xảy ra");

}

res += tempR;

}

return res;

}

public static String NormalizeString(String input) {

String output = input.trim();

return output;

}

}

## 25. Class ManipulationFile

package ClothingInventoryManagement.Helper;

import java.util.ArrayList;

import java.util.Scanner;

import java.io.File;

import java.io.FileWriter;

import java.util.Date;

public class ManipulationFile {

public static ArrayList<String> reader(String fileName) {

ArrayList<String> res = new ArrayList<String>();

try {

File f = new File("D:\\Project\\ClothingInventory\\ClothingInventoryManagerment\\src\\main\\java\\ClothingInventoryManagement\\Stored\\" + fileName + ".txt");

Scanner scanner = new Scanner(f);

while (scanner.hasNextLine()) {

String line = scanner.nextLine();

res.add(line);

}

scanner.close();

}

catch (Exception e) {

System.out.println("Có lỗi xảy ra hoặc file không tồn tại");

}

return res;

}

public static void Writer(String fileName, String data) {

Date d = new Date();

long time = d.getTime();

String label = Long.toString(time);

String fullFileName = fileName + label;

String filePath = "D:\\Project\\ClothingInventory\\ClothingInventoryManagerment\\src\\main\\java\\ClothingInventoryManagement\\Stored\\" + fullFileName + ".txt";

try {

File newFile = new File(filePath);

if(newFile.createNewFile()) {

System.out.println("Tạo file: " + newFile.getName());

}

else {

System.out.println("File đã tồn tại");

}

}

catch(Exception e) {

System.out.print("Có lỗi sảy ra trong quá trình tạo file: " + e);

}

try {

FileWriter writer = new FileWriter(filePath);

writer.write(data);

System.out.println("Xuất ra file thành công tên file của bạn là " + fullFileName + ".txt");

writer.close();

}

catch (Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình viết file: " + e);

}

}

}

## 26. Class UserInput

package ClothingInventoryManagement.Helper;

import java.io.Console;

import java.lang.reflect.Field;

public class UserInput {

public static <T> void InputIntoInstance(T sample) throws Exception{

Class<?> tObj = sample.getClass();

Field[] fields = tObj.getDeclaredFields();

Console sc = System.console();

try {

for(Field field : fields) {

System.out.println("Nhap " + field.getName());

if (field.getType() == int.class) {

int data = Integer.parseInt(sc.readLine());

field.setInt(sample, data);

}

else {

String data = sc.readLine();

field.set(sample, data);

}

}

}

catch(Exception e) {

System.out.println("Có lỗi xảy ra trong quá trình nhâp dữ liệu");

throw e;

}

}

}

## 27. Class ClearConsole

package ClothingInventoryManagement.View;

public class ClearConsole {

public static void clear() {

System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");

}

}

## 28. Class MenuChungTu

package ClothingInventoryManagement.View;

import java.io.Console;

import java.sql.Connection;

import java.util.ArrayList;

import ClothingInventoryManagement.Controler.DeleteChungTu;

import ClothingInventoryManagement.Controler.InsertChungTu;

import ClothingInventoryManagement.Controler.ReadChungTu;

import ClothingInventoryManagement.Controler.UpdateChungTu;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.DataFormat;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Model.ChungTu;

public class MenuChungTu {

public static void generate(Connection connection) {

boolean menuExit = false;

ArrayList<ChungTu> data = new ArrayList<ChungTu>();

ChungTu sample = new ChungTu();

while(!menuExit) {

ClearConsole.clear();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_BẢNG CHỨNG TỪ\_\_\_\_\_\_\_\_\_");

System.out.println("1. Tải dữ liệu từ database");

System.out.println("2. Lấy dữ liệu từ file và thêm vào database");

System.out.println("3. Thêm dữ liệu từ bàn phím");

System.out.println("4. Xem bảng và xuất ra file");

System.out.println("5. Cập nhật dữ liệu");

System.out.println("6. Xóa dữ liệu");

System.out.println("7. Tìm dữ liệu");

System.out.println("8. Thoát");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console c = System.console();

int choise = Integer.parseInt(c.readLine());

switch (choise) {

case 1:

{

ClearConsole.clear();

String extend = "";

System.out.println("Bạn có cần sắp xếp không dữ liêu không (y/n)");

String hasSort = c.readLine();

if(hasSort.equalsIgnoreCase("y")) {

ArrayList<String> propertyName = ClassName.getClassPropertyName(sample);

extend += "order by ";

System.out.println("\_\_\_\_\_\_\_\_\_Sắp xếp theo\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(int i = 0; i < propertyName.size(); i++) {

System.out.println((i + 1) + ". " + propertyName.get(i));

}

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

try {

int sortCol = Integer.parseInt(c.readLine());

if(sortCol > propertyName.size()) sortCol = 1;

extend += propertyName.get(sortCol - 1).toUpperCase();

}

catch (Exception e) {

extend += propertyName.get(0);

System.out.println("Nhập sai hệ thống đã chọn cột sắp xếp mặc định");

}

System.out.println("Sắp xếp theo thứ tự nào");

System.out.println("1. ASC");

System.out.println("2. DESC");

try {

int orderChoise = Integer.parseInt(c.readLine()) ;

if(orderChoise == 1) extend += " ASC";

else extend += " DESC";

}

catch (Exception e) {

extend += " ASC";

System.out.println("Nhập sai hệ thống đã chọn thứ tự sắp xếp mặc định");

}

}

data = ReadChungTu.readFromDataBase(connection, extend);

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 2:

{

ClearConsole.clear();

System.out.println("Nhập tên file ");

String fileName = c.readLine();

ArrayList<ChungTu> temp = new ArrayList<ChungTu>();

temp = ReadChungTu.readFromFile(fileName);

if(temp.size() > 0) {

ShowChungTu.view(temp);

System.out.print("Xác nhân thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertChungTu insert = new InsertChungTu(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 3:

{

ClearConsole.clear();

ArrayList<ChungTu> temp = new ArrayList<ChungTu>();

temp = ReadChungTu.readFromKeyboad();

if(temp.size() > 0) {

ShowChungTu.view(temp);

System.out.print("Xác nhận thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertChungTu insert = new InsertChungTu(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 4:

{

ClearConsole.clear();

ShowChungTu.view(data);

System.out.println("Bạn có muốn xuất bảng dữ liệu ra file không (y/n)");

String exportingDataToFileComfirm = c.readLine();

if(exportingDataToFileComfirm.equalsIgnoreCase("y")) {

String fileName = "ChungTu";

ManipulationFile.Writer(fileName, DataFormat.Format(data));

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 5:

{

ClearConsole.clear();

ArrayList<ChungTu> temp = new ArrayList<ChungTu>();

System.out.println("Yêu cầu nhập đầy đủ dữ liệu để cập nhật và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

temp = ReadChungTu.readFromKeyboad();

if(temp.size() > 0) {

ShowChungTu.view(temp);

System.out.print("Xác nhận update dữ liệu vào database (y/n)");

String updateIntoDatabaseComfirm = c.readLine();

if(updateIntoDatabaseComfirm.equalsIgnoreCase("y")) {

UpdateChungTu insert = new UpdateChungTu(connection);

insert.update(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 6:

{

ClearConsole.clear();

ArrayList<ChungTu> temp = new ArrayList<ChungTu>();

System.out.println("Yêu cầu nhập đầy đủ các mã để xóa và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

System.out.println("Nhập số bản ghi cần xóa");

try {

int numberIntance = Integer.parseInt(c.readLine());

for(int i = 0; i < numberIntance; i++) {

ChungTu instance = new ChungTu();

System.out.println("Nhập bản ghi " + (i + 1));

System.out.println("Nhập mã hàng");

instance.maHang = c.readLine();

System.out.println("Nhập mã nhà cung cấp");

instance.maNCC = c.readLine();

System.out.println("Nhập mã xuất xứ");

instance.maXX = c.readLine();

temp.add(instance);

}

} catch (Exception e) {

System.out.println("Đã xảy ra lỗi có thể do bạn nhập số bản ghi hoặc mã không hợp lệ");

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

if(temp.size() > 0) {

ShowChungTu.view(temp);

System.out.print("Xác nhận xóa dữ liệu database (y/n)");

String deleteDatabaseComfirm = c.readLine();

if(deleteDatabaseComfirm.equalsIgnoreCase("y")) {

DeleteChungTu del = new DeleteChungTu(connection);

del.delete(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 7:

{

ClearConsole.clear();

ArrayList<ChungTu> temp = new ArrayList<ChungTu>();

System.out.println("Đảm bảo rằng là bạn đã tải dữ liệu từ cơ sở dữ liệu nếu không vui lòng thực hiên lựa chọn 1");

System.out.println("Bạn đã tải dữ liệu về chưa? (y/n)");

String hasDownDataFromDatabase = c.readLine();

if(hasDownDataFromDatabase.equalsIgnoreCase("n")) {

break;

}

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

ChungTu instance = new ChungTu();

System.out.println("Nhập các mã để tìm dữ liệu");

System.out.println("Nhập mã hàng");

instance.maHang = c.readLine();

System.out.println("Nhập mã nhà cung cấp");

instance.maNCC = c.readLine();

System.out.println("Nhập mã xuất xứ");

instance.maXX = c.readLine();

for(ChungTu i : data) {

if(DataFormat.NormalizeString(i.maHang).equalsIgnoreCase(instance.maHang) && DataFormat.NormalizeString(i.maNCC).equalsIgnoreCase(instance.maNCC) && DataFormat.NormalizeString(i.maXX).equalsIgnoreCase(instance.maXX)) {

temp.add(i);

}

}

if(temp.size() > 0) {

System.out.println("Đã tìm thấy");

ShowChungTu.view(temp);

}

else {

System.out.println("Không tìm thấy");

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 8:

{

menuExit = true;

break;

}

default:

break;

}

}

}

}

## 29. Class MenuHangHoa

package ClothingInventoryManagement.View;

import java.io.Console;

import java.sql.Connection;

import java.util.ArrayList;

import ClothingInventoryManagement.Controler.DeleteHangHoa;

import ClothingInventoryManagement.Controler.InsertHangHoa;

import ClothingInventoryManagement.Controler.ReadHangHoa;

import ClothingInventoryManagement.Controler.UpdateHangHoa;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.DataFormat;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Model.HangHoa;

public class MenuHangHoa {

public static void generate(Connection connection) {

boolean menuExit = false;

ArrayList<HangHoa> data = new ArrayList<HangHoa>();

HangHoa sample = new HangHoa();

while(!menuExit) {

ClearConsole.clear();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_BẢNG HÀNG HÓA\_\_\_\_\_\_\_\_\_");

System.out.println("1. Tải dữ liệu từ database");

System.out.println("2. Lấy dữ liệu từ file và thêm vào database");

System.out.println("3. Thêm dữ liệu từ bàn phím");

System.out.println("4. Xem bảng và xuất ra file");

System.out.println("5. Cập nhật dữ liệu");

System.out.println("6. Xóa dữ liệu");

System.out.println("7. Tìm dữ liệu");

System.out.println("8. Thoát");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console c = System.console();

int choise = Integer.parseInt(c.readLine());

switch (choise) {

case 1:

{

ClearConsole.clear();

String extend = "";

System.out.println("Bạn có cần sắp xếp không dữ liêu không (y/n)");

String hasSort = c.readLine();

if(hasSort.equalsIgnoreCase("y")) {

ArrayList<String> propertyName = ClassName.getClassPropertyName(sample);

extend += "order by ";

System.out.println("\_\_\_\_\_\_\_\_\_Sắp xếp theo\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(int i = 0; i < propertyName.size(); i++) {

System.out.println((i + 1) + ". " + propertyName.get(i));

}

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

try {

int sortCol = Integer.parseInt(c.readLine());

if(sortCol > propertyName.size()) sortCol = 1;

extend += propertyName.get(sortCol - 1).toUpperCase();

}

catch (Exception e) {

extend += propertyName.get(0);

System.out.println("Nhập sai hệ thống đã chọn cột sắp xếp mặc định");

}

System.out.println("Sắp xếp theo thứ tự nào");

System.out.println("1. ASC");

System.out.println("2. DESC");

try {

int orderChoise = Integer.parseInt(c.readLine()) ;

if(orderChoise == 1) extend += " ASC";

else extend += " DESC";

}

catch (Exception e) {

extend += " ASC";

System.out.println("Nhập sai hệ thống đã chọn thứ tự sắp xếp mặc định");

}

}

data = ReadHangHoa.readFromDataBase(connection, extend);

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 2:

{

ClearConsole.clear();

System.out.println("Nhập tên file ");

String fileName = c.readLine();

ArrayList<HangHoa> temp = new ArrayList<HangHoa>();

temp = ReadHangHoa.readFromFile(fileName);

if(temp.size() > 0) {

ShowHangHoa.view(temp);

System.out.print("Xác nhân thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertHangHoa insert = new InsertHangHoa(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 3:

{

ClearConsole.clear();

ArrayList<HangHoa> temp = new ArrayList<HangHoa>();

temp = ReadHangHoa.readFromKeyboad();

if(temp.size() > 0) {

ShowHangHoa.view(temp);

System.out.print("Xác nhận thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertHangHoa insert = new InsertHangHoa(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 4:

{

ClearConsole.clear();

ShowHangHoa.view(data);

System.out.println("Bạn có muốn xuất bảng dữ liệu ra file không (y/n)");

String exportingDataToFileComfirm = c.readLine();

if(exportingDataToFileComfirm.equalsIgnoreCase("y")) {

String fileName = "HangHoa";

ManipulationFile.Writer(fileName, DataFormat.Format(data));

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 5:

{

ClearConsole.clear();

ArrayList<HangHoa> temp = new ArrayList<HangHoa>();

System.out.println("Yêu cầu nhập đầy đủ dữ liệu để cập nhật và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

temp = ReadHangHoa.readFromKeyboad();

if(temp.size() > 0) {

ShowHangHoa.view(temp);

System.out.print("Xác nhận update dữ liệu vào database (y/n)");

String updateIntoDatabaseComfirm = c.readLine();

if(updateIntoDatabaseComfirm.equalsIgnoreCase("y")) {

UpdateHangHoa insert = new UpdateHangHoa(connection);

insert.update(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 6:

{

ClearConsole.clear();

ArrayList<String> temp = new ArrayList<String>();

System.out.println("Yêu cầu nhập đầy đủ các mã để xóa và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

System.out.println("Nhập số bản ghi cần xóa");

try {

int numberIntance = Integer.parseInt(c.readLine());

for(int i = 0; i < numberIntance; i++) {

System.out.println("Nhập bản ghi " + (i + 1));

System.out.println("Nhập mã hàng");

temp.add(c.readLine());

}

} catch (Exception e) {

System.out.println("Đã xảy ra lỗi có thể do bạn nhập số bản ghi hoặc mã không hợp lệ");

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

if(temp.size() > 0) {

System.out.print("Xác nhận xóa dữ liệu database (y/n)");

String deleteDatabaseComfirm = c.readLine();

if(deleteDatabaseComfirm.equalsIgnoreCase("y")) {

DeleteHangHoa del = new DeleteHangHoa(connection);

del.delete(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 7:

{

ClearConsole.clear();

ArrayList<HangHoa> temp = new ArrayList<HangHoa>();

System.out.println("Đảm bảo rằng là bạn đã tải dữ liệu từ cơ sở dữ liệu nếu không vui lòng thực hiên lựa chọn 1");

System.out.println("Bạn đã tải dữ liệu về chưa? (y/n)");

String hasDownDataFromDatabase = c.readLine();

if(hasDownDataFromDatabase.equalsIgnoreCase("n")) {

break;

}

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

HangHoa instance = new HangHoa();

System.out.println("Nhập mã để tìm dữ liệu");

System.out.println("Nhập mã hàng");

instance.maHang = c.readLine();

for(HangHoa i : data) {

if(DataFormat.NormalizeString(i.maHang).equalsIgnoreCase(instance.maHang)) {

temp.add(i);

}

}

if(temp.size() > 0) {

System.out.println("Đã tìm thấy");

ShowHangHoa.view(temp);

}

else {

System.out.println("Không tìm thấy");

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 8:

{

menuExit = true;

break;

}

default:

break;

}

}

}

}

## 30. Class MenuNhaCC

package ClothingInventoryManagement.View;

import java.io.Console;

import java.sql.Connection;

import java.util.ArrayList;

import ClothingInventoryManagement.Controler.DeleteNhaCC;

import ClothingInventoryManagement.Controler.InsertNhaCC;

import ClothingInventoryManagement.Controler.ReadNhaCC;

import ClothingInventoryManagement.Controler.UpdateNhaCC;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.DataFormat;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Model.NhaCC;

public class MenuNhaCC {

public static void generate(Connection connection) {

boolean menuExit = false;

ArrayList<NhaCC> data = new ArrayList<NhaCC>();

NhaCC sample = new NhaCC();

while(!menuExit) {

ClearConsole.clear();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_BẢNG NHÀ CUNG CẤP\_\_\_\_\_\_\_\_\_");

System.out.println("1. Tải dữ liệu từ database");

System.out.println("2. Lấy dữ liệu từ file và thêm vào database");

System.out.println("3. Thêm dữ liệu từ bàn phím");

System.out.println("4. Xem bảng và xuất ra file");

System.out.println("5. Cập nhật dữ liệu");

System.out.println("6. Xóa dữ liệu");

System.out.println("7. Tìm dữ liệu");

System.out.println("8. Thoát");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console c = System.console();

int choise = Integer.parseInt(c.readLine());

switch (choise) {

case 1:

{

ClearConsole.clear();

String extend = "";

System.out.println("Bạn có cần sắp xếp không dữ liêu không (y/n)");

String hasSort = c.readLine();

if(hasSort.equalsIgnoreCase("y")) {

ArrayList<String> propertyName = ClassName.getClassPropertyName(sample);

extend += "order by ";

System.out.println("\_\_\_\_\_\_\_\_\_Sắp xếp theo\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(int i = 0; i < propertyName.size(); i++) {

System.out.println((i + 1) + ". " + propertyName.get(i));

}

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

try {

int sortCol = Integer.parseInt(c.readLine());

if(sortCol > propertyName.size()) sortCol = 1;

extend += propertyName.get(sortCol - 1).toUpperCase();

}

catch (Exception e) {

extend += propertyName.get(0);

System.out.println("Nhập sai hệ thống đã chọn cột sắp xếp mặc định");

}

System.out.println("Sắp xếp theo thứ tự nào");

System.out.println("1. ASC");

System.out.println("2. DESC");

try {

int orderChoise = Integer.parseInt(c.readLine()) ;

if(orderChoise == 1) extend += " ASC";

else extend += " DESC";

}

catch (Exception e) {

extend += " ASC";

System.out.println("Nhập sai hệ thống đã chọn thứ tự sắp xếp mặc định");

}

}

data = ReadNhaCC.readFromDataBase(connection, extend);

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 2:

{

ClearConsole.clear();

System.out.println("Nhập tên file ");

String fileName = c.readLine();

ArrayList<NhaCC> temp = new ArrayList<NhaCC>();

temp = ReadNhaCC.readFromFile(fileName);

if(temp.size() > 0) {

ShowNhaCC.view(temp);

System.out.print("Xác nhân thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertNhaCC insert = new InsertNhaCC(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 3:

{

ClearConsole.clear();

ArrayList<NhaCC> temp = new ArrayList<NhaCC>();

temp = ReadNhaCC.readFromKeyboad();

if(temp.size() > 0) {

ShowNhaCC.view(temp);

System.out.print("Xác nhận thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertNhaCC insert = new InsertNhaCC(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 4:

{

ClearConsole.clear();

ShowNhaCC.view(data);

System.out.println("Bạn có muốn xuất bảng dữ liệu ra file không (y/n)");

String exportingDataToFileComfirm = c.readLine();

if(exportingDataToFileComfirm.equalsIgnoreCase("y")) {

String fileName = "NhaCC";

ManipulationFile.Writer(fileName, DataFormat.Format(data));

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 5:

{

ClearConsole.clear();

ArrayList<NhaCC> temp = new ArrayList<NhaCC>();

System.out.println("Yêu cầu nhập đầy đủ dữ liệu để cập nhật và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

temp = ReadNhaCC.readFromKeyboad();

if(temp.size() > 0) {

ShowNhaCC.view(temp);

System.out.print("Xác nhận update dữ liệu vào database (y/n)");

String updateIntoDatabaseComfirm = c.readLine();

if(updateIntoDatabaseComfirm.equalsIgnoreCase("y")) {

UpdateNhaCC insert = new UpdateNhaCC(connection);

insert.update(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 6:

{

ClearConsole.clear();

ArrayList<String> temp = new ArrayList<String>();

System.out.println("Yêu cầu nhập đầy đủ các mã để xóa và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

System.out.println("Nhập số bản ghi cần xóa");

try {

int numberIntance = Integer.parseInt(c.readLine());

for(int i = 0; i < numberIntance; i++) {

System.out.println("Nhập bản ghi " + (i + 1));

System.out.println("Nhập mã hàng");

temp.add(c.readLine());

}

} catch (Exception e) {

System.out.println("Đã xảy ra lỗi có thể do bạn nhập số bản ghi hoặc mã không hợp lệ");

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

if(temp.size() > 0) {

System.out.print("Xác nhận xóa dữ liệu database (y/n)");

String deleteDatabaseComfirm = c.readLine();

if(deleteDatabaseComfirm.equalsIgnoreCase("y")) {

DeleteNhaCC del = new DeleteNhaCC(connection);

del.delete(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 7:

{

ClearConsole.clear();

ArrayList<NhaCC> temp = new ArrayList<NhaCC>();

System.out.println("Đảm bảo rằng là bạn đã tải dữ liệu từ cơ sở dữ liệu nếu không vui lòng thực hiên lựa chọn 1");

System.out.println("Bạn đã tải dữ liệu về chưa? (y/n)");

String hasDownDataFromDatabase = c.readLine();

if(hasDownDataFromDatabase.equalsIgnoreCase("n")) {

break;

}

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

NhaCC instance = new NhaCC();

System.out.println("Nhập mã để tìm dữ liệu");

System.out.println("Nhập mã nhà cung cáp");

instance.maNCC = c.readLine();

for(NhaCC i : data) {

if(DataFormat.NormalizeString(i.maNCC).equalsIgnoreCase(instance.maNCC)) {

temp.add(i);

}

}

if(temp.size() > 0) {

System.out.println("Đã tìm thấy");

ShowNhaCC.view(temp);

}

else {

System.out.println("Không tìm thấy");

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 8:

{

menuExit = true;

break;

}

default:

break;

}

}

}

}

## 31. Class MenuXuatXu

package ClothingInventoryManagement.View;

import java.io.Console;

import java.sql.Connection;

import java.util.ArrayList;

import ClothingInventoryManagement.Controler.DeleteXuatXu;

import ClothingInventoryManagement.Controler.InsertXuatXu;

import ClothingInventoryManagement.Controler.ReadXuatXu;

import ClothingInventoryManagement.Controler.UpdateXuatXu;

import ClothingInventoryManagement.Helper.ClassName;

import ClothingInventoryManagement.Helper.DataFormat;

import ClothingInventoryManagement.Helper.ManipulationFile;

import ClothingInventoryManagement.Model.XuatXu;

public class MenuXuatXu {

public static void generate(Connection connection) {

boolean menuExit = false;

ArrayList<XuatXu> data = new ArrayList<XuatXu>();

XuatXu sample = new XuatXu();

while(!menuExit) {

ClearConsole.clear();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_BẢNG HÀNG HÓA\_\_\_\_\_\_\_\_\_");

System.out.println("1. Tải dữ liệu từ database");

System.out.println("2. Lấy dữ liệu từ file và thêm vào database");

System.out.println("3. Thêm dữ liệu từ bàn phím");

System.out.println("4. Xem bảng và xuất ra file");

System.out.println("5. Cập nhật dữ liệu");

System.out.println("6. Xóa dữ liệu");

System.out.println("7. Tìm dữ liệu");

System.out.println("8. Thoát");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console c = System.console();

int choise = Integer.parseInt(c.readLine());

switch (choise) {

case 1:

{

ClearConsole.clear();

String extend = "";

System.out.println("Bạn có cần sắp xếp không dữ liêu không (y/n)");

String hasSort = c.readLine();

if(hasSort.equalsIgnoreCase("y")) {

ArrayList<String> propertyName = ClassName.getClassPropertyName(sample);

extend += "order by ";

System.out.println("\_\_\_\_\_\_\_\_\_Sắp xếp theo\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(int i = 0; i < propertyName.size(); i++) {

System.out.println((i + 1) + ". " + propertyName.get(i));

}

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

try {

int sortCol = Integer.parseInt(c.readLine());

if(sortCol > propertyName.size()) sortCol = 1;

extend += propertyName.get(sortCol - 1).toUpperCase();

}

catch (Exception e) {

extend += propertyName.get(0);

System.out.println("Nhập sai hệ thống đã chọn cột sắp xếp mặc định");

}

System.out.println("Sắp xếp theo thứ tự nào");

System.out.println("1. ASC");

System.out.println("2. DESC");

try {

int orderChoise = Integer.parseInt(c.readLine()) ;

if(orderChoise == 1) extend += " ASC";

else extend += " DESC";

}

catch (Exception e) {

extend += " ASC";

System.out.println("Nhập sai hệ thống đã chọn thứ tự sắp xếp mặc định");

}

}

data = ReadXuatXu.readFromDataBase(connection, extend);

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 2:

{

ClearConsole.clear();

System.out.println("Nhập tên file ");

String fileName = c.readLine();

ArrayList<XuatXu> temp = new ArrayList<XuatXu>();

temp = ReadXuatXu.readFromFile(fileName);

if(temp.size() > 0) {

ShowXuatXu.view(temp);

System.out.print("Xác nhân thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertXuatXu insert = new InsertXuatXu(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 3:

{

ClearConsole.clear();

ArrayList<XuatXu> temp = new ArrayList<XuatXu>();

temp = ReadXuatXu.readFromKeyboad();

if(temp.size() > 0) {

ShowXuatXu.view(temp);

System.out.print("Xác nhận thêm vào database (y/n)");

String insertIntoDatabaseComfirm = c.readLine();

if(insertIntoDatabaseComfirm.equalsIgnoreCase("y")) {

InsertXuatXu insert = new InsertXuatXu(connection);

insert.insert(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 4:

{

ClearConsole.clear();

ShowXuatXu.view(data);

System.out.println("Bạn có muốn xuất bảng dữ liệu ra file không (y/n)");

String exportingDataToFileComfirm = c.readLine();

if(exportingDataToFileComfirm.equalsIgnoreCase("y")) {

String fileName = "XuatXu";

ManipulationFile.Writer(fileName, DataFormat.Format(data));

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 5:

{

ClearConsole.clear();

ArrayList<XuatXu> temp = new ArrayList<XuatXu>();

System.out.println("Yêu cầu nhập đầy đủ dữ liệu để cập nhật và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

temp = ReadXuatXu.readFromKeyboad();

if(temp.size() > 0) {

ShowXuatXu.view(temp);

System.out.print("Xác nhận update dữ liệu vào database (y/n)");

String updateIntoDatabaseComfirm = c.readLine();

if(updateIntoDatabaseComfirm.equalsIgnoreCase("y")) {

UpdateXuatXu insert = new UpdateXuatXu(connection);

insert.update(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 6:

{

ClearConsole.clear();

ArrayList<String> temp = new ArrayList<String>();

System.out.println("Yêu cầu nhập đầy đủ các mã để xóa và đảm bảo các mã phải có trong bảng");

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

System.out.println("Nhập số bản ghi cần xóa");

try {

int numberIntance = Integer.parseInt(c.readLine());

for(int i = 0; i < numberIntance; i++) {

System.out.println("Nhập bản ghi " + (i + 1));

System.out.println("Nhập mã hàng");

temp.add(c.readLine());

}

} catch (Exception e) {

System.out.println("Đã xảy ra lỗi có thể do bạn nhập số bản ghi hoặc mã không hợp lệ");

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

if(temp.size() > 0) {

System.out.print("Xác nhận xóa dữ liệu database (y/n)");

String deleteDatabaseComfirm = c.readLine();

if(deleteDatabaseComfirm.equalsIgnoreCase("y")) {

DeleteXuatXu del = new DeleteXuatXu(connection);

del.delete(temp);

}

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 7:

{

ClearConsole.clear();

ArrayList<XuatXu> temp = new ArrayList<XuatXu>();

System.out.println("Đảm bảo rằng là bạn đã tải dữ liệu từ cơ sở dữ liệu nếu không vui lòng thực hiên lựa chọn 1");

System.out.println("Bạn đã tải dữ liệu về chưa? (y/n)");

String hasDownDataFromDatabase = c.readLine();

if(hasDownDataFromDatabase.equalsIgnoreCase("n")) {

break;

}

System.out.println("Ấn phím ENTER để tiếp tục");

c.readLine();

XuatXu instance = new XuatXu();

System.out.println("Nhập mã để tìm dữ liệu");

System.out.println("Nhập mã hàng");

instance.maXX = c.readLine();

for(XuatXu i : data) {

if(DataFormat.NormalizeString(i.maXX).equalsIgnoreCase(instance.maXX)) {

temp.add(i);

}

}

if(temp.size() > 0) {

System.out.println("Đã tìm thấy");

ShowXuatXu.view(temp);

}

else {

System.out.println("Không tìm thấy");

}

System.out.println("Ấn phím ENTER để thoát");

c.readLine();

break;

}

case 8:

{

menuExit = true;

break;

}

default:

break;

}

}

}

}

## 32. Class MenuMain

package ClothingInventoryManagement.View;

import java.io.Console;

import java.sql.Connection;

public class MenuMain {

public static void generate(Connection connection) {

boolean menuExit = false;

while(!menuExit) {

ClearConsole.clear();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_CHƯƠNG TRÌNH QUẢN LÝ KHO HÀNG\_\_\_\_\_\_\_\_\_");

System.out.println("1. Thao tác trên bảng CHỨNG TỪ");

System.out.println("2. Thao tác trên bảng HÀNG HÓA");

System.out.println("3. Thao tác trên bảng NHÀ CUNG CẤP");

System.out.println("4. Thao tác trên bảng XUẤT XỨ");

System.out.println("5. Thoát");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console c = System.console();

int choise = Integer.parseInt(c.readLine());

switch(choise) {

case 1:

{

MenuChungTu.generate(connection);

break;

}

case 2:

{

MenuHangHoa.generate(connection);

break;

}

case 3:

{

MenuNhaCC.generate(connection);

break;

}

case 4:

{

MenuXuatXu.generate(connection);

break;

}

case 5:

{

menuExit = true;

break;

}

default:

{

break;

}

}

}

}

}

## 33. Class ShowChungTu

package ClothingInventoryManagement.View;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.CreateTable;

import ClothingInventoryManagement.Model.ChungTu;

public class ShowChungTu {

public static void view(ArrayList<ChungTu> data) {

System.out.println(CreateTable.createTable(data));

}

}

## 34. Class ShowHangHoa

package ClothingInventoryManagement.View;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.CreateTable;

import ClothingInventoryManagement.Model.HangHoa;

public class ShowHangHoa {

public static void view(ArrayList<HangHoa> data) {

System.out.println(CreateTable.createTable(data));

}

}

## 35. Class ShowNhaCC

package ClothingInventoryManagement.View;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.CreateTable;

import ClothingInventoryManagement.Model.NhaCC;

public class ShowNhaCC {

public static void view(ArrayList<NhaCC> data) {

System.out.println(CreateTable.createTable(data));

}

}

## 36. Class ShowXuatXu

package ClothingInventoryManagement.View;

import java.util.ArrayList;

import ClothingInventoryManagement.Helper.CreateTable;

import ClothingInventoryManagement.Model.XuatXu;

public class ShowXuatXu {

public static void view(ArrayList<XuatXu> data) {

System.out.println(CreateTable.createTable(data));

}

}

## Class App

package ClothingInventoryManagement;

import ClothingInventoryManagement.DatabaseConnection.\*;

import java.sql.Connection;

import ClothingInventoryManagement.View.MenuMain;

public class App

{

public static void main( String[] args )

{

String url = "jdbc:sqlserver://LAPTOP-BF9157I6\\SQLEXPRESS:1433;databaseName=KHOQUANAO;encrypt=false;trustServerCertificate=true";

String user = "guest";

String password = "123456";

Connection connection = new Connecting(url, user, password).getConnection();

MenuMain.generate(connection);

}

}