|  |
| --- |
| **BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**  **TRƯỜNG ĐH TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**  **KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**    **BÁO CÁO ĐỒ ÁN MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**  **QUẢN LÝ KHO ĐIỆN THOẠI DÒNG**  **CAO CẤP**  Thành viên nhóm:  1.Nguyễn Thanh Hà (1050080048)  2.Trần Công Đạt (1050080047)  3.Phạm Ngọc Hà Minh (1050080061)  Lớp: 10 CNTT2    Giáo Viên Phụ Trách :  Ths Phạm Trọng Huynh  **TP. Hồ Chí Minh, ngày 26/ tháng 3 năm 2023** |

**MỤC LỤC**

[I. Mô Tả Bài Toán 4](#_Toc133484492)

[II.Xác định các lớp của bài toán 5](#_Toc133484493)

[1. Package quanlikho 5](#_Toc133484494)

[1.1. Class Phone 5](#_Toc133484495)

[1.2. Class iPhone kế thừa từ class Phone 5](#_Toc133484496)

[1.3.Class Samsung kế thừa từ class Phone 5](#_Toc133484497)

[1.4. Class Xiaomi kế thừa từ class Phone 5](#_Toc133484498)

[1.5.Class Oppo kế thừa từ class Phone 5](#_Toc133484499)

[1.6.Class Phieu 6](#_Toc133484500)

[1.7.Class PhieuNhap kế thừa từ class Phieu 6](#_Toc133484501)

[1.8. Class PhieuXuat kế thừa từ class Phieu 6](#_Toc133484502)

[1.9.Class Quanli 6](#_Toc133484503)

[Package DAO 7](#_Toc133484504)

[2.1. Class DAOInterface 7](#_Toc133484505)

[2.2. Class DAOiPhone implements class DAOInterface 7](#_Toc133484506)

[2.3. Class DAOSamsung implements class DAOInterface 7](#_Toc133484507)

[2.4. Class DAOOppo implements class DAOInterface 8](#_Toc133484508)

[2.5. Class DAOXiaomi implements class DAOInterface 8](#_Toc133484509)

[2.6.Class DAOPhieuInterface 9](#_Toc133484510)

[2.7 Class DAOPhieuNhap implements class DAOPhieuInterface 9](#_Toc133484511)

[2.8 Class DAOPhieuXuat implements class DAOPhieuInterface 9](#_Toc133484512)

[III. Mô tả thuật toán 10](#_Toc133484513)

[1. Chỉnh sửa thông tin điện thoại 10](#_Toc133484514)

[2. Hiển thị danh sách điện thoại hiện có trong cơ sở dữ liệu kho 10](#_Toc133484515)

[3. Tìm kiếm 10](#_Toc133484516)

[4. Sắp xếp 10](#_Toc133484517)

[IV. Sơ đồ 11](#_Toc133484518)

[IV. Cài đặt bài toán 15](#_Toc133484519)

[1.Package quanlikho 15](#_Toc133484520)

[1.1. Class iPhone 15](#_Toc133484521)

[1.2. Class Samsung 15](#_Toc133484522)

[1.3. Class Android 16](#_Toc133484523)

[1.4. Class Xiaomi 17](#_Toc133484524)

[1.5. Class PhieuNhap 17](#_Toc133484525)

[1.6.Class PhieuXuat 19](#_Toc133484526)

[1.7. Class Quanli 20](#_Toc133484527)

[1.7. Class Main 40](#_Toc133484528)

[2. Package DAO 41](#_Toc133484529)

[2.1. Class DAOInterface 41](#_Toc133484530)

[2.2. Class DAOiPhone 41](#_Toc133484531)

[2.3. Class DAOSamsung 56](#_Toc133484532)

[2.4. Class DAOXiaomi 70](#_Toc133484533)

[2.5. Class DAOOppo 83](#_Toc133484534)

[2.6. Class DAOPhieu 95](#_Toc133484535)

[2.7. Class DAOPhieuXuat 98](#_Toc133484536)

[V. Kiểm thử 101](#_Toc133484537)

[1. Thêm điện thoại, phiếu nhập: 101](#_Toc133484538)

[2. Thêm phiếu xuất, xóa điện thoại khi xuất hàng 103](#_Toc133484539)

[3. Hiển thị danh sách điện thoại 103](#_Toc133484540)

[4. Tìm kiếm điện thoại: 105](#_Toc133484541)

[ Theo mã điện thoại: 105](#_Toc133484542)

[ Theo tên: 105](#_Toc133484543)

[VI. Tài liệu tham khảo 106](#_Toc133484544)

# I. Mô Tả Bài Toán

Hằng năm , điện thoại nói riêng và các dòng điện tử nói chung ngày càng phát triển .Vì thế sẽ có hàng ngàn mẫu điện thoại và các thiết bị điện tử ngày càng ra đời , Số lượng các sản phẩm quá lớn đòi hỏi công tác quản lý sản phẩm phải ngày càng tối ưu và hiệu quả hơn, đặc biệt là về công tác quản lý kho. trở nên khó khăn . Vì quản lý kho đóng một vai trò quan trọng trong kinh doanh. Sự thiếu hụt hay dư thừa hàng tồn kho đều có ảnh hưởng rất lớn đến chi phí kinh doanh và kế hoạch phát triển của doanh nghiệp. Để hỗ trợ công tác quản lý kho hiệu quả, hiện nay trên thị trường có rất nhiều loại phần mềm quản lý kho , tiêu biểu có thể là Bota, Gtool , và từ những nền tảng đã được học và tìm hiểu nhóm chúng em đã lên ý tưởng thiết kế và phát triển đồ án về đề tài "Quản lý kho điện thoại cao cấp". Với đề tài này, nhóm chúng em mong rằng sẽ góp phần cải thiện được tính hiệu quả, tính nhanh chóng và nâng cao hiệu suất quản lý trong công tác quản lý kho điện thoại của các doanh nghiệp

Và điều mấu chốt là dễ dàng sử dụng- mục đích đề nhắm đến ai cũng có thể sử dụng được mà không phải là chỉ mỗi doanh nghiệp

Class Phone: sẽ có các dòng điện thoại cao cấp , tên sản phẩm , mẫu mã, hệ điều hành, số lượng, đơn giá từ đó sẽ có class con là IOS và Android , Class IOS kế thừa từ class Phone, Class Android kế thừa từ class Phone.

Class Phieu: sẽ có mã phiếu, mã sản phẩm, ngày nhập/xuất, nhà cung cấp và công ty. Class PhieuNhap, PhieuXuat kế thừa từ class Phieu.

# II.Xác định các lớp của bài toán

# Package quanlikho

## 1.1. Class Phone

* Thuộc tính:
* protected String ma;
* protected String ten;
* protected String heDieuHanh;
* protected String mau;
* protected String dungLuong;
* protected double soLuong;
* protected double donGia;
* Phương thức:
* Constructor: Hàm tạo có đối số và không có đối số
* Getter, Setter: phương thức get, set cho các thuộc tính kiểu public
* Nhap: nhập thông tin các thuộc tính
* Hien: hiện ra các thông tin

## 1.2. Class iPhone kế thừa từ class Phone

* Phương thức:
* Nhap: override ở class Phone
* Hien: override ở class Phone

## 1.3.Class Samsung kế thừa từ class Phone

* Phương thức:
* Nhap: override ở class Phone
* Hien: override ở class Phone

## 1.4. Class Xiaomi kế thừa từ class Phone

* Phương thức:
* Nhap: override ở class Phone
* Hien: override ở class Phone

## 1.5.Class Oppo kế thừa từ class Phone

* Phương thức:
* Nhap: override ở class Phone
* Hien: override ở class Phone

## 1.6.Class Phieu

* Thuộc tính:
* public String maPhieu;
* public String ma;
* Phương thức:
* Constructor: hàm tạo có đối số và không có đối số
* Nhap: nhập thông tin các thuộc tính
* Hien: hiện thông tin thuộc tính
* Getter, setter: phương thức get, set cho các thuộc tính kiểu public

## 1.7.Class PhieuNhap kế thừa từ class Phieu

* Thuộc tính:
* public String tenNcc;
* public String ngayNhap;
* Phương thức:
* Nhap: override ở class Phieu
* Hien: override ở class Phieu

## 1.8. Class PhieuXuat kế thừa từ class Phieu

* Thuộc tính:
* public String congTy;
* public String ngayXuat;
* Phương thức:
* Nhap: override ở class Phieu
* Hien: override ở class Phieu

## 1.9.Class Quanli

* Thuộc tính:
* Phone phone
* PhieuNhap pn
* PhieuXuat px
* Phương thức:

+ void Menu();

# Package DAO

## 2.1. Class DAOInterface

* Phương thức:

+ public int insert(T t);

+ public int update(T t);

+ public int delete(T t);

+ public ArrayList<T> selectAll();

+ public T selectById(T t);

+ public ArrayList<T> selectByTen(T t);

+ public ArrayList<T> selectBySoluongBetween();

+ public ArrayList<T> selectBySoluong1000();

+ public ArrayList<T> selectBySoluong5000();

+ public ArrayList<T> sapXepTangDan();

+ public ArrayList<T> sapXepGiamDan();

## 2.2. Class DAOiPhone implements class DAOInterface

* Phương thức:

+ public int insert(T t);

+ public int update(T t);

+ public int delete(T t);

+ public ArrayList<T> selectAll();

+ public T selectById(T t);

+ public ArrayList<T> selectByTen(T t);

+ public ArrayList<T> selectBySoluongBetween();

+ public ArrayList<T> selectBySoluong1000();

+ public ArrayList<T> selectBySoluong5000();

+ public ArrayList<T> sapXepTangDan();

+ public ArrayList<T> sapXepGiamDan();

## 2.3. Class DAOSamsung implements class DAOInterface

* Phương thức:

+ public int insert(T t);

+ public int update(T t);

+ public int delete(T t);

+ public ArrayList<T> selectAll();

+ public T selectById(T t);

+ public ArrayList<T> selectByTen(T t);

+ public ArrayList<T> selectBySoluongBetween();

+ public ArrayList<T> selectBySoluong1000();

+ public ArrayList<T> selectBySoluong5000();

+ public ArrayList<T> sapXepTangDan();

+ public ArrayList<T> sapXepGiamDan();

## 2.4. Class DAOOppo implements class DAOInterface

* Phương thức:

+ public int insert(T t);

+ public int update(T t);

+ public int delete(T t);

+ public ArrayList<T> selectAll();

+ public T selectById(T t);

+ public ArrayList<T> selectByTen(T t);

+ public ArrayList<T> selectBySoluongBetween();

+ public ArrayList<T> selectBySoluong1000();

+ public ArrayList<T> selectBySoluong5000();

+ public ArrayList<T> sapXepTangDan();

+ public ArrayList<T> sapXepGiamDan();

## 2.5. Class DAOXiaomi implements class DAOInterface

* Phương thức:

+ public int insert(T t);

+ public int update(T t);

+ public int delete(T t);

+ public ArrayList<T> selectAll();

+ public T selectById(T t);

+ public ArrayList<T> selectByTen(T t);

+ public ArrayList<T> selectBySoluongBetween();

+ public ArrayList<T> selectBySoluong1000();

+ public ArrayList<T> selectBySoluong5000();

+ public ArrayList<T> sapXepTangDan();

+ public ArrayList<T> sapXepGiamDan();

## 2.6.Class DAOPhieuInterface

* public void insert(T t);

## 2.7 Class DAOPhieuNhap implements class DAOPhieuInterface

* Phương thức:

+ public void insert(T t);

## 2.8 Class DAOPhieuXuat implements class DAOPhieuInterface

* Phương thức:

+ public void insert(T t);

## 

# III. Mô tả thuật toán

## Chỉnh sửa thông tin điện thoại

* Thêm: nhập một đối tượng mới vào cơ sở dữ liệu

Sử dụng câu lệnh SQL:

* INSERT INTO tên bảng

(cot1, cot2, cot3,...cotN)

VALUES (gia\_tri1, gia\_tri2, gia\_tri3,...gia\_triN);

* Xóa: xóa một đối tượng trong cơ sở dữ liệu

Sử dụng câu lệnh SQL:

* DELETE from tên bảng WHERE điều kiện;

## Hiển thị danh sách điện thoại hiện có trong cơ sở dữ liệu kho

* Hiển thị danh sách toàn bộ các đối tượng trong cơ sở dữ liệu:

+ Sử dụng câu lệnh SQL:

+SELECT \* FROM tên bảng

* Hiển thị danh sách theo số lượng 1000-5000

+ Sử dụng câu lệnh SQL:

+ SELECT \* FROM tên bảng WHERE SOLUONG BETWEEN 1000 AND 5000

* Hiển thị danh sách bé hơn 1000

+ Sử dụng câu lệnh SQL:

+ SELECT \* FROM tên bảng WHERE SOLUONG <1000

* Hiển thị danh sách lớn hơn 5000

+ Sử dụng câu lệnh SQL:

+ SELECT \* FROM tên bảng WHERE SOLUONG >5000

## Tìm kiếm

Tìm kiếm theo mã / tên

Sử dụng câu lệnh SQL:

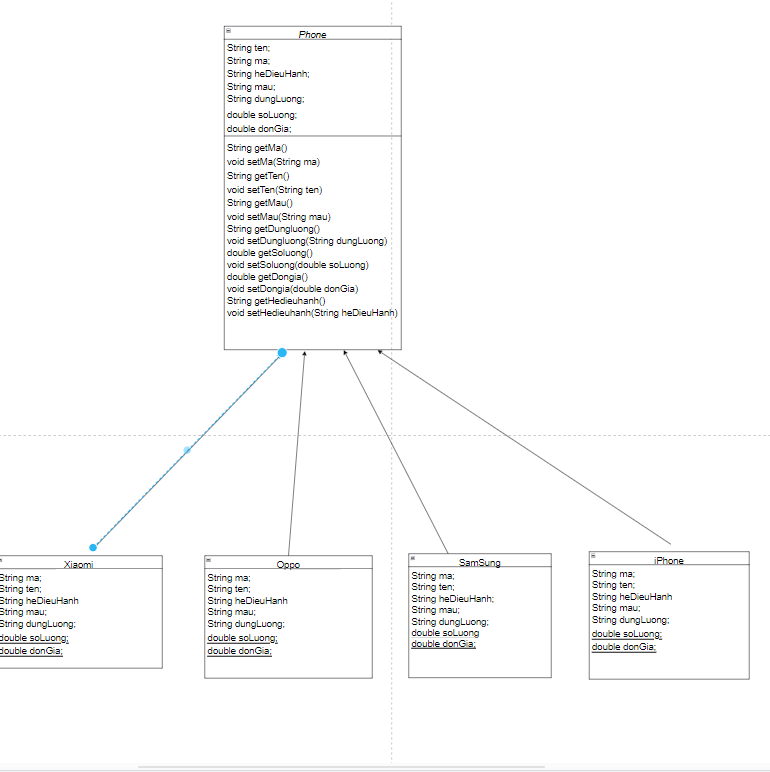
* SELECT \* FROM tên bảng WHERE điều kiện

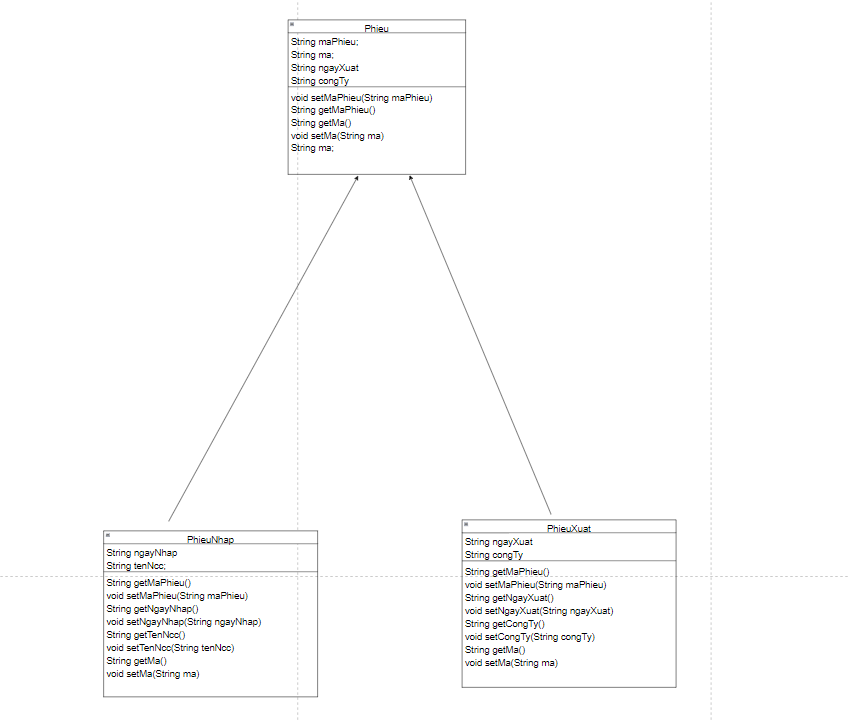
## Sắp xếp

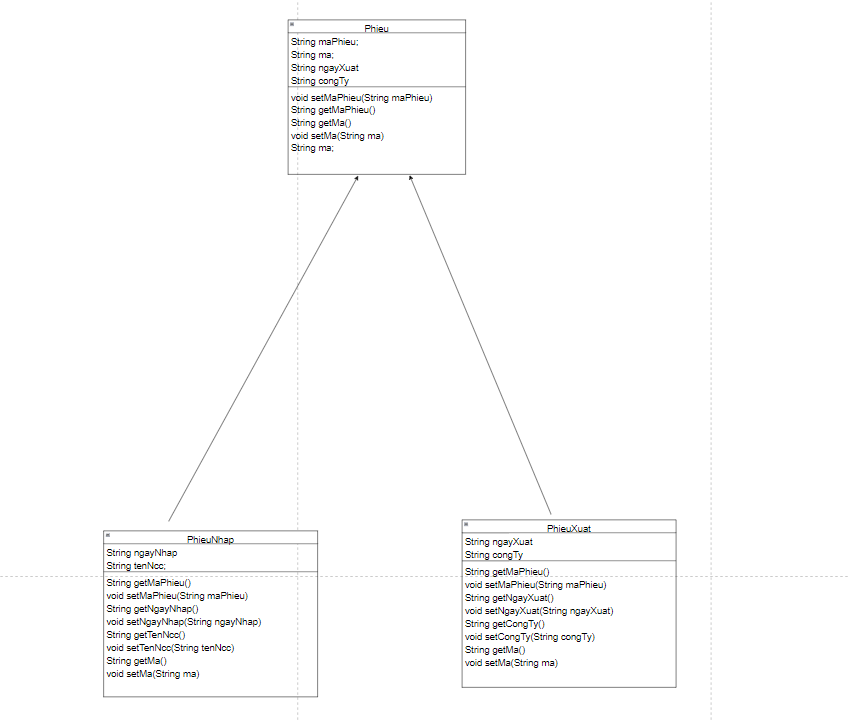
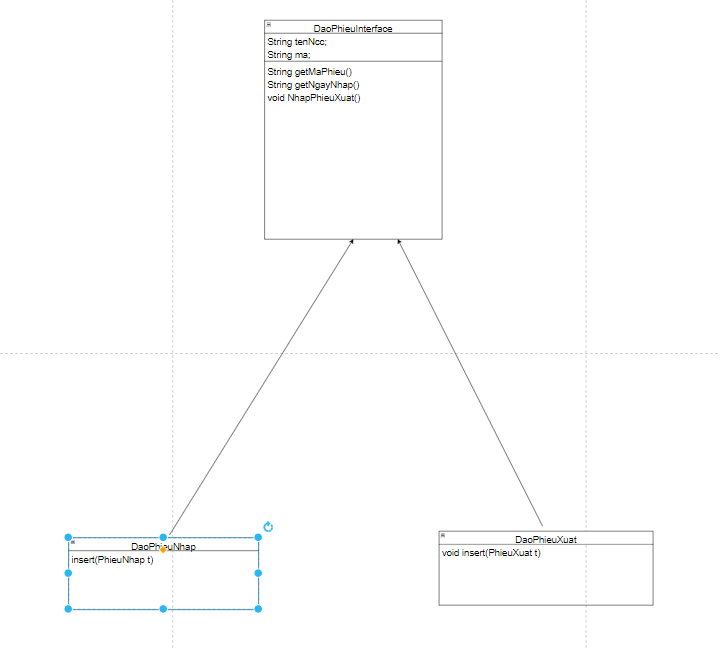
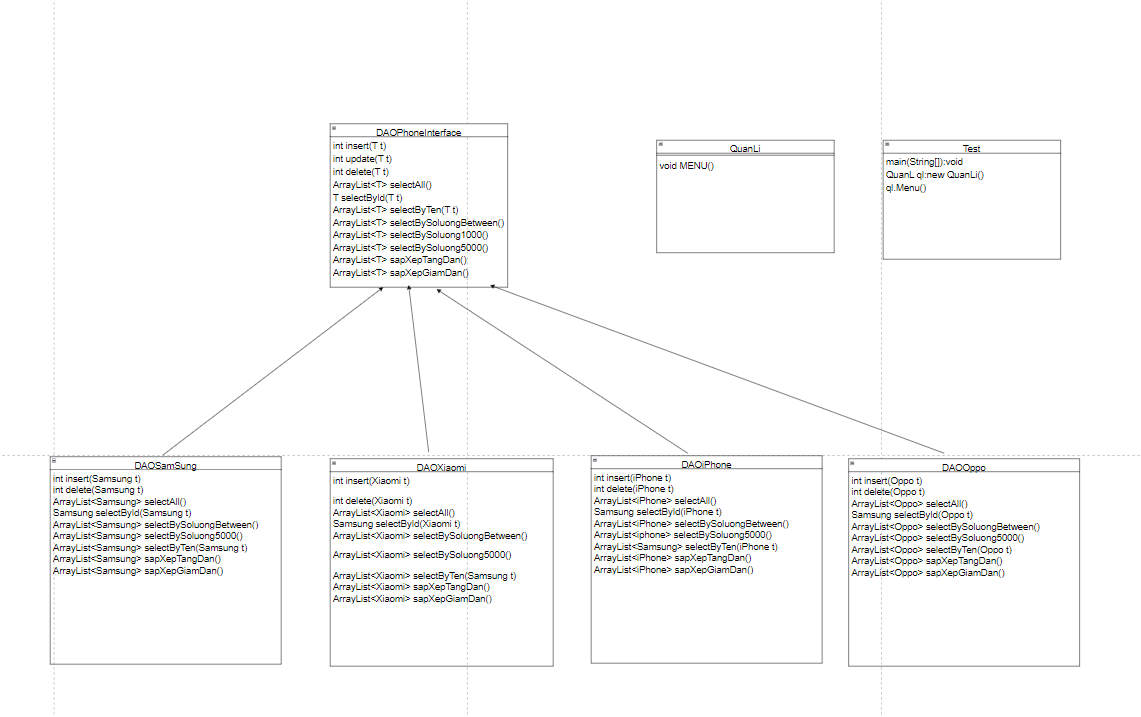
Sắp xếp số lượng tăng dần / giảm dần

# IV. Sơ đồ

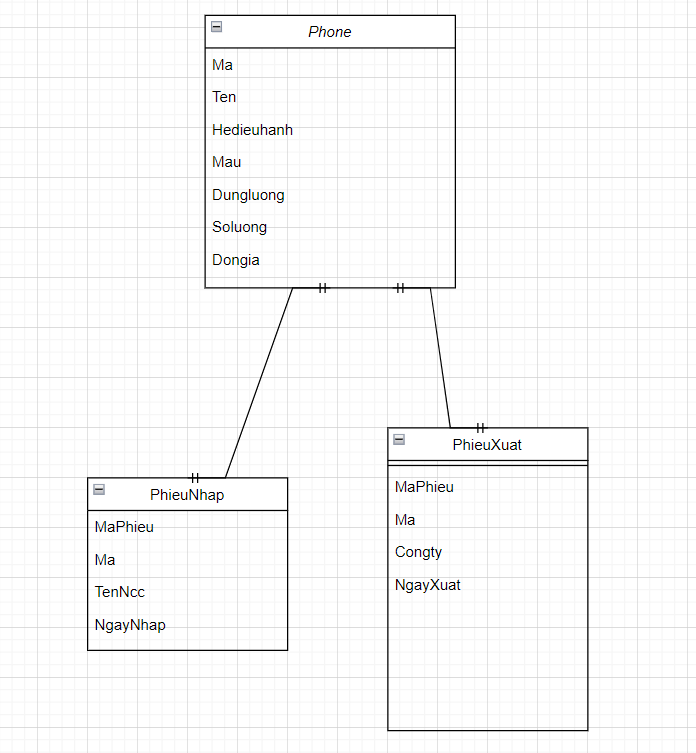
* + - 1. **Sơ đồ Class**







* + - 1. **Lược đồ CSDL**



# 

# IV. Cài đặt bài toán

# 1.Package quanlikho

## 1.1. Class iPhone

package quanlikho;

public class iPhone extends Phone

{

public iPhone()

{

super();

}

public iPhone(String ma, String ten, String heDieuHanh, String mau, String dungLuong, double soLuong, double donGia) {

super(ma, ten, heDieuHanh, mau, dungLuong, soLuong, donGia);

}

public void Hien() {

super.Hien(); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody

}

public void NhapPhone() {

super.NhapPhone(); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody

}

}

## 1.2. Class ****Samsung****

package quanlikho;

public class Samsung extends Phone

{

public Samsung() {

super();

}

public Samsung(String ma, String ten, String heDieuHanh, String mau, String dungLuong, double soLuong, double donGia) {

super(ma, ten, heDieuHanh, mau, dungLuong, soLuong, donGia);

}

@Override

public void Hien() {

super.Hien(); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody

}

@Override

public void NhapPhone() {

super.NhapPhone(); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody

}

## 1.3. Class Android

package quanlikho;

public class Android extends Phone {

public Android() {

}

public Android(String ma, String ten, String heDieuHanh, String mau, String dungLuong, double soLuong, double donGia) {

super(ma, ten, heDieuHanh, mau, dungLuong, soLuong, donGia);

}

@Override

public void NhapPhone() {

super.NhapPhone();

}

@Override

public void Hien() {

super.Hien(); }

}

## 1.4. Class ****Xiaomi****

package quanlikho;

public class Xiaomi extends Phone

{

public Xiaomi() {

super();

}

public Xiaomi(String ma, String ten, String heDieuHanh, String mau, String dungLuong, double soLuong, double donGia) {

super(ma, ten, heDieuHanh, mau, dungLuong, soLuong, donGia);

}

@Override

public void Hien() {

super.Hien(); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody

}

@Override

public void NhapPhone() {

super.NhapPhone(); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody

}

}

## 1.5. Class PhieuNhap

package quanlikho;

import java.util.Scanner;

public class PhieuNhap extends Phieu

{ public String ngayNhap;

public String tenNcc;

public PhieuNhap(String ngayNhap, String tenNcc, String maPhieu, String ma) {

super(maPhieu, ma);

this.ngayNhap = ngayNhap;

this.tenNcc = tenNcc;

}

public PhieuNhap() {

}

public String getMaPhieu() {

return maPhieu;

}

public void setMaPhieu(String maPhieu) {

this.maPhieu = maPhieu;

}

public String getNgayNhap() {

return ngayNhap;

}

public void setNgayNhap(String ngayNhap) {

this.ngayNhap = ngayNhap;

}

public String getTenNcc() {

return tenNcc;

}

public void setTenNcc(String tenNcc) {

this.tenNcc = tenNcc;

}

public String getMa() {

return ma;

}

public void setMa(String ma) {

this.ma = ma;

}

}

## 1.6.Class PhieuXuat

package quanlikho;

public class PhieuXuat extends Phieu

{ String ngayXuat;

String congTy;

public PhieuXuat(String ngayXuat, String congTy, String maPhieu, String ma) {

super(maPhieu, ma);

this.ngayXuat = ngayXuat;

this.congTy = congTy;

}

public PhieuXuat() {

}

public String getMaPhieu() {

return maPhieu;

}

public void setMaPhieu(String maPhieu) {

this.maPhieu = maPhieu;

}

public String getNgayXuat() {

return ngayXuat;

}

public void setNgayXuat(String ngayXuat) {

this.ngayXuat = ngayXuat;

}

public String getCongTy() {

return congTy;

}

public void setCongTy(String congTy) {

this.congTy = congTy;

}

public String getMa() {

return ma;

}

public void setMa(String ma) {

this.ma = ma;

}

}

## 1.7. Class Quanli

package quanlikho;

import DAO.DAOOppo;

import DAO.DAOPhieuNhap;

import DAO.DAOPhieuXuat;

import java.util.ArrayList;

import java.util.Scanner;

import DAO.DAOSamsung;

import DAO.DAOXiaomi;

import DAO.DAOiPhone;

/\*\*

\*

\* @author thanh ha

\*/

public class QuanLi

{

public void MENU()

{

Scanner input = new Scanner(System.in);

iPhone ip;

Samsung ss;

Oppo op;

Xiaomi xm;

PhieuNhap phieuNhap;

PhieuXuat phieuXuat;

int chon;

do {

System.out.println("\n+---------------------MENU QUAN LY KHO DIEN THOAI CAO CAP---------------------------+");

System.out.println("1. Nhap dien thoai");

System.out.println("2. Xuat dien thoai");

System.out.println("3. Hien thi danh sach dien thoai trong kho");

System.out.println("4. Tim kiem");

System.out.println("5. Sap xep");

System.out.println("+------------------------------------------------------------------------------------+");

System.out.print("\n Chon chuc nang: ");

chon = input.nextInt();

if (chon == 1) {

System.out.println("\n+---------NHAP DIEN THOAI---------+");

System.out.println("1. Nhap iPhone");

System.out.println("2. Nhap Samsung");

System.out.println("3. Nhap Oppo");

System.out.println("4. Nhap Xiaomi");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

DAOiPhone dp = new DAOiPhone();

ip = new iPhone();

DAOPhieuNhap dpn = new DAOPhieuNhap();

phieuNhap = new PhieuNhap();

System.out.print("Nhap so lan nhap: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan nhap thu "+i+" ---");

dpn.insert(phieuNhap);

dp.insert(ip);

}

}

else if (chon==2)

{

DAOSamsung ds = new DAOSamsung();

ss = new Samsung();

DAOPhieuNhap dpn = new DAOPhieuNhap();

phieuNhap = new PhieuNhap();

System.out.print("Nhap so lan nhap: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan nhap thu "+i+" ---");

dpn.insert(phieuNhap);

ds.insert(ss);

}

}

else if(chon==3)

{

DAOOppo dop = new DAOOppo();

op = new Oppo();

DAOPhieuNhap dpn = new DAOPhieuNhap();

phieuNhap = new PhieuNhap();

System.out.print("Nhap so lan nhap: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan nhap thu "+i+" ---");

dpn.insert(phieuNhap);

dop.insert(op);

}

}

else if (chon==4)

{

DAOXiaomi dxm = new DAOXiaomi();

xm = new Xiaomi();

DAOPhieuNhap dpn = new DAOPhieuNhap();

phieuNhap = new PhieuNhap();

System.out.print("Nhap so lan nhap: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan nhap thu "+i+" ---");

dpn.insert(phieuNhap);

dxm.insert(xm);

}

}

}

else if (chon==2)

{

System.out.println("\n+---------XUAT DIEN THOAI---------+");

System.out.println("1.Xuat iPHONE");

System.out.println("2.Xuat SAMSUNG");

System.out.println("3.Xuat OPPO");

System.out.println("4.Xuat XIAOMI");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ip = new iPhone();

DAOiPhone dp = new DAOiPhone();

phieuXuat = new PhieuXuat();

DAOPhieuXuat dpx = new DAOPhieuXuat();

System.out.print("Nhap so lan xuat san pham: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan thu "+i+" ---");

dpx.insert(phieuXuat);

dp.delete(ip);

}

}

else if (chon==2)

{

ss = new Samsung();

DAOSamsung ds = new DAOSamsung();

phieuXuat = new PhieuXuat();

DAOPhieuXuat dpx = new DAOPhieuXuat();

System.out.print("Nhap so lan xuat san pham: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan thu "+i+" ---");

dpx.insert(phieuXuat);

ds.delete(ss);

}

}

else if (chon==3)

{

op = new Oppo();

DAOOppo dop = new DAOOppo();

phieuXuat = new PhieuXuat();

DAOPhieuXuat dpx = new DAOPhieuXuat();

System.out.print("Nhap so lan xuat san pham: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan thu "+i+" ---");

dpx.insert(phieuXuat);

dop.delete(op);

}

}

else if (chon==4)

{

xm = new Xiaomi();

DAOXiaomi dxm = new DAOXiaomi();

phieuXuat = new PhieuXuat();

DAOPhieuXuat dpx = new DAOPhieuXuat();

System.out.print("Nhap so lan xuat san pham: ");

int n = input.nextInt();

for (int i=1; i<=n;i++)

{ System.out.println("---Lan thu "+i+" ---");

dpx.insert(phieuXuat);

dxm.delete(xm);

}

}

}

else if (chon == 3) {

System.out.println("+---------HIEN THI DANH SACH TRONG KHO-------------+");

System.out.println("1.Hien thi danh sach dien thoai iPHONE");

System.out.println("2.Hien thi danh sach dien thoai Samsung");

System.out.println("3.Hien thi danh sach dien thoai Oppo");

System.out.println("4.Hien thi danh sach dien thoai Xiaomi");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if(chon==1)

{ System.out.println("+---------HIEN THI DANH SACH iPHONE---------+");

System.out.println("1.Hien thi danh sach tat ca");

System.out.println("2.Hien thi danh sach so luong tu 1000 - 5000");

System.out.println("3.Hien thi danh sach so luong < 1000");

System.out.println("4.Hien thi danh sach so luong > 5000");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<iPhone> list = DAOiPhone.getInstance().selectAll();

for (iPhone p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<iPhone> list = DAOiPhone.getInstance().selectBySoluongBetween();

for (iPhone p : list)

{

p.Hien();

}

}

else if (chon==3)

{

ArrayList<iPhone> list = DAOiPhone.getInstance().selectBySoluong1000();

for (iPhone p : list)

{

p.Hien();

}

}

else if (chon==4)

{

ArrayList<iPhone> list = DAOiPhone.getInstance().selectBySoluong5000();

for (iPhone p : list)

{

p.Hien();

}

}

}

else if(chon==2)

{

System.out.println("+---------HIEN THI DANH SACH SAMSUNG---------+");

System.out.println("1.Hien thi danh sach tat ca");

System.out.println("2.Hien thi danh sach so luong tu 1000 - 5000");

System.out.println("3.Hien thi danh sach so luong < 1000");

System.out.println("4.Hien thi danh sach so luong > 5000");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<Samsung> list = DAOSamsung.getInstance().selectAll();

for (Samsung p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<Samsung> list = DAOSamsung.getInstance().selectBySoluongBetween();

for (Samsung p : list)

{

p.Hien();

}

}

else if (chon==3)

{

ArrayList<Samsung> list = DAOSamsung.getInstance().selectBySoluong1000();

for (Samsung p : list)

{

p.Hien();

}

}

else if (chon==4)

{

ArrayList<Samsung> list = DAOSamsung.getInstance().selectBySoluong5000();

for (Samsung p : list)

{

p.Hien();

}

}

}

else if (chon==3)

{

System.out.println("+---------HIEN THI DANH SACH OPPO---------+");

System.out.println("1.Hien thi danh sach tat ca");

System.out.println("2.Hien thi danh sach so luong tu 1000 - 5000");

System.out.println("3.Hien thi danh sach so luong < 1000");

System.out.println("4.Hien thi danh sach so luong > 5000");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<Oppo> list = DAOOppo.getInstance().selectAll();

for (Oppo p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<Oppo> list = DAOOppo.getInstance().selectBySoluong1000();

for (Oppo p : list)

{

p.Hien();

}

}

else if (chon==3)

{

ArrayList<Oppo> list = DAOOppo.getInstance().selectBySoluong5000();

for (Oppo p : list)

{

p.Hien();

}

}

}

else if(chon==4)

{

System.out.println("+---------HIEN THI DANH SACH XIAOMI---------+");

System.out.println("1.Hien thi danh sach tat ca");

System.out.println("2.Hien thi danh sach so luong tu 1000 - 5000");

System.out.println("3.Hien thi danh sach so luong < 1000");

System.out.println("4.Hien thi danh sach so luong > 5000");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<Xiaomi> list = DAOXiaomi.getInstance().selectAll();

for (Xiaomi p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<Xiaomi> list = DAOXiaomi.getInstance().selectBySoluong1000();

for (Xiaomi p : list)

{

p.Hien();

}

}

else if (chon==3)

{

ArrayList<Xiaomi> list = DAOXiaomi.getInstance().selectBySoluong5000();

for (Xiaomi p : list)

{

p.Hien();

}

}

}

}

else if (chon == 4) {

System.out.println("+---------TIM KIEM DIEN THOAI TRONG KHO---------+");

System.out.println("1.Tim kiem iPHONE");

System.out.println("2.Tim kiem Samsung");

System.out.println("3.Tim kiem Oppo");

System.out.println("4.Tim kiem Xiaomi");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{ System.out.println("+---------TIM KIEM iPHONE---------+");

System.out.println("1.Tim kiem theo ma");

System.out.println("2.Tim kiem theo ten");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

input.nextLine();

DAOiPhone dp = new DAOiPhone();

ip = new iPhone();

dp.selectById(ip);

}

else if (chon==2)

{

ip = new iPhone();

ArrayList<iPhone> list = DAOiPhone.getInstance().selectByTen(ip);

for (iPhone p : list)

{

p.Hien();

}

}

}

else if (chon==2)

{

System.out.println("+---------TIM KIEM SAMSUNG---------+");

System.out.println("1.Tim kiem theo ma");

System.out.println("2.Tim kiem theo ten");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

input.nextLine();

DAOSamsung ds = new DAOSamsung();

ss = new Samsung();

ds.selectAll();

}

else if (chon==2)

{

ss = new Samsung();

ArrayList<Samsung> list = DAOSamsung.getInstance().selectByTen(ss);

for (Samsung p : list)

{

p.Hien();

}

}

}

else if (chon==3)

{

System.out.println("+---------TIM KIEM OPPO---------+");

System.out.println("1.Tim kiem theo ma");

System.out.println("2.Tim kiem theo ten");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

input.nextLine();

DAOOppo dop = new DAOOppo();

op = new Oppo();

dop.selectAll();

}

else if (chon==2)

{

op = new Oppo();

ArrayList<Oppo> list = DAOOppo.getInstance().selectByTen(op);

for (Oppo p : list)

{

p.Hien();

}

}

}

else if (chon==4)

{

System.out.println("+---------TIM KIEM XIAOMI---------+");

System.out.println("1.Tim kiem theo ma");

System.out.println("2.Tim kiem theo ten");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

input.nextLine();

DAOXiaomi dxm = new DAOXiaomi();

xm = new Xiaomi();

dxm.selectAll();

}

else if (chon==2)

{

xm = new Xiaomi();

ArrayList<Xiaomi> list = DAOXiaomi.getInstance().selectByTen(xm);

for (Xiaomi p : list)

{

p.Hien();

}

}

}

}

else if (chon==5)

{ System.out.println("+---------SAP XEP ---------+");

System.out.println("1.Sap xep iPHONE");

System.out.println("2.Sap xep Samsung");

System.out.println("3.Sap xep Oppo");

System.out.println("4.Sap xep Xiaomi");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

System.out.println("+---------SAP XEP iPHONE ---------+");

System.out.println("1.Sap xep so luong tang dan");

System.out.println("2.Sap xep so luong giam dan");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<iPhone> list = DAOiPhone.getInstance().sapXepTangDan();

for (iPhone p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<iPhone> list = DAOiPhone.getInstance().sapXepGiamDan();

for (iPhone p : list)

{

p.Hien();

}

}

}

else if (chon==2)

{

System.out.println("+---------SAP XEP SAMSUNG----------+");

System.out.println("1.Sap xep so luong tang dan");

System.out.println("2.Sap xep so luong giam dan");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<Samsung> list = DAOSamsung.getInstance().sapXepTangDan();

for (Samsung p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<Samsung> list = DAOSamsung.getInstance().sapXepGiamDan();

for (Samsung p : list)

{

p.Hien();

}

}

}

else if (chon==3)

{

System.out.println("+---------SAP XEP OPPO----------+");

System.out.println("1.Sap xep so luong tang dan");

System.out.println("2.Sap xep so luong giam dan");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<Oppo> list = DAOOppo.getInstance().sapXepTangDan();

for (Oppo p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<Oppo> list = DAOOppo.getInstance().sapXepGiamDan();

for (Oppo p : list)

{

p.Hien();

}

}

}

else if (chon==4)

{

System.out.println("+---------SAP XEP XIAOMI----------+");

System.out.println("1.Sap xep so luong tang dan");

System.out.println("2.Sap xep so luong giam dan");

System.out.print("Vui long chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

ArrayList<Xiaomi> list = DAOXiaomi.getInstance().sapXepTangDan();

for (Xiaomi p : list)

{

p.Hien();

}

}

else if (chon==2)

{

ArrayList<Xiaomi> list = DAOXiaomi.getInstance().sapXepGiamDan();

for (Xiaomi p : list)

{

p.Hien();

}

}

}

}

} while (chon!= 0);

}

}

package quanlikho;

import DAO.DAOPhone;

import java.util.ArrayList;

import java.util.Scanner;

import DAO.DAOPhieu;

/\*\*

\*

\* @author thanh ha

\*/

public class QuanLi

{

public void MENU()

{

Scanner input = new Scanner(System.in);

Phone phone;

Phieu phieuNhap;

Phieu phieuXuat;

int chon;

do {

System.out.println("+---------------------MENU QUAN LY KHO DIEN THOAI CAO CAP---------------------------+");

System.out.println("1. Phieu Nhap");

System.out.println("2. Phieu Xuat");

System.out.println("3. Hien thi danh sach trong kho");

System.out.println("4. Tim kiem");

System.out.println("5. Xoa");

System.out.println("6. Sap xep");

System.out.println("+------------------------------------------------------------------------------------+");

System.out.print("\n Chon chuc nang: ");

chon = input.nextInt();

if (chon == 1) {

System.out.println("\n+---------PHIEU NHAP DIEN THOAI---------+");

DAOPhieu dpn = new DAOPhieu();

DAOPhone dp = new DAOPhone();

phieuNhap = new Phieu();

phone = new Phone();

phieuNhap.NhapPhieuNhap();

phone.NhapPhone();

dpn.insertPN(phieuNhap);

dp.insert(phone);

}

else if (chon==2)

{

System.out.println("\n+---------PHIEU XUAT DIEN THOAI---------+");

//Xuất hàng thì phải xóa hàng

DAOPhieu dpx = new DAOPhieu();

DAOPhone dp = new DAOPhone();

phieuXuat = new Phieu();

phone = new Phone();

phieuXuat.NhapPhieuXuat();

phone.NhapPhone();

dpx.insert(phieuXuat);

dp.insertPhoneXuat(phone);

}

else if (chon == 3) {

System.out.println("+---------HIEN THI DANH SACH TRONG KHO---------+");

System.out.println("1.Hien thi danh sach dien thoai nhap");

System.out.println("2.Hien thi danh sach dien thoai xuat");

System.out.println("3.Hien thi danh sach phieu nhap");

System.out.println("4.Hien thi thong tin phieu xuat");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if(chon==1)

{

ArrayList<Phone> list = DAOPhone.getInstance().selectAll();

for (Phone p : list)

{

p.Hien();

}

}

else if(chon==2)

{

ArrayList<Phone> list = DAOPhone.getInstance().selectAllPhoneXuat();

for (Phone p : list)

{

p.Hien();

}

}

else if (chon==3)

{

DAOPhieu dpn = new DAOPhieu();

dpn.selectAllPN();

}

else if(chon==4)

{

DAOPhieu dpx = new DAOPhieu();

dpx.selectAll();

}

}

else if (chon == 4) {

System.out.println("+---------TIM KIEM DIEN THOAI TRONG KHO---------+");

System.out.println("1.Tim kiem theo ma");

System.out.println("2.Tim kiem theo ten");

System.out.println("Chon chuc nang: ");

chon = input.nextInt();

if (chon==1)

{

input.nextLine();

Phone find = new Phone();

System.out.println("Nhap ma can tim: ");

String ma = input.nextLine();

find.setMa(ma);

Phone ip2 = DAOPhone.getInstance().selectById(find);

ip2.Hien();

}

else if (chon==2)

{

input.nextLine();

Phone find = new Phone();

System.out.println("Nhap ten can tim: ");

String ten = input.nextLine();

find.setTen(ten);

Phone ip2 = DAOPhone.getInstance().selectByTen(find);

ip2.Hien();

}

}

else if (chon==5)

{ System.out.println("+--------------XOA-------------+");

phone = new Phone();

DAOPhone dp = new DAOPhone();

dp.delete(phone);

}

else if (chon==6)

{ }

} while (chon!= 0);

}

}

## 1.7. Class Main

package quanlikho;

public class Main {

public static void main(String[] args)

{

QuanLi ql = new QuanLi();

ql.MENU();

}

}

# 2. Package DAO

## 2.1. Class DAOInterface

package DAO;

import java.util.ArrayList;

public interface DAOInterface <T>

{

public int insert(T t);

public int update(T t);

public int delete(T t);

public ArrayList<T> selectAll();

public T selectById(T t);

public ArrayList<T> selectByTen(T t);

public ArrayList<T> selectBySoluongBetween();

public ArrayList<T> selectBySoluong1000();

public ArrayList<T> selectBySoluong5000();

public ArrayList<T> sapXepTangDan();

public ArrayList<T> sapXepGiamDan();

}

## 2.2. Class DAOiPhone

package DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

import quanlikho.Phone;

import quanlikho.iPhone;

/\*\*

\*

\* @author thanh ha

\*/

public class DAOiPhone implements DAOInterface <iPhone>

{

public static DAOiPhone getInstance()

{

return new DAOiPhone();

}

@Override

public int insert(iPhone t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

Scanner input = new Scanner(System.in);

iPhone ip = new iPhone();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

ResultSet rs = stm.executeQuery("SELECT \* FROM iPHONE");

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten san pham: ");

String ten = input.nextLine();

System.out.println("Nhap he dieu hanh: ");

String heDieuHanh = input.nextLine();

System.out.println("Nhap mau: ");

String mau = input.nextLine();

System.out.println("Nhap dung luong: ");

String dungLuong = input.nextLine();

System.out.println("Nhap so luong: ");

Double soLuong = input.nextDouble();

System.out.println("Nhap don gia: ");

Double donGia = input.nextDouble();

String sqlInsert = "INSERT INTO iPHONE (MA, TEN, HEDIEUHANH, MAU, DUNGLUONG, SOLUONG, DONGIA)"

+ "VALUES ('" + ma

+ "' , '" + ten

+ "' , '" + heDieuHanh

+ "' , '" + mau

+ "' , '" + dungLuong

+ "' , '" + soLuong

+ "' , '" + donGia + "')";

rows = stm.executeUpdate(sqlInsert);

System.out.println("----INSERT DIEN THOAI THANH CONG!----");

connection.close();

rs.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public int update(iPhone t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

Scanner input = new Scanner (System.in);

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten san pham: ");

String ten = input.nextLine();

System.out.println("Nhap he dieu hanh: ");

String heDieuHanh = input.nextLine();

System.out.println("Nhap mau: ");

String mau = input.nextLine();

System.out.println("Nhap dung luong: ");

String dungLuong = input.nextLine();

Double soLuong;

do{

System.out.println("Nhap so luong: ");

soLuong = input.nextDouble();

}while (soLuong<0);

Double donGia;

do{

System.out.println("Nhap don gia: ");

donGia = input.nextDouble();

} while(donGia<0);

String sqlUpdate = "UPDATE iPHONE SET TEN=?, HEDIEUHANH=?, MAU=?, DUNGLUONG=?, SOLUONG=?, DONGIA=? WHERE MA=?";

rows = stm.executeUpdate(sqlUpdate);

System.out.println("----UPDATE THANH CONG!----");

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println("Nhap du lieu chua chinh xac ");

}

return rows;

}

@Override

public int delete(iPhone t) {

Scanner input = new Scanner (System.in);

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap ma iPhone can xuat: ");

String ma = input.nextLine();

String sqlInsert = "DELETE from iPHONE "

+ " WHERE MA='" + ma + "'";

rows = stm.executeUpdate(sqlInsert);

System.out.println("XUAT DIEN THOAI THANH CONG!");

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public ArrayList<iPhone> selectAll() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public iPhone selectById(iPhone t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

iPhone rows = null;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

Scanner input = new Scanner(System.in);

System.out.println("Nhap ma can tim: ");

String ma = input.nextLine();

String selectSql = "SELECT\* FROM iPHONE WHERE MA='"+ma+"'";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

rows = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

System.out.format("%-15s %-40s %-20s %-15s %-15s %-15s %-15s\n",

"MA SP",

"TEN",

"HE DIEU HANH",

"MAU",

"DUNG LUONG",

"SO LUONG",

"DON GIA"

);

System.out.format("%-15s %-40s %-20s %-15s %-15s %-15s %-15s\n",

ma,

ten,

heDieuHanh,

mau,

dungLuong,

soLuong,

donGia

);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");;

}

return rows;

}

@Override

public ArrayList<iPhone> selectBySoluongBetween() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE WHERE SOLUONG BETWEEN 1000 AND 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<iPhone> selectBySoluong1000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE WHERE SOLUONG < 1000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<iPhone> selectBySoluong5000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE WHERE SOLUONG > 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<iPhone> selectByTen(iPhone t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

Scanner input = new Scanner(System.in);

System.out.println("Nhap ten can tim: ");

String ten = input.nextLine();

String selectSql = "SELECT\* FROM iPHONE WHERE TEN='"+ten+"'";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<iPhone> sapXepTangDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE ORDER BY SOLUONG ASC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<iPhone> sapXepGiamDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <iPhone> rows = new ArrayList<iPhone>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM iPHONE ORDER BY SOLUONG DESC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

iPhone ip = new iPhone(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ip);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

}

## 2.3. Class ****DAOSamsung****

package DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

import quanlikho.Samsung;

import quanlikho.iPhone;

/\*\*

\*

\* @author thanh ha

\*/

public class DAOSamsung implements DAOInterface <Samsung> {

public static DAOSamsung getInstance()

{

return new DAOSamsung();

}

@Override

public int insert(Samsung t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

Scanner input = new Scanner(System.in);

Samsung ss = new Samsung();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

ResultSet rs = stm.executeQuery("SELECT \* FROM SAMSUNG");

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten san pham: ");

String ten = input.nextLine();

System.out.println("Nhap he dieu hanh: ");

String heDieuHanh = input.nextLine();

System.out.println("Nhap mau: ");

String mau = input.nextLine();

System.out.println("Nhap dung luong: ");

String dungLuong = input.nextLine();

Double soLuong;

do{

System.out.println("Nhap so luong: ");

soLuong = input.nextDouble();

}while (soLuong<0);

Double donGia;

do{

System.out.println("Nhap don gia: ");

donGia = input.nextDouble();

} while(donGia<0);

String sqlInsert = "INSERT INTO SAMSUNG (MA, TEN, HEDIEUHANH, MAU, DUNGLUONG, SOLUONG, DONGIA)"

+ "VALUES ('" + ma

+ "' , '" + ten

+ "' , '" + heDieuHanh

+ "' , '" + mau

+ "' , '" + dungLuong

+ "' , '" + soLuong

+ "' , '" + donGia + "')";

rows = stm.executeUpdate(sqlInsert);

System.out.println("----INSERT DIEN THOAI THANH CONG!----");

connection.close();

rs.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public int update(Samsung t) {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public int delete(Samsung t) {

Scanner input = new Scanner (System.in);

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap so lan muon xuat san pham: ");

int n = input.nextInt();

input.nextLine();

for (int i=1; i<=n; i++)

{ System.out.println("-----Lan xoa thu " +i+"-----");

System.out.println("Nhap ma Samsung: ");

String ma = input.nextLine();

String sqlInsert = "DELETE from SAMSUNG "

+ " WHERE MA='" + ma + "'";

rows = stm.executeUpdate(sqlInsert);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public ArrayList<Samsung> selectAll() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public Samsung selectById(Samsung t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

Samsung rows = null;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

Scanner input = new Scanner(System.in);

System.out.println("Nhap ma can tim: ");

String ma = input.nextLine();

String selectSql = "SELECT\* FROM iPHONE WHERE MA='"+ma+"'";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

rows = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

System.out.format("%-15s %-40s %-20s %-15s %-15s %-15s %-15s\n",

"MA SP",

"TEN",

"HE DIEU HANH",

"MAU",

"DUNG LUONG",

"SO LUONG",

"DON GIA"

);

System.out.format("%-15s %-40s %-20s %-15s %-15s %-15s %-15s\n",

ma,

ten,

heDieuHanh,

mau,

dungLuong,

soLuong,

donGia

);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");;

}

return rows;

}

@Override

public ArrayList<Samsung> selectBySoluongBetween() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG WHERE SOLUONG BETWEEN 1000 AND 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Samsung> selectBySoluong1000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG WHERE SOLUONG < 1000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Samsung> selectBySoluong5000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG WHERE SOLUONG > 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Samsung> selectByTen(Samsung t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

Scanner input = new Scanner(System.in);

System.out.println("Nhap ten can tim: ");

String ten = input.nextLine();

String selectSql = "SELECT\* FROM SAMSUNG WHERE TEN='"+ten+"'";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Samsung> sapXepTangDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG ORDER BY SOLUONG ASC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Samsung> sapXepGiamDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Samsung> rows = new ArrayList<Samsung>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM SAMSUNG ORDER BY SOLUONG DESC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Samsung ss = new Samsung(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(ss);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

}

## 2.4. Class ****DAOXiaomi****

package DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

import quanlikho.Oppo;

import quanlikho.Samsung;

import quanlikho.Xiaomi;

import quanlikho.iPhone;

/\*\*

\*

\* @author thanh ha

\*/

public class DAOXiaomi implements DAOInterface <Xiaomi>

{

public static DAOXiaomi getInstance()

{

return new DAOXiaomi();

}

@Override

public int insert(Xiaomi t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

Scanner input = new Scanner(System.in);

Xiaomi xm = new Xiaomi();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

ResultSet rs = stm.executeQuery("SELECT \* FROM XIAOMI");

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten san pham: ");

String ten = input.nextLine();

System.out.println("Nhap he dieu hanh: ");

String heDieuHanh = input.nextLine();

System.out.println("Nhap mau: ");

String mau = input.nextLine();

System.out.println("Nhap dung luong: ");

String dungLuong = input.nextLine();

Double soLuong;

do{

System.out.println("Nhap so luong: ");

soLuong = input.nextDouble();

}while (soLuong<0);

Double donGia;

do{

System.out.println("Nhap don gia: ");

donGia = input.nextDouble();

} while(donGia<0);

String sqlInsert = "INSERT INTO XIAOMI (MA, TEN, HEDIEUHANH, MAU, DUNGLUONG, SOLUONG, DONGIA)"

+ "VALUES ('" + ma

+ "' , '" + ten

+ "' , '" + heDieuHanh

+ "' , '" + mau

+ "' , '" + dungLuong

+ "' , '" + soLuong

+ "' , '" + donGia + "')";

rows = stm.executeUpdate(sqlInsert);

System.out.println("----INSERT DIEN THOAI THANH CONG!----");

connection.close();

rs.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public int update(Xiaomi t) {

Scanner input = new Scanner (System.in);

System.out.println("Nhap ma Xiaomi can xoa: ");

String ma = input.nextLine();

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI";

ResultSet rs = stm.executeQuery(selectSql);

String sqlInsert ="DELETE from XIAOMI "+

" WHERE MA='"+ma+"'";

rows = stm.executeUpdate(sqlInsert);

System.out.println("DELETE DIEN THOAI THANH CONG!");

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public int delete(Xiaomi t) {

Scanner input = new Scanner (System.in);

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap so lan muon xuat san pham: ");

int n = input.nextInt();

input.nextLine();

for (int i=1; i<=n; i++)

{ System.out.println("-----Lan xoa thu " +i+"-----");

System.out.println("Nhap ma Xiaomi: ");

String ma = input.nextLine();

String sqlInsert = "DELETE from XIAOMI "

+ " WHERE MA='" + ma + "'";

rows = stm.executeUpdate(sqlInsert);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public ArrayList<Xiaomi> selectAll() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public Xiaomi selectById(Xiaomi t) {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public ArrayList<Xiaomi> selectBySoluongBetween() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI WHERE SOLUONG BETWEEN 1000 AND 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Xiaomi> selectBySoluong1000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI WHERE SOLUONG < 1000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Xiaomi> selectBySoluong5000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI WHERE SOLUONG > 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Xiaomi> selectByTen(Xiaomi t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

Scanner input = new Scanner(System.in);

System.out.println("Nhap ten can tim: ");

String ten = input.nextLine();

String selectSql = "SELECT\* FROM XIAOMI WHERE TEN='"+ten+"'";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Xiaomi> sapXepTangDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI ORDER BY SOLUONG ASC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Xiaomi> sapXepGiamDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Xiaomi> rows = new ArrayList<Xiaomi>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM XIAOMI ORDER BY SOLUONG DESC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Xiaomi xm = new Xiaomi(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(xm);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

}

## 2.5. Class ****DAOOppo****

package DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

import quanlikho.Oppo;

import quanlikho.Samsung;

import quanlikho.iPhone;

/\*\*

\*

\* @author thanh ha

\*/

public class DAOOppo implements DAOInterface <Oppo> {

public static DAOOppo getInstance()

{

return new DAOOppo();

}

@Override

public int insert(Oppo t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

Scanner input = new Scanner(System.in);

Oppo op = new Oppo();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

ResultSet rs = stm.executeQuery("SELECT \* FROM OPPO");

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten san pham: ");

String ten = input.nextLine();

System.out.println("Nhap he dieu hanh: ");

String heDieuHanh = input.nextLine();

System.out.println("Nhap mau: ");

String mau = input.nextLine();

System.out.println("Nhap dung luong: ");

String dungLuong = input.nextLine();

Double soLuong;

do{

System.out.println("Nhap so luong: ");

soLuong = input.nextDouble();

}while (soLuong<0);

Double donGia;

do{

System.out.println("Nhap don gia: ");

donGia = input.nextDouble();

} while(donGia<0);

String sqlInsert = "INSERT INTO OPPO (MA, TEN, HEDIEUHANH, MAU, DUNGLUONG, SOLUONG, DONGIA)"

+ "VALUES ('" + ma

+ "' , '" + ten

+ "' , '" + heDieuHanh

+ "' , '" + mau

+ "' , '" + dungLuong

+ "' , '" + soLuong

+ "' , '" + donGia + "')";

rows = stm.executeUpdate(sqlInsert);

System.out.println("----INSERT DIEN THOAI THANH CONG!----");

connection.close();

rs.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public int update(Oppo t) {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public int delete(Oppo t) {

Scanner input = new Scanner (System.in);

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap so lan muon xuat san pham: ");

int n = input.nextInt();

input.nextLine();

for (int i=1; i<=n; i++)

{ System.out.println("-----Lan xoa thu " +i+"-----");

System.out.println("Nhap ma Oppo: ");

String ma = input.nextLine();

String sqlInsert = "DELETE from OPPO "

+ " WHERE MA='" + ma + "'";

rows = stm.executeUpdate(sqlInsert);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.print("Nhap du lieu chua chinh xac");

}

catch(InputMismatchException e)

{

System.out.println("Nhap du lieu chua chinh xac");

}

return rows;

}

@Override

public ArrayList<Oppo> selectAll() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public Oppo selectById(Oppo t) {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public ArrayList<Oppo> selectBySoluongBetween() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO WHERE SOLUONG BETWEEN 1000 AND 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Oppo> selectBySoluong1000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO WHERE SOLUONG < 1000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Oppo> selectBySoluong5000() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO WHERE SOLUONG > 5000";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Oppo> selectByTen(Oppo t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

Scanner input = new Scanner(System.in);

System.out.println("Nhap ten can tim: ");

String ten = input.nextLine();

String selectSql = "SELECT\* FROM OPPO WHERE TEN='"+ten+"'";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Oppo> sapXepTangDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO ORDER BY SOLUONG ASC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

@Override

public ArrayList<Oppo> sapXepGiamDan() {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

ArrayList <Oppo> rows = new ArrayList<Oppo>();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM OPPO ORDER BY SOLUONG DESC";

ResultSet rs = stm.executeQuery(selectSql);

while (rs.next())

{

String ma = rs.getString(1);

String ten = rs.getString(2);

String heDieuHanh = rs.getString(3);

String mau = rs.getString(4);

String dungLuong = rs.getString(5);

double soLuong = rs.getDouble(6);

double donGia = rs.getDouble(7);

Oppo op = new Oppo(ma, ten, heDieuHanh, mau,dungLuong,soLuong,donGia);

rows.add(op);

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

return rows;

}

}

## 2.6. Class DAOPhieu

package DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.time.LocalDate;

import java.util.ArrayList;

import java.sql.Date;

import java.util.Scanner;

import quanlikho.Phieu;

import quanlikho.PhieuNhap;

import java.sql.\*;

/\*\*

\*

\* @author thanh ha

\*/

public class DAOPhieuNhap implements DAOPhieuInterface <PhieuNhap> {

public static DAOPhieuNhap getInstance()

{

return new DAOPhieuNhap();

}

@Override

public void insert(PhieuNhap t) {

String connectionUrl

= "jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows = 0;

Scanner input = new Scanner(System.in);

PhieuNhap pn = new PhieuNhap();

try (Connection connection = DriverManager.getConnection(connectionUrl); Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM PHIEUNHAP";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap ma phieu: ");

String maPhieu = input.nextLine();

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten nha cung cap: ");

String tenNcc = input.nextLine();

int ngay;

do{

System.out.println("Nhap ngay: ");

ngay = input.nextInt();

} while (ngay>31 || ngay<0);

int thang;

do{

System.out.println("Nhap thang: ");

thang = input.nextInt();

}while (thang>12 || thang<0);

int nam;

do{

System.out.println("Nhap nam: ");

nam = input.nextInt();

}while (nam<1000 || nam>2023);

LocalDate ngayNhap = LocalDate.of(nam, thang, ngay);

String sqlInsert = "INSERT INTO PHIEUNHAP (MAPHIEU, MA, TENNCC, NGAYNHAP) VALUES (?,?,?,?)";

// rows = stm.executeUpdate(sqlInsert);

System.out.println("INSERT PHIEU NHAP THANH CONG!");

PreparedStatement psm = connection.prepareStatement(sqlInsert);

psm.setString(1, maPhieu);

psm.setString(2, ma);

psm.setString(3, tenNcc);

psm.setDate(4, Date.valueOf(ngayNhap));

psm.executeUpdate();

connection.close();

stm.close();

return ;

} // Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

}

}

## 2.7. Class ****DAOPhieuXuat****

package DAO;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.Scanner;

import quanlikho.PhieuNhap;

import quanlikho.PhieuXuat;

import java.sql.\*;

import java.sql.Date;

/\*\*

\*

\* @author thanh ha

\*/

public class DAOPhieuXuat implements DAOPhieuInterface <PhieuXuat> {

public static DAOPhieuXuat getInstance()

{

return new DAOPhieuXuat();

}

public void insert(PhieuXuat t) {

String connectionUrl =

"jdbc:sqlserver://localhost:1433;"

+ "database=KHODIENTHOAI;"

+ "user=sa;"

+ "password=123;"

+ "encrypt=true;"

+ "trustServerCertificate=true;"

+ "loginTimeout=30;";

int rows=0;

Scanner input = new Scanner(System.in);

PhieuXuat px = new PhieuXuat();

try (Connection connection = DriverManager.getConnection(connectionUrl);

Statement stm = connection.createStatement();) {

// Code here.

String selectSql = "SELECT\* FROM PHIEUXUAT";

ResultSet rs = stm.executeQuery(selectSql);

System.out.println("Nhap ma phieu: ");

String maPhieu = input.nextLine();

System.out.println("Nhap ma san pham: ");

String ma = input.nextLine();

System.out.println("Nhap ten cong ty: ");

String congTy = input.nextLine();

int ngay;

do{

System.out.println("Nhap ngay: ");

ngay = input.nextInt();

} while (ngay>31 || ngay<0);

int thang;

do{

System.out.println("Nhap thang: ");

thang = input.nextInt();

}while (thang>12 || thang<0);

int nam;

do{

System.out.println("Nhap nam: ");

nam = input.nextInt();

}while (nam<1000 || nam>2023);

LocalDate ngayNhap = LocalDate.of(nam, thang, ngay);

String sqlInsert ="INSERT INTO PHIEUXUAT (MAPHIEU, MA, CONGTY, NGAYXUAT)"

+ "VALUES ('"+maPhieu+

"' , '"+ma+

"' , '"+congTy+

"' , '"+Date.valueOf(ngayNhap)+

"')";

rows = stm.executeUpdate(sqlInsert);

System.out.println("INSERT PHIEU XUAT THANH CONG!");

while (rs.next())

{

System.out.println(rs.getString(1) +

" " +rs.getString(2) +" "

+ rs.getString(3) +" "

+ rs.getString(4) );

}

connection.close();

}

// Handle any errors that may have occurred.

catch (SQLException e) {

System.out.println(" ");

}

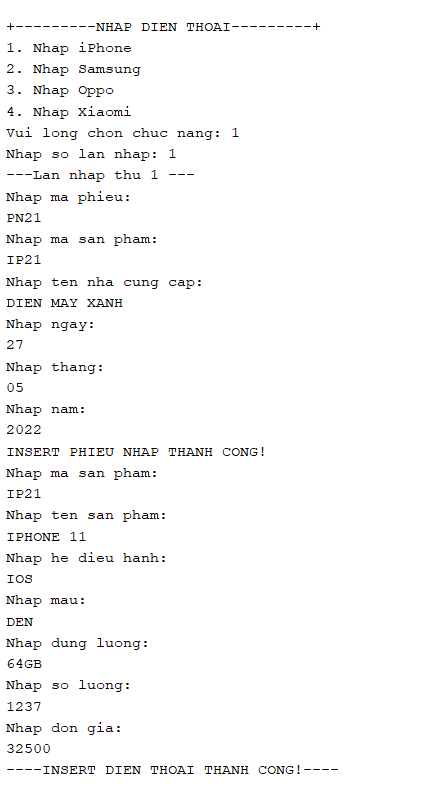
}

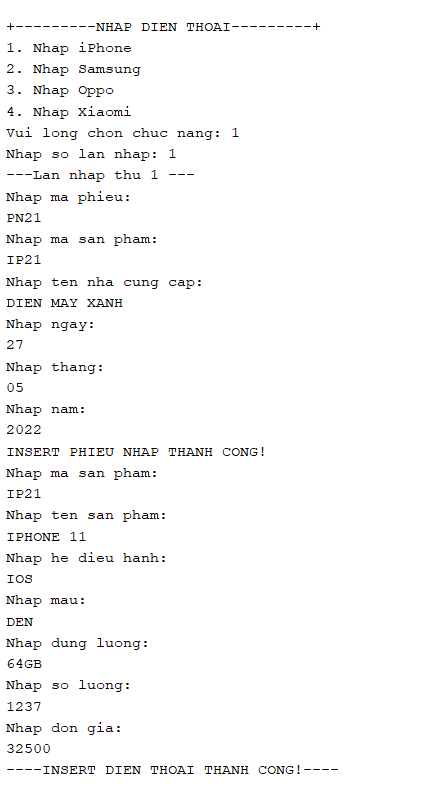
}

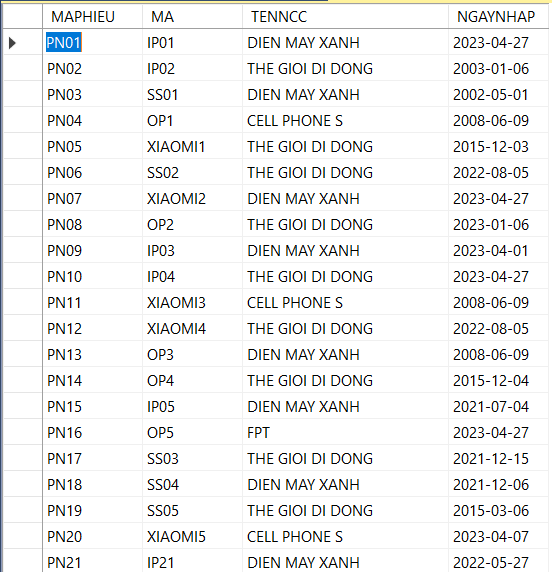
# 

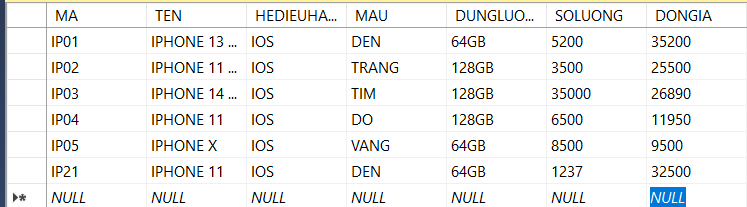
# V. Kiểm thử

# 1. Thêm điện thoại, phiếu nhập:

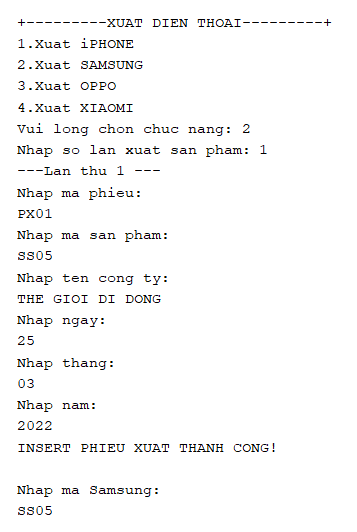






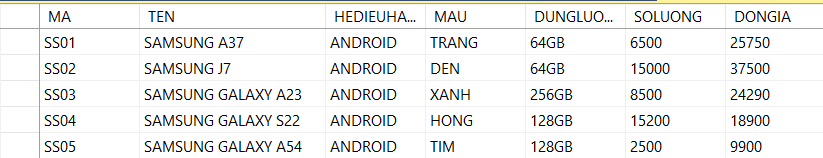


# Thêm phiếu xuất, xóa điện thoại khi xuất hàng

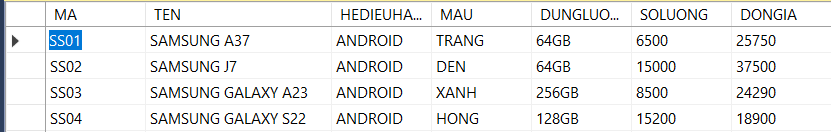




* Trước khi xuất hàng:

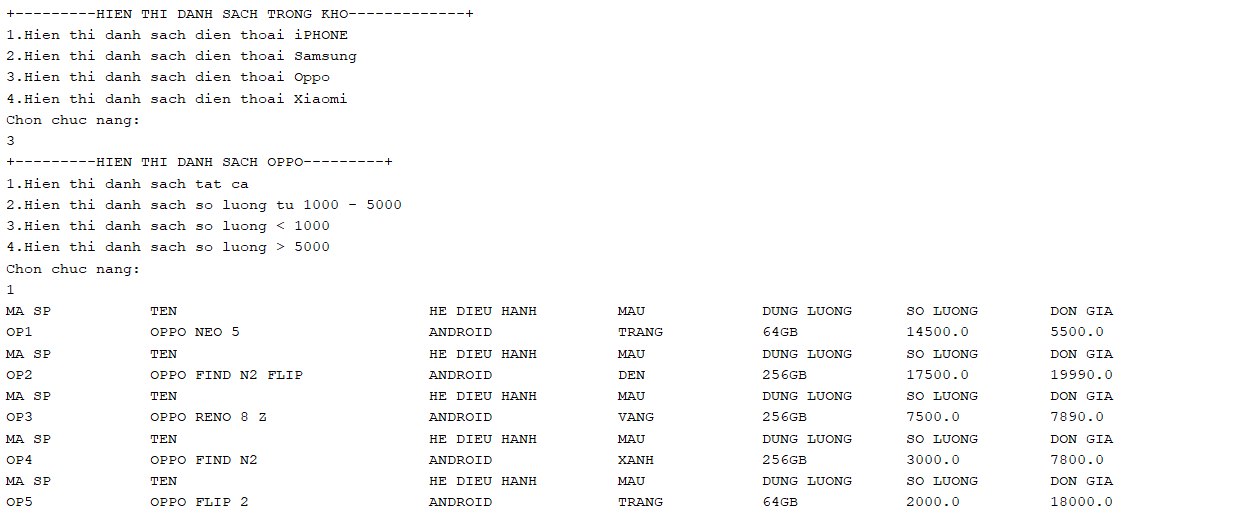


* Sau khi xuất hàng:

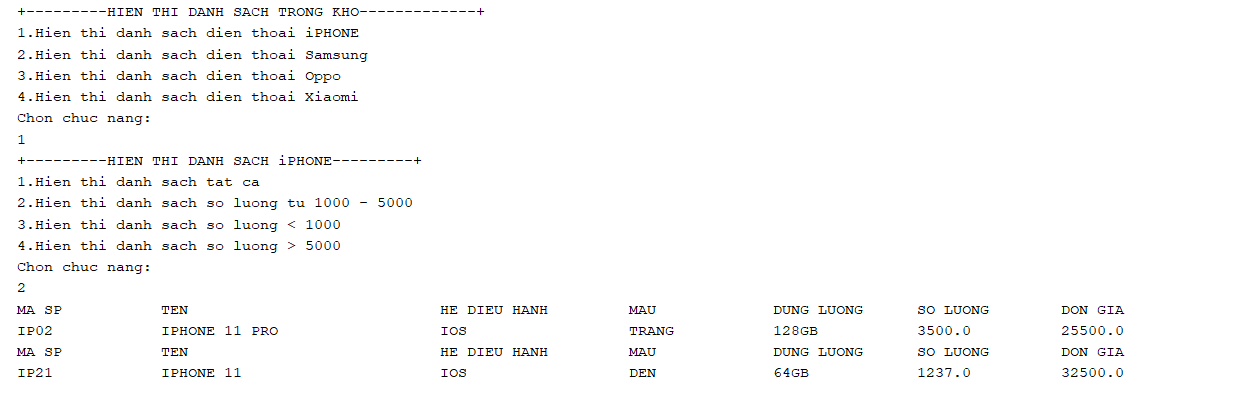


# Hiển thị danh sách điện thoại

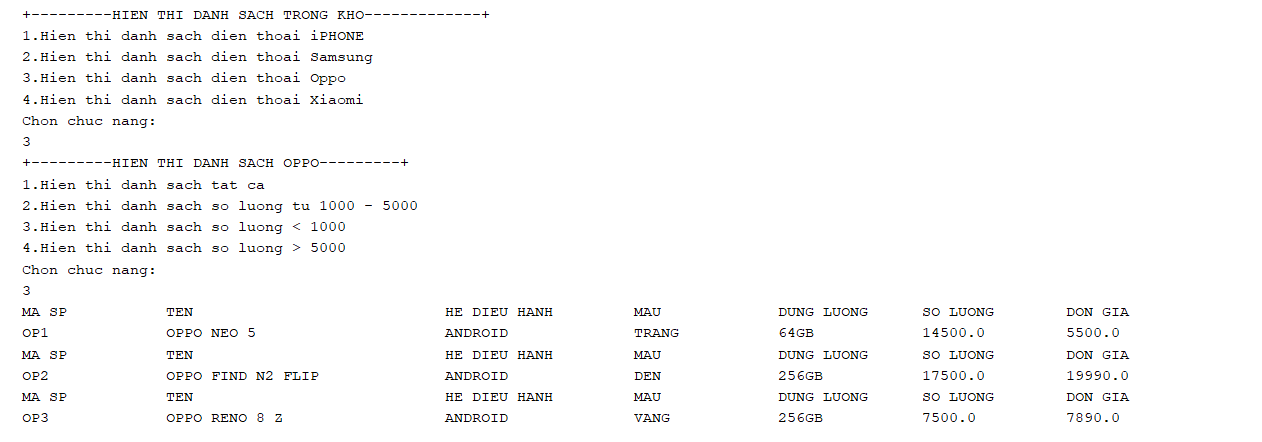
* **Hiển thị danh sách tất cả điện thoại**



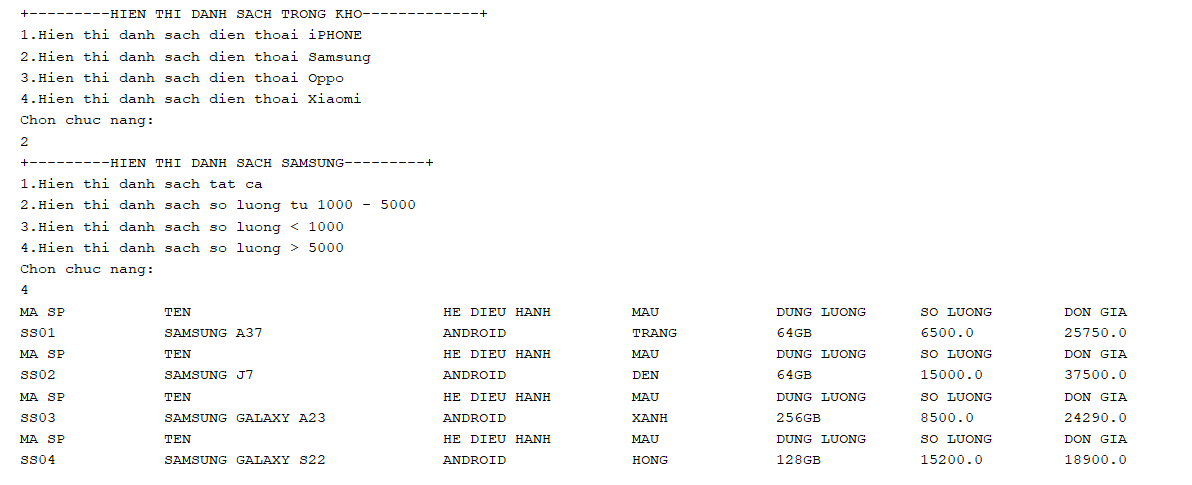
* **Hiển thị danh sách số lượng điện thoại từ 1000 – 5000**



* **Hiển thị danh sách số lượng điện thoại < 1000**

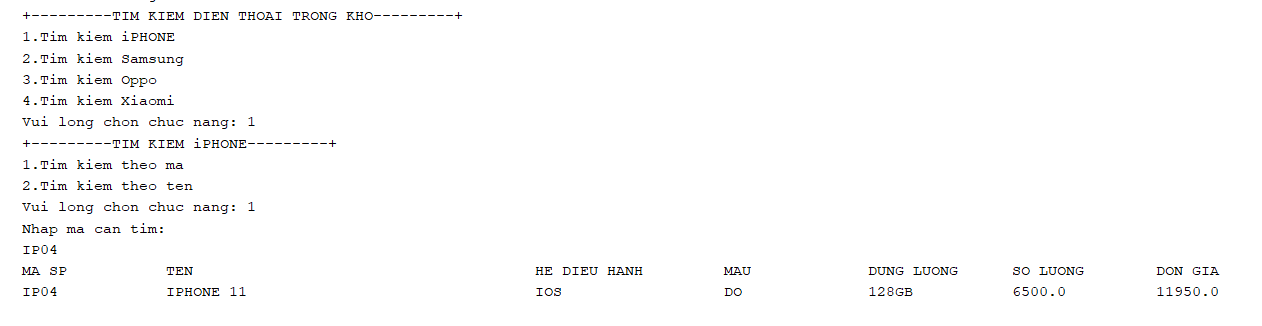


* **Hiển thị danh sách số lượng điện thoại > 5000**

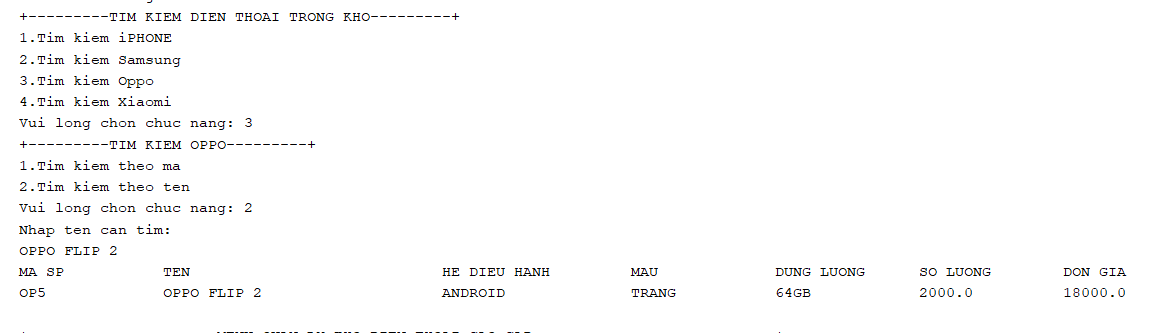


# Tìm kiếm điện thoại:

## Theo mã điện thoại:

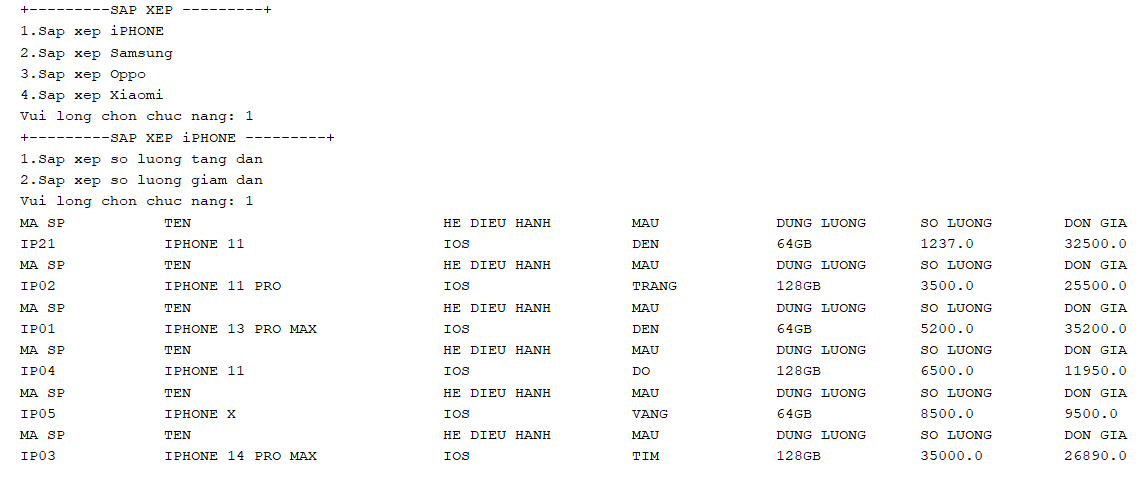


## Theo tên:

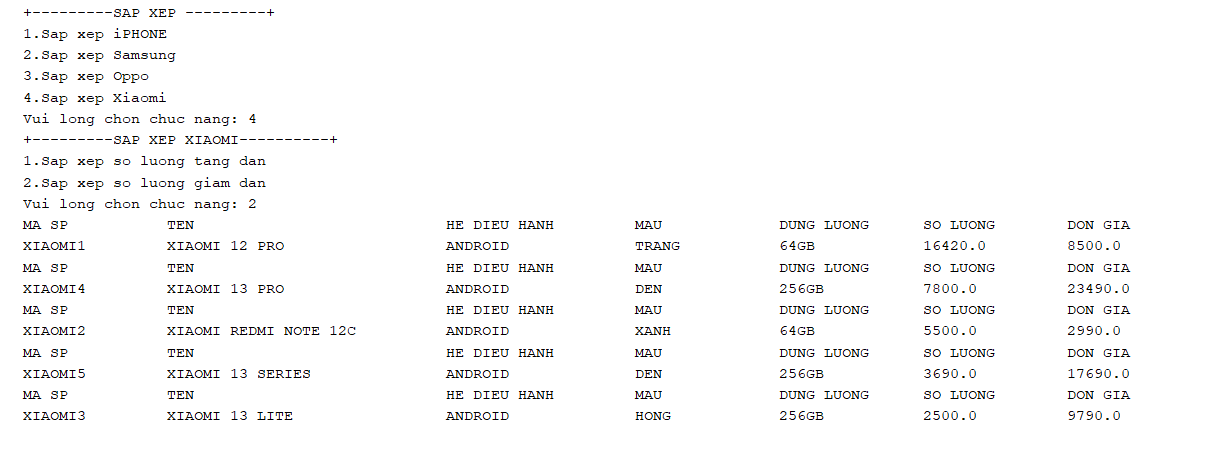


1. **Sắp xếp**

* **Sắp xếp số lượng điện thoại tăng dần**



* **Sắp xếp số lượng điện thoại giảm dần**



# VI. Tài liệu tham khảo

1. <https://learn.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server?view=sql-server-ver16>
2. <https://titv.vn/courses-page/lap-trinh-java-tuong-tac-csdl/>