ĐẠI HỌC MỞ HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO BÀI TẬP LỚN**

|  |
| --- |
| **Môn : Lập trình hướng đối tượng**  **Đề 9: Quản lý kinh doanh cửa hàng thời trang**  Giảng viên hướng dẫn : **Trịnh Thị Xuân**  Sinh viên thực hiện : Phạm Quang Linh  Nguyễn Văn Kiên  Nguyễn Thanh Phong  Nguyễn Chí Tạo 1710A02    Hà Nội – Năm 2019 |
|  |
|  |

1. **Mô tả bài toán**

Trong những năm gần đây, với sự phát triển mạnh mẽ và rộng khắp của ngành công nghệ thông tin (CNTT), kỹ thuật lập trình, máy vi tính đã nhanh chóng thâm nhập và giữ vai trò chủ đạo trong xã hội, kinh tế, quân sự,…Việc ứng dụng tin học các lĩnh vực quản lý hết sức cần thiết và không thể thiếu trong quá trình làm việc để đạt hiệu quả cao nhất. Sự lớn mạnh của máy vi tính, kỹ thuật lập trình đã giúp cho con người rất nhiều trong công việc đặc biệt là trong công tác quản lý, nghiên cứu khoa học,… Thông qua máy vi tính con người có thể xử lý những công việc rất phức tạp mà con người phải rất khó khăn để thực hiện được hoặc không thể thực hiện được. Với sự phát triển không ngừng của CNTT, máy vi tính và kỹ thuật lập trình đã giúp đỡ cho công ty, cửa hàng vừa và nhỏ trong lĩnh vực trao đổi buôn bán hàng hóa đã giúp người quản lý giải quyết công việc một cách nhanh hơn với tốc độ chính xác cao. Chính vì vậy việc tạo ra các phần mềm quản lý hàng hóa ngày càng trở nên thiết thực.

 Ngày nay, khi cuộc sống của mỗi người dân đang từng bước được nâng cao thì nhu cầu về mua sắm, vui chơi giải trí ngày càng được chú trọng. Do đó, các doanh nghiệp cũng quan tâm ưu tiên cho việc phát triển các khu thương mại mua sắm nhằm đáp ứng nhu cầu của người dân. Và là một hình thức khu mua sắm có hiệu quả trong việc đáp ứng các nhu cầu đi lại của mọi người thuộc mọi tầng lớp.Với sự phát triển ngày càng rộng, yêu cầu điện toán hóa các quy trình nghiệp vụ trong cửa hàng thời trang là điều tất yếu, nhằm mục đích làm đơn giản hóa các nghiệp vụ cũng như làm cho khả năng phục vụ khách hàng trở nên nhanh chóng và hiệu quả. Phần mềm “quản lý cửa hàng thời trang” tìm hiểu và xây dựng chương trình sử dụng mà ở đó người dùng có thể dễ dàng quản lý các nghiệp vụ trong một cửa hàng như quản lý hàng hóa (nhập, xuất), quản lý nhân viên, khách hàng…

* Quy trình quản lý

Hàng ngày, các nhân viên bán hàng phải tiếp nhận một số lượng lớn các yêu cầu mua hàng của khách hàng. Hệ thống phải cho phép nhân viên bán lập hoá đơn tính tiền cho khách hàng một cách nhanh chóng . Đối với khách hàng thân thiết thì nhân viên lưu lại thông tin của khách hàng và cấp cho khách hàng thẻ khách hàng thân thiết với số điểm thưởng tương ứng giá trị hoá đơn mua hàng. Trong cửa hàng có rất quầy thu ngân được phụ trách bởi nhân viên bán hàng. Tại mỗi thời điểm chỉ có một nhân viên phụ trách một quầy thu ngân. Hệ thống phải lưu lại phiên làm việc của nhân viên để tiện cho việc quản lý sau này. Vào cuối ngày, bộ phận kế toán sẽ thống kê tình hình bán hàng của cửa hàng, tính toán xem lượng tồn, hàng đã hết để lên kế hoạch nhập hàng mới hay thanh lý hàng tồn. Hệ thống phải cho phép bộ phận kế toán thống kê chi tiết về tình hình của cửa hàng. Vào cuối tháng hoặc khi được ban giám đốc yêu cầu, bộ phận kế toán sẽ lập báo cáo thống kê về tình hình bán của cửa hàng gửi lên ban giám đốc. Hệ thống phải cho phép bộ phận kế toán làm báo cáo thống kê về tình hỉnh bán hàng trong tháng hoặc trong khoảng thời gian cho trước. Khi có hàng hết hoặc cần bổ sung hàng mới, cửa hàng sẽ nhập hàng từ các nhà cung cấp. Thông tin về hàng cần nhập phải được lưu lại để bộ phận kế toán có thể thống kê tình hình mua bán trong cửa hàng. Thông tin này sẽ được trình lên ban giám đốc để duyệt xem có được phép nhập hàng hay không. Khi nhập hàng mới bộ phận kho, tính toán số lượng rồi nhập hàng vào hệ thống. Hệ thống phải cho phép bộ phận kho nhập hàng mới vào hay cập nhật thông tin hàng.

* Trong cơ sở dữ liệu lưu trữ:

+ Thông tin nhân viên gồm: tên nhân viên, giới tính, chức vụ, lương, địa chỉ, điện thoại, ngày sinh

+ Thông tin khách hàng gồm: họ tên, địa chỉ, số điện thoại.

+ Sản phẩm bao gồm: mã sản phẩm, tên sản phẩm,chất liệu vải, Size, số lượng, đơn giá nhập, đơn giá bán, ghi chú.

+ Thông tin về hóa đơn bao gồm: khách hàng mua, ngày mua và các sản phẩm được mua. mỗi hóa đơn có thẻ có nhiều dòng, trong mỗi dòng của hóa đơn có chứa thông tin về tên sản phẩm, đơn vị , số lượng và giá cả.

1. **Các đối tượng được quản lý**
2. **Sơ đồ các lớp**

NhanVien

MaNV : String

ChucVu : String

LuongCoBan : static float

NgaySinh: Date

HeSoLuong :int

NhanVien()

Nhap() :void

Xuat() : void

Get()

Set()

TinhLuong() float

toString()

KhachHang

MaKH :String

KhachHang()

Nhap()

Xuat()

Get()

Set()

toString()

PerSon

HoTen : String

DiaChi : String

DienThoai: long

GioiTinh : String

PerSon ()

Nhap() : void

Xuat() : void

Get()

Set()

toString()

ChiTietHDBan

MaSP : String

SoLuong : int

GiamGia :double

ChiTietHDBan()

Nhap() : void

Xuat() : void

ThanhTien()

toString() :String

get()

set()

DS\_PerSon

DocGhiFile<PerSon> DSperson DocGhiFile<NHANVIEN> DSnhanVien DocGhiFile<KHACHHANG> DSkhachHang

NhapDS()

XuatDSNguoi()

XuatDSNhanVien()

XuatDSKhachHang()

TimNV()

TongLuongNV()

Them1Nguoi()

Xoa1Nguoi()

SapXepNVTheoTen()

SapXepNVTheoLuong()

SapXepNVTheoNgaySinh()

SuaThongTin1NV()

XemThongTinHoaDonCuaKH()

SapXepTheo\_Ten\_NgaySinh\_Luong()

HoaDonBan

MaHDB : String

MaNV : String

MaKH : String

NgayBan : Date

ArrayList<ChiTietHDBan> listCTHD

HoaDonBan ()

Nhap() :void

Xuat() :void

toString() :String

get()

set()

TinhTongTien() :double

SanPham

MaSP : String

TenSP : String

ChatLieuVai :String

Size : int

Soluong : int

DonGiaNhap :double

DonGiaBan : double

SanPham()

Nhap() :void

Xuat() :void

toString() :String

get()

set()

DocGhiFile < T >: generic class

listDT : ArrayList< T >

GhiFile() : void

DocFile() : void

Get()

Set()

DSHoaDon

NhapDSHD()

XuatDSHD()

TongTienTatCaCacHD()

TimHDtheoNgay()

TimHDtheoNV()

Them1HD()

DSSanPham

NhapDSSP()

XuatDSSP()

TimSPBanHet()

SuaInfor()

Them1SP()

1. **Các chức năng xây dựng**

* Nhập danh sách và in lại danh sách các đối tượng Người, Sản Phẩm, Hóa Đơn.
* Xuất danh sách Nhân Viên, Khách Hàng.
* Ghi file và Đọc file cho các đối tượng Người, Sản Phẩm, Hóa Đơn.
* Tính tổng tiền lương của các nhân viên.
* Tính tổng tiền của các hóa đơn.
* Xuất ra những sản phẩm nào đã bán hết.
* Thêm mới một Người, Sản Phẩm, Hóa Đơn.
* Tìm các hóa đơn do nhân viên nào đó lập theo mã nhân viên.
* Tìm nhân viên theo tên.
* Xóa bỏ một Người nào đó theo mã.
* Tìm các hóa đơn được xuất từ ngày này đến ngày khác.
* Sắp xếp danh sách nhân viên theo theo tên,theo ngày sinh, lương tăng dần,.
* Sửa thông tin của một nhân viên.
* Sửa thông tin của một sản phẩm.
* Tìm các hóa đơn đã được mua bởi khách hàng theo mã hoặc tên khách hàng.
* Sắp xếp danh sách nhân viên tăng dần theo tên, nếu tên giống nhau thì sắp xếp theo ngày sinh, nếu ngày sinh trùng nhau sắp xếp theo lương.

1. **Code**

* **Class person**

import java.io.Serializable;

import java.util.Scanner;

public class PerSon implements Serializable{

private static final long serialVersionUID=7890567L;

private String hoTen,diaChi,gioiTinh;

private long dienThoai;

public PerSon() {

}

public PerSon(String hoTen, String diaChi, String gioiTinh, long dienThoai) {

this.hoTen = hoTen;

this.diaChi = diaChi;

this.gioiTinh = gioiTinh;

this.dienThoai = dienThoai;

}

public String getHoTen() {

return hoTen;

}

public void setHoTen(String hoTen) {

this.hoTen = hoTen;

}

public String getDiaChi() {

return diaChi;

}

public void setDiaChi(String diaChi) {

this.diaChi = diaChi;

}

public String getGioiTinh() {

return gioiTinh;

}

public void setGioiTinh(String gioiTinh) {

this.gioiTinh = gioiTinh;

}

public long getDienThoai() {

return dienThoai;

}

public void setDienThoai(long dienThoai) {

this.dienThoai = dienThoai;

}

public String getMa() {

return "";

}

public void setMa(String ma) {

}

public void nhapPerSon(){

Scanner sc=new Scanner(System.in);

do {

System.out.print("\n Nhập tên cho Người này:");

hoTen=sc.nextLine();

if(hoTen.isEmpty())

System.out.print("\n!!! Tên không được để trống.!!!");

} while (hoTen.isEmpty());

System.out.print("\n Nhập địa chỉ cho Người này:");

diaChi=sc.nextLine();

do {

System.out.print("\n Nhập giới tính cho Người này:");

gioiTinh=sc.nextLine();

if(!gioiTinh.toLowerCase().equals("nu") && !gioiTinh.toLowerCase().equals("nam") )

System.out.print("\n!!!Wrong. bạn đã nhập sai giới tính.!!!!");

} while (!gioiTinh.toLowerCase().equals("nu") && !gioiTinh.toLowerCase().equals("nam"));

try {

System.out.print("\n Nhập điện thoại cho Người này:");

dienThoai=sc.nextLong();

} catch (Exception e) {

System.out.print("\n!!!Lỗi :"+e);

}

}

public void xuat(){

}

@Override

public String toString() {

return ", Họ Tên=" + hoTen + ", Địa Chỉ=" + diaChi + ", Giới Tính=" + gioiTinh + ",s Điện Thoại=" + dienThoai ;

}

}

* **Class Nhân Viên**

import java.io.Serializable;

import java.util.Comparator;

import java.util.Date;

import java.util.Scanner;

public class NHANVIEN extends PerSon implements Comparable<NHANVIEN>,Serializable{

private static final long serialVersionUID=123456L;

private String maNV,chucVu;

private static float luongCoBan=1100.0f;

private int heSoLuong;

private Date ngaySinh;

private float luong;

public NHANVIEN() {

super();

}

public NHANVIEN(float lcb){

luongCoBan=lcb;

}

public NHANVIEN(String maNV, String chucVu, int heSoLuong, Date ngaySinh) {

this.maNV = maNV;

this.chucVu = chucVu;

this.heSoLuong = heSoLuong;

this.ngaySinh = ngaySinh;

}

public NHANVIEN(String maNV, String chucVu, int heSoLuong, Date ngaySinh, String hoTen, String diaChi, String gioiTinh, long dienThoai) {

super(hoTen, diaChi, gioiTinh, dienThoai);

this.maNV = maNV;

this.chucVu = chucVu;

this.heSoLuong = heSoLuong;

this.ngaySinh = ngaySinh;

}

@Override

public String getMa() {

return maNV;

}

public float getLuong() {

return luong;

}

public void setLuong(float luong) {

this.luong = luong;

}

@Override

public void setMa(String maNV) {

this.maNV = maNV;

}

public String getChucVu() {

return chucVu;

}

public void setChucVu(String chucVu) {

this.chucVu = chucVu;

}

public static float getLuongCoBan() {

return luongCoBan;

}

public static void setLuongCoBan( float luongCoBan) {

NHANVIEN.luongCoBan = luongCoBan;

}

public int getHeSoLuong() {

return heSoLuong;

}

public void setHeSoLuong(int heSoLuong) {

this.heSoLuong = heSoLuong;

}

public Date getNgaySinh() {

return ngaySinh;

}

public void setNgaySinh(Date ngaySinh) {

this.ngaySinh = ngaySinh;

}

public void nhap(){

Scanner sc=new Scanner(System.in);

do {

System.out.print("\n Nhập mã nhân viên:");

maNV=sc.nextLine();

if(maNV.isEmpty())

System.out.print("\n!!! mã nhân viên không được để trống.!!!");

} while (maNV.isEmpty());

super.nhapPerSon();

System.out.print("\n Nhập chức vụ nhân viên:");

chucVu=sc.nextLine();

do{

System.out.print("\n Nhập hệ số lương nhân viên:");

try {

heSoLuong=sc.nextInt();

sc.nextLine();

if (heSoLuong<=0) {

System.out.print("\n !!!hệ số lương phải lớn hơn 0.!!!");

}

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while(heSoLuong<=0);

luong=luongCoBan\*heSoLuong;

sc.nextLine();

String day;

String[] arrday=new String[3];

//lưu trữ giá trị kiểm tra tính hợp lệ ngày nhập ,=1 thì nhập đúng ngày.

boolean NgayThangNam=false;

do{

do{

System.out.print("\n nhập ngày sinh (dd/mm/yyyy):");

day=sc.nextLine();

if(day.split("/").length!=3)

System.out.print("\n!!!!Bạn đã nhập ngày tháng sai !!!!.");

}while( day.split("/").length!=3);

arrday=day.split("/");

NgayThangNam=NgayHopLe(Integer.valueOf(arrday[0]), Integer.valueOf(arrday[1]), Integer.valueOf(arrday[2]));

if(NgayThangNam==false)

System.out.print("\n !!!!!! ngày tháng năm không chính xác. !!!!!! ");

}while(NgayThangNam==false);

ngaySinh=new Date(Integer.valueOf(arrday[2]),Integer.valueOf(arrday[1]),

Integer.valueOf(arrday[0]));

}

public boolean NamNhuan(int year)

{

return(((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0));

}

public boolean NgayHopLe(int ngay, int thang,int nam)

{

int[] ngayTrongThang = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if(thang <= 0 || thang > 12)

return false;

//Năm nhuận tháng 2 29 ngày:

if(NamNhuan(nam))

ngayTrongThang[1]++;

if(nam<=0)

return false;

return (ngay > 0 && ngay <= ngayTrongThang[thang - 1]);

}/\* \*/

@Override

public void xuat(){

System.out.printf("%-10s%-20s%-25s%-8s%-20s%-15d%-15.3f%-2d/%-2d/%-4d\n",

maNV,this.getHoTen(),this.getDiaChi(),this.getGioiTinh(),chucVu,this.getDienThoai(),luong,ngaySinh.getDate(),

(ngaySinh.getMonth()==0)?12:ngaySinh.getMonth(),

(ngaySinh.getMonth()==0)?ngaySinh.getYear()-1:ngaySinh.getYear());

}

@Override

public String toString() {

return "NHANVIEN{" + "Mã Nhân Viên=" + maNV +super.toString()+ ", Chức vụ=" + chucVu +", Hệ Số Lương=" + heSoLuong

+ ", Ngày Sinh=" + ngaySinh.getDate()+"/"+ ((ngaySinh.getMonth()==0)?12:ngaySinh.getMonth())+"/"+

((ngaySinh.getMonth()==0)?ngaySinh.getYear()-1:ngaySinh.getYear())+", lương="+luong+ '}';

}

public boolean equals(String obj){

String[] temp=(super.getHoTen()).split(" ");

for(int i=0;i<temp.length;i++)

if(obj.equals(temp[i])){

return true;

}

return false;

}

@Override

public int compareTo(NHANVIEN nv) {

if (this.getLuong() == nv.getLuong())

return 0;

else if (this.getLuong() > nv.getLuong())

return 1;

else

return -1;

}

}

class NgaySinhComparator implements Comparator<NHANVIEN> {

@Override

public int compare(NHANVIEN s1, NHANVIEN s2) {

if (s1.getNgaySinh().getYear() == s2.getNgaySinh().getYear())

{

if(s1.getNgaySinh().getMonth()==s2.getNgaySinh().getMonth())

{

if(s1.getNgaySinh().getDate()==s2.getNgaySinh().getDate())

return 0;

else

if(s1.getNgaySinh().getDate()>s2.getNgaySinh().getDate())

return 1;

else

return -1;

}

else

if(s1.getNgaySinh().getMonth()>s2.getNgaySinh().getMonth())

return 1;

else

return -1;

}

else

if (s1.getNgaySinh().getYear() > s2.getNgaySinh().getYear())

return 1;

else

return -1;

}

}

class NameComparator implements Comparator<NHANVIEN> {

@Override

public int compare(NHANVIEN s1, NHANVIEN s2) {

String[] ten1=s1.getHoTen().split(" ");

String[] ten2=s2.getHoTen().split(" ");

return ten1[ten1.length-1].compareTo(ten2[ten2.length-1]);

}

}

class NameNgaysinhLuong implements Comparator<NHANVIEN>{

@Override

public int compare(NHANVIEN nv1,NHANVIEN nv2){

String[] ten1=nv1.getHoTen().split(" ");

String[] ten2=nv2.getHoTen().split(" ");

int value1= ten1[ten1.length-1].compareTo(ten2[ten2.length-1]);

if(value1==0)

{

int value2=nv1.getNgaySinh().compareTo(nv2.getNgaySinh());

if(value2==0)

{

if (nv1.getLuong() == nv2.getLuong())

return 0;

else if (nv1.getLuong() > nv2.getLuong())

return 1;

else

return -1;

}

else return value2;

}

return value1;

}

}

* **Class Khách Hàng**

import java.io.Serializable;

import java.util.Scanner;

public class KHACHHANG extends PerSon implements Serializable{

private static final long serialVersionUID=3214567L;

private String maKH;

public KHACHHANG() {

}

public KHACHHANG(String maKH, String tenKH, String diaChi, long dt,String gt) {

super(tenKH,diaChi,gt,dt);

this.maKH = maKH;

}

@Override

public String getMa() {

return maKH;

}

@Override

public void setMa(String maKH) {

this.maKH = maKH;

}

@Override

public String toString() {

return "KHACHHANG{" + "Mã Khách Hàng=" + maKH + super.toString()+ '}';

}

public void nhap(){

Scanner sc=new Scanner(System.in);

do {

System.out.print("\n nhập mã khách hàng:");

maKH=sc.nextLine();

if(maKH.isEmpty())

System.out.print("\n!!! mã khách hàng không được để trống.!!!");

} while (maKH.isEmpty());

super.nhapPerSon();

}

@Override

public void xuat(){

System.out.printf("\n%-10s%-20s%-25s%-15d%-7s",maKH,this.getHoTen(),

this.getDiaChi(),this.getDienThoai(),this.getGioiTinh());

}

}

* **Class danh sách Người**

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class DS\_PERSON {

Scanner sc =new Scanner(System.in);

DocGhiFile<PerSon> DSperson=new DocGhiFile<>();

DocGhiFile<NHANVIEN> DSnhanVien=new DocGhiFile<>();

DocGhiFile<KHACHHANG> DSkhachHang=new DocGhiFile<>();

public void nhap(){

int i,soluong=-1;

do{

System.out.print("\nnhập số lượng người:");

try {

soluong=sc.nextInt();

sc.nextLine();

if(soluong <= 0)

System.out.print("\n !!!!!! SỐ LƯỢNG PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (soluong <= 0);

String loaiNguoi;

for(i=0;i<soluong;i++)

{

System.out.print("\n Nhập loại người (\"nhan vien\" HOẶC \"khach hang\"):");

loaiNguoi=sc.nextLine();

if(loaiNguoi.toLowerCase().equals("nhanvien")||loaiNguoi.toLowerCase().equals("nhan vien"))

{

NHANVIEN xNhanvien=new NHANVIEN();

xNhanvien.nhap();

PerSon temp=new PerSon();

for ( int j=0;j<DSperson.listDT.size();j++) {

temp=DSperson.listDT.get(j);

if(xNhanvien.getMa().equals(temp.getMa()))//==temp.maNV

{

System.out.println("\n !!!!!! WRONG!!: bạn đã nhập trùng mã người,mời nhập lại !!!!!! ");

//break;

sc.nextLine();

String mamoi;

System.out.print("\n Nhập mã mới cho người này:");

mamoi=sc.nextLine();

xNhanvien.setMa(mamoi);//=mamoi;

j=-1;

}

}

DSperson.listDT.add(xNhanvien);

DSnhanVien.listDT.add(xNhanvien);

}

else if(loaiNguoi.toLowerCase().equals("khachhang")||loaiNguoi.toLowerCase().equals("khach hang"))

{

KHACHHANG xKhachhang=new KHACHHANG();

xKhachhang.nhap();

PerSon temp=new PerSon();

for ( int j=0;j<DSperson.listDT.size();j++) {

temp=DSperson.listDT.get(j);

if(xKhachhang.getMa().equals(temp.getMa()))//==temp.maNV

{

System.out.println("\n !!!!!! WRONG!!: bạn đã nhập trùng mã người,mời nhập lại !!!!!! ");

//break;

sc.nextLine();

String mamoi;

System.out.print("\n Nhập mã mới cho người này:");

mamoi=sc.nextLine();

xKhachhang.setMa(mamoi);//=mamoi;

j=-1;

}

}

DSperson.listDT.add(xKhachhang);

DSkhachHang.listDT.add(xKhachhang);

}

else i--;

}

}

public void xuat(){

for (PerSon x : DSperson.listDT) {

System.out.print(x.toString()+"\n");

}

}

public void ghiFilePerSon() throws IOException{

DSperson.ghiFile("person.dat");

DSnhanVien.ghiFile("nhanvien.dat");

DSkhachHang.ghiFile("khachhang.dat");

}

public void ghiFileNhanVien() throws IOException{

DSperson.ghiFile("person.dat");

DSnhanVien.ghiFile("nhanvien.dat");

}

public void ghiFileKhachHang() throws IOException{

DSperson.ghiFile("person.dat");

DSkhachHang.ghiFile("khachhang.dat");

}

public void docFilePerSon() throws IOException, FileNotFoundException, ClassNotFoundException{

DSperson.docfile("person.dat");

DSnhanVien.docfile("nhanvien.dat");

DSkhachHang.docfile("khachhang.dat");

}

public void XuatDSKH(){

if(DSkhachHang.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");

return;

}

System.out.print("\ndanh sách khách hàng:");

System.out.printf("\n%-10s%-20s%-25s%-15s%-7s","mã KH","tên KH","địa chỉ","số đt","GT");

for (KHACHHANG x : DSkhachHang.listDT) {

x.xuat();

}

}

public void xuatDSNV(){

if(DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");

return;

}

System.out.println("\ndanh sách nhân viên:");

System.out.printf("%-10s%-20s%-25s%-8s%-20s%-15s%-15s%-8s\n","ma nv","ten nv",

"dia chi","gt","chuc vu","DT","luong","NS");

for (NHANVIEN temp : DSnhanVien.listDT) {

temp.xuat();

}

}

public void XemThongTinHD\_DaMua(DSHD dshd, DSSP dssp){

if(this.DSperson.listDT.isEmpty())

{

System.out.print("\n!!!Chưa đọc file người hoặc file rỗng.!!!");

sc.nextLine();

return;

}

if(dshd.listDT.isEmpty())

{

System.out.print("\n!!!Chưa đọc file hóa đơn hoặc file rỗng.!!!");

sc.nextLine();

return;

}

if(dssp.listDT.isEmpty())

{

System.out.print("\n!!!Chưa đọc file sản phẩm hoặc file rỗng.!!!");

sc.nextLine();

return;

}

String dauVao;//đầu vào có thể là tên hoặc mã khách hàng.

System.out.print("\n Nhập mã hoặc tên khách hàng muốn xem hóa đơn:");

dauVao=sc.nextLine();

for (KHACHHANG kh : this.DSkhachHang.listDT)

{

if(kh.getMa().equals(dauVao) || kh.getHoTen().equals(dauVao))

{

String maKHDuocChon;

maKHDuocChon=kh.getMa();

System.out.printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.printf("\nMã Khách Hàng:%-10s Tên Khách Hàng:%-20s Điện Thoại:%-14s",maKHDuocChon,kh.getHoTen(),kh.getDienThoai());

for (HoaDonBan hdb : dshd.listDT)

{

if(hdb.getMaKH().equals(maKHDuocChon))

{

String maHDDuocChon ;

maHDDuocChon=hdb.getMaHD();

System.out.printf("\nMã Hóa Đơn:%-10s Ngày Bán:%2d/%2d/%4d\n",maHDDuocChon,hdb.getNgayBan().getDate(),

(hdb.getNgayBan().getMonth()==0)?12:hdb.getNgayBan().getMonth(),

(hdb.getNgayBan().getMonth()==0)?hdb.getNgayBan().getYear()-1:hdb.getNgayBan().getYear());

for (int i = 0; i < 87; i++) {

System.out.print("-");

}

System.out.printf("\n%-15s%-15s%-15s%-8s%-12s%-12s%-15s\n",

"Tên Sản Phẩm","Đơn Giá Bán","Chất Liệu","Size","Số Lượng","Giảm Giá(%)","Thành Tiền");

for (int i = 0; i < 87; i++) {

System.out.print("-");

}

for (ChiTietHoaDon cthd : hdb.getListCTHD())

{

String maSPDuocMua = cthd.getMaSP();

for (SanPham sp : dssp.listDT)

{

if(sp.getMaSP().equals(maSPDuocMua))

{

System.out.printf("\n%-15s%-15.2f%-15s%-8d%-12d%-12.2f%-15.2f",

sp.getTenSP(),sp.getDonGiaBan(),sp.getChatLieuVai(),sp.getSize(),cthd.getSoLuong(),cthd.getGiamGia(),cthd.ThanhTien());

}

}

}

System.out.println("");

for (int i = 0; i < 87; i++) {

System.out.print("-");

}

System.out.printf("\n %70sTổng Tiền:%.3f"," ",hdb.TinhTongTien());

}

}

}

}

}

public void timNV()

{

System.out.print("\n nhập tên nhân viên muốn tìm:");

String temp="";

temp=sc.nextLine();

System.out.print("\ndanh sách các nhân viên muốn tìm:\n");

System.out.printf("%-10s%-20s%-25s%-8s%-20s%-15s%-15s%-8s\n","ma nv","ten nv",

"dia chi","gt","chuc vu","DT","luong","NS");

for (NHANVIEN x : DSnhanVien.listDT) {

if(x.equals(temp))

x.xuat();

}

}

public float tongluong(){

float sum=0;

for (NHANVIEN temp : DSnhanVien.listDT) {

sum+=temp.getLuong();

}

return sum;

}

public void add1PerSon() throws IOException{

String loaiNguoi;

int i;

System.out.print("\n Nhập loại người (\"nhan vien\" HOẶC \"khach hang\"):");

loaiNguoi=sc.nextLine();

if(loaiNguoi.toLowerCase().equals("nhanvien")||loaiNguoi.toLowerCase().equals("nhan vien"))

{

NHANVIEN xNhanvien=new NHANVIEN();

xNhanvien.nhap();

PerSon temp=new PerSon();

for ( int j=0;j<DSperson.listDT.size();j++) {

temp=DSperson.listDT.get(j);

if(xNhanvien.getMa().equals(temp.getMa()))

{

System.out.println("\n !!!!!! WRONG!!: bạn đã nhập trùng mã người,mời nhập lại !!!!!! ");

//break;

sc.nextLine();

String mamoi;

System.out.print("\n Nhập mã mới cho người này:");

mamoi=sc.nextLine();

xNhanvien.setMa(mamoi);

j=-1;

}

}

DSperson.listDT.add(xNhanvien);

DSnhanVien.listDT.add(xNhanvien);

ghiFileNhanVien();

}

else if(loaiNguoi.toLowerCase().equals("khachhang")||loaiNguoi.toLowerCase().equals("khach hang"))

{

KHACHHANG xKhachhang=new KHACHHANG();

xKhachhang.nhap();

PerSon temp=new PerSon();

for ( int j=0;j<DSperson.listDT.size();j++) {

temp=DSperson.listDT.get(j);

if(xKhachhang.getMa().equals(temp.getMa()))

{

System.out.println("\n !!!!!! WRONG!!: bạn đã nhập trùng mã người,mời nhập lại !!!!!! ");

sc.nextLine();

String mamoi;

System.out.print("\n Nhập mã mới cho người này:");

mamoi=sc.nextLine();

xKhachhang.setMa(mamoi);

j=-1;

}

}

DSperson.listDT.add(xKhachhang);

DSkhachHang.listDT.add(xKhachhang);

ghiFileKhachHang();

}

else System.out.print("\n!!!! Bạn đã không chọn đúng loại người muốn nhập.");

}

public int TimViTriTheoMa(){

String maTim="";

int indexs = -1;

System.out.print("\n nhập mã của người:");

maTim=sc.nextLine();

for (PerSon x : DSperson.listDT) {

if(maTim.equals(x.getMa()))

{

indexs=DSperson.listDT.indexOf(x);

return indexs;

}

}

return -1;

}

public void xoa1PerSon() throws IOException{

int x=TimViTriTheoMa();

if(x==-1)

System.out.print("\ntrong danh sách không có người muốn xóa.");

else{

DSperson.listDT.remove(x);

if(DSperson.listDT.get(x) instanceof NHANVIEN)

{

DSnhanVien.listDT.remove(x);

ghiFileNhanVien();

}

else if(DSperson.listDT.get(x) instanceof KHACHHANG)

{

DSkhachHang.listDT.remove(x);

ghiFileKhachHang();

}

}

}

public void suaInfor() throws IOException{

int index=TimViTriTheoMa();//chỉ số trong ds người

int indexNV = -1; //chỉ số trong ds nhân viên.

if(index==-1)

{

System.out.print("\n!!!trong danh sách không có người này.");sc.nextLine();

return;

}

String maNV=DSperson.listDT.get(index).getMa();

for (NHANVIEN x : DSnhanVien.listDT) {

if(maNV.equals(x.getMa()))

{

indexNV=DSnhanVien.listDT.indexOf(x);

break;

}

}

if(indexNV==-1 )

{

System.out.print("\n!!!trong danh sách không có nhân viên này.");

sc.nextLine();

}

else

{

NHANVIEN temp=DSnhanVien.listDT.get(indexNV);

System.out.print("\n bạn muốn sửa thông tin gì, mời lựa chọn:");

System.out.print("\n1.địa chỉ; 2.chức vụ; 3.số điện thoại; 4. hệ số lương");

int luachon=-1;

System.out.print("\nvui lòng nhập lựa chọn:");

try {

luachon=sc.nextInt();

sc.nextLine();

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString());

return;

}

switch(luachon)

{

case 1:{

String dc;

System.out.print("\n nhập địa chỉ mới cho nhân viên \""+temp.getHoTen()+"\":");

dc=sc.nextLine();

temp.setDiaChi(dc);//=dc;

DSperson.listDT.set(index, temp);

DSnhanVien.listDT.set(indexNV, temp);

ghiFileNhanVien();

};break;

case 2:{

String cv;

System.out.print("\n nhập chức vụ mới cho nhân viên \""+temp.getHoTen()+"\":");

cv=sc.nextLine();

temp.setChucVu(cv);//=cv;

DSperson.listDT.set(index, temp);

DSnhanVien.listDT.set(indexNV, temp);

ghiFileNhanVien();

};break;

case 3:{

long dienthoai=0;

System.out.print("\n nhập số điện thoại mới cho nhân viên \""+temp.getHoTen()+"\":");

try {

dienthoai=sc.nextLong();

sc.nextLine();

} catch (Exception e) {

System.out.print("\n!!!Lỗi :"+e);

}

temp.setDienThoai(dienthoai);

DSperson.listDT.set(index, temp);

DSnhanVien.listDT.set(indexNV, temp);

ghiFileNhanVien();

};break;

case 4:{

int hsl=0;

do{

System.out.print("\n nhập hệ sô lương mới cho nhân viên \""+temp.getHoTen()+"\":");

try {

hsl=sc.nextInt();

sc.nextLine();

if (hsl<=0) {

System.out.print("\n !!!hệ số lương phải lớn hơn 0.!!!");

}

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while(hsl<=0);

temp.setHeSoLuong(hsl);

temp.setLuong(NHANVIEN.getLuongCoBan()\*hsl);

DSperson.listDT.set(index, temp);

DSnhanVien.listDT.set(indexNV, temp);

ghiFileNhanVien();

};break;

default:System.out.println("bạn đã không lựa chọn mục để sửa nên thông "

+ "tin của nhân viên:\""+temp.getHoTen()+"\" sẽ giữ nguyên");

}

}

}

public void SortByLuong(){//sắp xếp lương giảm dần

Collections.sort(DSnhanVien.listDT);

}

public void SortByName(){

Collections.sort(DSnhanVien.listDT, new NameComparator());

}

public void SortByYear(){

Collections.sort(DSnhanVien.listDT, new NgaySinhComparator());

}

public void sortByNameNSLuong(){

Collections.sort(DSnhanVien.listDT,new NameNgaysinhLuong());

}

}

* **Class Sản Phẩm**

import java.io.Serializable;

import java.util.Scanner;

public class SanPham implements Serializable{

private static final long serialVersionUID=351289098L;

private String maSP, tenSP,ChatLieuVai;//

private int Size,SoLuong;//

private double DonGiaNhap,DonGiaBan;//

public SanPham() {

}

public SanPham(String maSP, String tenSP, String ChatLieuVai, int Size,

int SoLuong, double DonGiaNhap, double DonGiaBan) {

this.maSP = maSP;

this.tenSP = tenSP;

this.ChatLieuVai = ChatLieuVai;

this.Size = Size;

this.SoLuong = SoLuong;

this.DonGiaNhap = DonGiaNhap;

this.DonGiaBan = DonGiaBan;

}

public String getMaSP() {

return maSP;

}

public void setMaSP(String maSP) {

this.maSP = maSP;

}

public String getTenSP() {

return tenSP;

}

public void setTenSP(String tenSP) {

this.tenSP = tenSP;

}

public String getChatLieuVai() {

return ChatLieuVai;

}

public void setChatLieuVai(String ChatLieuVai) {

this.ChatLieuVai = ChatLieuVai;

}

public int getSize() {

return Size;

}

public void setSize(int Size) {

this.Size = Size;

}

public int getSoLuong() {

return SoLuong;

}

public void setSoLuong(int SoLuong) {

this.SoLuong = SoLuong;

}

public double getDonGiaNhap() {

return DonGiaNhap;

}

public void setDonGiaNhap(double DonGiaNhap) {

this.DonGiaNhap = DonGiaNhap;

}

public double getDonGiaBan() {

return DonGiaBan;

}

public void setDonGiaBan(double DonGiaBan) {

this.DonGiaBan = DonGiaBan;

}

public void nhapSP(){

Scanner sc =new Scanner(System.in);

do {

System.out.print("\n nhập mã sản phẩm:");

maSP =sc.nextLine();

if(maSP.isEmpty())

System.out.print("\n!!!mã sản không được để trống.!!");

} while (maSP.isEmpty());

do {

System.out.print("\n nhập tên sản phẩm:");

tenSP =sc.nextLine();

if(tenSP.isEmpty())

System.out.print("\n!!! Tên sản phẩm không được để trống.!!!");

} while (tenSP.isEmpty());

System.out.print("\n nhập chất liệu vải:");

ChatLieuVai =sc.nextLine();

do{

System.out.print("\n nhập size sản phẩm \""+ this.tenSP+"\":");

try {

Size =sc.nextInt();

sc.nextLine();

if(Size <= 0)

System.out.print("\n !!!!!! KÍCH CỠ PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (Size<= 0);

do{

System.out.print("\n nhập số lượng của sản phẩm \""+ this.tenSP+"\":");

try {

SoLuong =sc.nextInt();

sc.nextLine();

if(SoLuong <= 0)

System.out.print("\n !!!!!! SỐ LƯỢNG PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (SoLuong<= 0);

do{

System.out.print("\n nhập đơn giá nhập của sản phẩm \""+ this.tenSP+"\":");

try {

DonGiaNhap =sc.nextDouble();

sc.nextLine();

if(DonGiaNhap <= 0.0)

System.out.print("\n !!!!!! ĐƠN GIÁ NHẬP PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số thực.");

sc.nextLine();

}

}while (DonGiaNhap <= 0.0);

do{

System.out.print("\n nhập đơn giá bán của sản phẩm \""+ this.tenSP+"\":");

try {

DonGiaBan =sc.nextDouble();

sc.nextLine();

if(DonGiaBan < DonGiaNhap )

System.out.print("\n !!!!!! ĐƠN GIÁ BÁN PHẢI LỚN HƠN ĐƠN GIÁ NHẬP !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số thực.");

sc.nextLine();

}

}while(DonGiaBan < DonGiaNhap);

}

public void xuat(){

System.out.printf("\n%-10s%-15s%-15s%-7d%-7d%-15.2f%-15.2f",maSP,tenSP,

ChatLieuVai,Size,SoLuong,DonGiaNhap,DonGiaBan);

}

@Override

public String toString() {

return "SanPham{" + "maSP=" + maSP + ", tenSP=" + tenSP + ", ChatLieuVai="

+ ChatLieuVai + ", Size=" + Size + ", SoLuong=" + SoLuong +

", DonGiaNhap=" + DonGiaNhap + ", DonGiaBan=" + DonGiaBan + '}';

}

}

* **Class chi tiết hóa đơn**

import com.sun.javafx.font.FontResource;

import java.io.Serializable;

import java.util.Scanner;

public class ChiTietHoaDon implements Serializable

{

private static final long serialVersionUID=90817029173L;

private String maSP;

private int soLuong=0;

private double giamGia=0.0;

SanPham temp=new SanPham();

private int indexSP=-1;

public ChiTietHoaDon() {

}

public ChiTietHoaDon(String maSP, int soLuong, float giamGia) {

this.maSP = maSP;

this.soLuong = soLuong;

this.giamGia = giamGia;

}

public String getMaSP() {

return maSP;

}

public void setMaSP(String maSP) {

this.maSP = maSP;

}

public int getSoLuong() {

return soLuong;

}

public void setSoLuong(int soLuong) {

this.soLuong = soLuong;

}

public double getGiamGia() {

return giamGia;

}

public void setGiamGia(float giamGia) {

this.giamGia = giamGia;

}

@Override

public String toString() {

return "ChiTietHoaDon{" + "maSP=" + maSP + ", soLuong=" + soLuong + ", giamGia=" + giamGia + '}';

}

public void nhapCTHD(DSSP dssp){

Scanner sc =new Scanner(System.in);

int coSP=-1;

do {

System.out.print("\n Nhập mã sản phẩm:");

maSP=sc.nextLine();

for (SanPham x : dssp.listDT) {

if(maSP.equals(x.getMaSP()))

{

coSP=1;

if(x.getSoLuong()<=0)

{

coSP=-2;

break;

}

temp=x;

indexSP=dssp.listDT.indexOf(x);

break;

}

}

if(coSP==-1)

System.out.print("\n!!!Trong danh sách sản phẩm không có mã này.!!!");

if(coSP==-2)

System.out.print("\n!!!Sản phẩm này đã hết hàng.!!!");

} while (coSP!=1);

do {

System.out.print("\n Nhập số lượng mua sản phẩm này:");

try {

soLuong=sc.nextInt();

if(soLuong<=0)

System.out.print("\n!!!Wrong. số lượng phải lớn hơn không.!!!");

if(temp.getSoLuong()<soLuong)

System.out.print("\n!!!Số lượng nhập vào lớn hơn số lượng hiện có của sản phẩm trong danh sách.!!");

sc.nextLine();

} catch (Exception e) {

System.out.print("\n!!Lỗi:"+e+" ; nhập vào số nguyên.");

sc.nextLine();

}

} while (soLuong<=0 || temp.getSoLuong()<soLuong);

//cập nhật lại số lượng cho sản phẩm đang nhập này.

dssp.listDT.get(indexSP).setSoLuong(dssp.listDT.get(indexSP).getSoLuong()-this.soLuong);

do {

System.out.print("\n Nhập giảm giá (0 -> 100)%:");

try {

giamGia=sc.nextFloat();

if(giamGia<0 ||giamGia>100)

System.out.print("\n!!Wrong. nhập mức giảm giá sai.!!!");

sc.nextLine();

} catch (Exception e) {

System.out.print("\n!!Lỗi:"+e+" ; nhập vào số nguyên.");

sc.nextLine();

}

} while (giamGia<0 ||giamGia>100);

}

public void xuat(){

System.out.format("\n%-10s%-10d%-15.2f%-15.3f",maSP,soLuong,giamGia,ThanhTien());

}

public double ThanhTien(){

return soLuong\*temp.getDonGiaBan()-soLuong\*temp.getDonGiaBan()\*giamGia/100;

}

}

* **Class hóa đơn bán**

import java.io.Serializable;

import java.util.ArrayList;

import java.util.Date;

import java.util.Scanner;

public class HoaDonBan implements Serializable{

private static final long serialVersionUID=1098027319L;

private String maHD,maNV,maKH;

private Date ngayBan;

private ArrayList<ChiTietHoaDon> listCTHD=new ArrayList<>();

public HoaDonBan() {

}

public HoaDonBan(String maHD, String maNV, String maKH, Date ngayBan) {

this.maHD = maHD;

this.maNV = maNV;

this.maKH = maKH;

this.ngayBan = ngayBan;

}

public HoaDonBan(String maHD, String maNV, String maKH, Date ngayBan,ArrayList<ChiTietHoaDon> listCTHD) {

this.maHD = maHD;

this.maNV = maNV;

this.maKH = maKH;

this.ngayBan = ngayBan;

this.listCTHD = listCTHD;

}

public ArrayList<ChiTietHoaDon> getListCTHD() {

return listCTHD;

}

public void setListCTHD(ArrayList<ChiTietHoaDon> listCTHD) {

this.listCTHD = listCTHD;

}

public String getMaHD() {

return maHD;

}

public void setMaHD(String maHD) {

this.maHD = maHD;

}

public String getMaNV() {

return maNV;

}

public void setMaNV(String maNV) {

this.maNV = maNV;

}

public String getMaKH() {

return maKH;

}

public void setMaKH(String maKH) {

this.maKH = maKH;

}

public Date getNgayBan() {

return ngayBan;

}

public void setNgayBan(Date ngayBan) {

this.ngayBan = ngayBan;

}

@Override

public String toString() {

return "HoaDonBan{" + "maHD=" + maHD + ", maNV=" + maNV + ", maKH=" + maKH + ", ngayBan=" + ngayBan + '}';

}

public void nhapHoaDon(DS\_PERSON dsps,DSSP dssp){

Scanner sc =new Scanner(System.in);

do{

System.out.print("\n Nhập mã hóa đơn:");

maHD=sc.nextLine();

if(maHD.isEmpty())

System.out.print("\n!!!! Mã hóa đơn không được để trống.!!!");

}while(maHD.isEmpty());

int coNV=-1;//co=1 là mã nhân viên vừa nhập có trong danh sách nhân viên.

do{

System.out.print("\n Nhập mã nhân viên:");

maNV=sc.nextLine();

for (NHANVIEN temp : dsps.DSnhanVien.listDT) {

if(maNV.equals(temp.getMa()))

{

coNV=1;

break;

}

}

if(coNV!=1)

System.out.print("\n !!!!!! WRong.Trong danh sách nhân viên không có mã này. !!!!!! ");

}while(coNV!=1);

int coKh=-1;//co=1 là mã kh vừa nhập có trong danh sách kh.

do{

System.out.print("\n Nhập mã khách hàng:");

maKH=sc.nextLine();

for (KHACHHANG temp : dsps.DSkhachHang.listDT) {

if(maKH.equals(temp.getMa()))

{

coKh=1;

break;

}

}

if(coKh!=1)

System.out.print("\n !!!!!! WRong.Trong danh sách khách hàng không có mã này. !!!!!! ");

}while(coKh!=1);

String day;

String[] arrday=new String[3];

//int //lưu trữ giá trị kiểm tra tính hợp lệ ngày nhập ,=1 thì nhập đúng ngày.

boolean NgayThangNam=false;

do{

do{

System.out.print("\n nhập ngày bán (dd/mm/yyyy):");

day=sc.nextLine();

if(day.split("/").length!=3)

System.out.print("\n!!!!Bạn đã nhập ngày tháng sai !!!!.");

}while( day.split("/").length!=3);

arrday=day.split("/");

NgayThangNam=NgayHopLe(Integer.valueOf(arrday[0]), Integer.valueOf(arrday[1]), Integer.valueOf(arrday[2]));

if(NgayThangNam==false)

System.out.print("\n !!!!!! ngày tháng năm không chính xác. !!!!!! ");

}while(NgayThangNam==false);

ngayBan=new Date(Integer.valueOf(arrday[2]),Integer.valueOf(arrday[1]),

Integer.valueOf(arrday[0]));

int soLuongSanPhamMua=0;

System.out.print("\n Nhập số lượng sản phẩm được mua trong hóa đơn này:");

soLuongSanPhamMua=sc.nextInt();

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

ChiTietHoaDon cthd =new ChiTietHoaDon();

cthd.nhapCTHD(dssp);

listCTHD.add(cthd);

}

}

public boolean NamNhuan(int year)

{

return(((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0));

}

public boolean NgayHopLe(int ngay, int thang,int nam)

{

int[] ngayTrongThang = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if(thang <= 0 || thang > 12)

return false;

if(NamNhuan(nam))

ngayTrongThang[1]++;

if(nam<=0)

return false;

return (ngay > 0 && ngay <= ngayTrongThang[thang - 1]);

}

public void xuatHD(){

System.out.println();

for (int i = 0; i < 55; i++) {

System.out.print("\*");

}

System.out.printf("\nMã hóa đơn:%-10s Mã nhân viên:%-10s Mã khách hàng:%-10sNgày Bán:%2d/%2d/%4d",

maHD,maNV,maKH,ngayBan.getDate(),

(ngayBan.getMonth()==0)?12:ngayBan.getMonth(),

(ngayBan.getMonth()==0)?ngayBan.getYear()-1:ngayBan.getYear());

System.out.println();

for (int i = 0; i < 55; i++) {

System.out.print("-");

}

System.out.format("\n%-10s%-10s%-15s%-15s","Mã SP","số lượng","giảm giá(%)","thành tiền");

for (ChiTietHoaDon x : listCTHD) {

x.xuat();

}

System.out.println();

for (int i = 0; i < 55; i++) {

System.out.print("-");

}

}

public double TinhTongTien(){

double sums=0.0;//tổng tiền của 1 hóa đơn

for (ChiTietHoaDon cthd : this.getListCTHD()) {

sums+=cthd.ThanhTien();

}

return sums;

}

}

* **Class đọc ghi file**

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.ArrayList;

public class DocGhiFile <T>{

ArrayList<T> listDT=new ArrayList<>();

public ArrayList<T> getListDT() {

return listDT;

}

public void setListDT(ArrayList<T> listDT) {

this.listDT = listDT;

}

public void ghiFile(String filename) throws FileNotFoundException, IOException{

FileOutputStream fout = new FileOutputStream(filename);

ObjectOutputStream out =new ObjectOutputStream(fout);

out.writeObject(listDT);

out.close();

fout.close();

}

public void docfile(String filename) throws FileNotFoundException, IOException, ClassNotFoundException{

try{

FileInputStream fin =new FileInputStream(filename);

ObjectInputStream obin =new ObjectInputStream(fin);

listDT=(ArrayList)obin.readObject();

obin.close();//

fin.close();

}catch(Exception e){

System.out.print("\n!!!Lỗi :"+e);

}

}

}

* **Class danh sách sản phẩm**

import java.util.Scanner;

public class DSSP extends DocGhiFile<SanPham>{

public void nhapDSSP(){

int sl=-1;

Scanner sc =new Scanner(System.in);

do{

System.out.println("\n nhập số lượng các sản phẩm:");

try {

sl=sc.nextInt();

sc.nextLine();

if(sl <= 0)

System.out.print("\n !!!!!! SỐ LƯỢNG PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (sl <= 0);

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

SanPham sp =new SanPham();

sp.nhapSP();

SanPham temp=new SanPham();

for ( int j=0;j<listDT.size();j++) {

temp=listDT.get(j);

if(sp.getMaSP().equals(temp.getMaSP()))//==temp.maNV

{

System.out.println("\n !!!!!! WRONG!!: bạn đã nhập trùng mã sản phẩm,mời nhập lại !!!!!! ");

//break;

sc.nextLine();

String mamoi;

System.out.print("\n nhập mã mới cho sản phẩm:");

mamoi=sc.nextLine();

sp.setMaSP(mamoi);

j=-1;

}

}

listDT.add(sp);

}

}

public void xuatSP(){

System.out.print("\ndanh sách các sản phẩm:");

System.out.printf("\n%-10s%-15s%-15s%-7s%-7s%-15s%-15s","Mã SP","Tên SP",

"Chất Liệu","Size","SL","Giá Nhập","Giá Bán");

for (SanPham a : listDT) {

a.xuat();

}

}

public void TimSPHet(){

System.out.print("\n các sản phẩm đã hết hàng:");

System.out.printf("\n%-10s%-15s%-15s%-7s%-7s%-15s%-15s","Mã SP","Tên SP",

"Chất Liệu","Size","SL","Giá Nhập","Giá Bán");

for (SanPham x : listDT) {

if(x.getSoLuong()<=0)

x.xuat();

}

}

public void add1SP(){

Scanner sc =new Scanner(System.in);

SanPham sp =new SanPham();

sp.nhapSP();

SanPham temp=new SanPham();

for ( int j=0;j<listDT.size();j++) {

temp=listDT.get(j);

if(sp.getMaSP().equals(temp.getMaSP()))

{

System.out.println("\n !!!!!! WRONG!!: bạn đã nhập trùng mã sản phẩm,mời nhập lại !!!!!! ");

sc.nextLine();

String mamoi;

System.out.print("\n nhập mã mới cho sản phẩm:");

mamoi=sc.nextLine();

sp.setMaSP(mamoi);

j=-1;

}

}

listDT.add(sp);

}

public void SuaInforSP(){

Scanner sc =new Scanner(System.in);

String maTim="";

int index = -1;

System.out.print("\n Nhập mã của sản phẩm muốn sửa:");

maTim=sc.nextLine();

for (SanPham x : listDT) {

if(maTim.equals(x.getMaSP()))

{

index=listDT.indexOf(x);

break;

}

}

if(index==-1)

System.out.print("\ntrong danh sách không có mã sản phẩm muốn sửa thông tin.");

else

{

SanPham temp=listDT.get(index);

System.out.print("\n bạn muốn sửa thông tin gì, mời lựa chọn:");

System.out.print("\n1.Số lượng; 2.Đơn giá bán; 3.size; 4.Chất liệu vải");

int luachon=-1;

System.out.print("\nvui lòng nhập lựa chọn:");

try {

luachon=sc.nextInt();

sc.nextLine();

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString());

return;

}

switch(luachon)

{

case 1:{

int sl=-1;//số lượng

do{

System.out.print("\n nhập số lượng mới cho sản phẩm \""+temp.getTenSP()+"\":");

try {

sl=sc.nextInt();

sc.nextLine();

if(sl < 0)

System.out.print("\n !!!!!! SỐ LƯỢNG PHẢI LỚN HƠN HOẶC BẰNG 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (sl< 0);

temp.setSoLuong(sl);

listDT.set(index, temp);

};break;

case 2:{

double dgb=-1;//đơn giá bán

do{

System.out.print("\n nhập đơn giá bán mới cho sản phẩm \""+temp.getTenSP()+"\":");

try {

dgb=sc.nextDouble();

sc.nextLine();

if(dgb < temp.getDonGiaNhap() )

System.out.print("\n !!!!!! ĐƠN GIÁ BÁN PHẢI LỚN HƠN ĐƠN GIÁ NHẬP !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số thực.");

sc.nextLine();

}

}while(dgb < temp.getDonGiaNhap() );

temp.setDonGiaBan(dgb);

listDT.set(index, temp);

};break;

case 3:{

int kt=-1;//kích thước

do{

System.out.print("\n nhập size mới cho sản phẩm \""+temp.getTenSP()+"\":");

try {;

kt=sc.nextInt();

sc.nextLine();

if(kt <= 0)

System.out.print("\n !!!!!! KÍCH CỠ PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (kt<= 0);

temp.setSize(kt);//=kt;

listDT.set(index, temp);

};break;

case 4:{

String clv;//chất liệu vải

System.out.print("\n nhập chất liệu vải mới cho sản phẩm \""+temp.getTenSP()+"\":");

clv=sc.nextLine();

//sc.nextLine();

temp.setChatLieuVai(clv);

listDT.set(index, temp);

};break;

default:System.out.println("bạn đã không lựa chọn mục để sửa nên thông "

+ "tin của sản phẩm:\""+temp.getTenSP()+"\" sẽ giữ nguyên");

}

}

}

public int countElement() {

return listDT.size();

}

}

* **Class danh sách hóa đơn**

import java.util.Date;

import java.util.Scanner;

public class DSHD extends DocGhiFile<HoaDonBan>{

public void nhapDSHD(DS\_PERSON dsPerSons,DSSP dssp){

int i,soLuong=-1;

Scanner sc =new Scanner(System.in);

do{

System.out.print("\n nhập số lượng hóa đơn:");

try {

soLuong=sc.nextInt();

sc.nextLine();

if(soLuong <= 0)

System.out.print("\n !!!!!! SỐ LƯỢNG PHẢI LỚN HƠN 0. !!!!!! ");

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

}while (soLuong <= 0);

for ( i = 0; i < soLuong; i++) {

HoaDonBan hdb=new HoaDonBan();

hdb.nhapHoaDon(dsPerSons,dssp);

HoaDonBan temp=new HoaDonBan();

for ( int j=0;j<listDT.size();j++) {

temp=listDT.get(j);

if(hdb.getMaHD().equals(temp.getMaHD()))//==temp.maNV

{

System.out.println("\nWRONG!!: bạn đã nhập trùng mã hóa đơn,mời nhập lại");

sc.nextLine();

String mamoi;

System.out.print("\n nhập mã mới cho hóa đơn:");

mamoi=sc.nextLine();

hdb.setMaHD(mamoi);

j=-1;

}

}

listDT.add(hdb);

}

}

public void xuatDSHD(){

for (HoaDonBan x : listDT) {

x.xuatHD();

System.out.printf("\n%20s Tổng tiền của hóa đơn =%.3f"," ",x.TinhTongTien());

}

}

public void TinhTongTienCuaTatCaHD(){

if(this.listDT.isEmpty())

{

System.out.print("\n!!!Chưa đọc file hóa đơn hoặc đọc nhầm file hoặc file rỗng.!!!");

return;

}

double sums=0.0; //tổng tiền của tất cả các hóa đơn

for (HoaDonBan hdb : this.listDT) {

sums+=hdb.TinhTongTien();

}

System.out.printf("\n Tổng tiền của tất cả các hóa đơn là:%.3f",sums);

}

public void add1HD(DS\_PERSON ds\_person,DSSP dssp){

Scanner sc =new Scanner(System.in);

HoaDonBan hdb=new HoaDonBan();

hdb.nhapHoaDon(ds\_person,dssp);

HoaDonBan temp=new HoaDonBan();

for ( int j=0;j<listDT.size();j++) {

temp=listDT.get(j);

if(hdb.getMaHD().equals(temp.getMaHD()))//==temp.maNV

{

System.out.println("\nWRONG!!: bạn đã nhập trùng mã hóa đơn,mời nhập lại");

sc.nextLine();

String mamoi;

System.out.print("\n nhập mã mới cho hóa đơn:");

mamoi=sc.nextLine();

hdb.setMaHD(mamoi);

j=-1;

}

}

listDT.add(hdb);

}

public int countElement() {

return listDT.size();

}

public void TimHDtheoNgay(){

if(this.listDT.isEmpty())

{

System.out.print("\n!!!Danh sách hóa đơn chưa được đọc.!!!");

return;

}

Date ngayBatDau=new Date(1970, 1, 1),ngayKetThuc=new Date(1970, 2, 1);

int duocNgayBatDau=-1;//=1 khi đã xác nhận được ngày bắt đầu tìm kiếm.

Scanner sc =new Scanner(System.in);

for(int i=0;i<2;i++)

{

String day;

String[] arrday=new String[3];

int NgayThangNam=-1;//lưu trữ giá trị kiểm tra tính hợp lệ ngày nhập ,=1 thì nhập đúng ngày.

do{

if(duocNgayBatDau==-1)

do{

System.out.print("\n nhập ngày bắt đầu tìm kiếm (dd/mm/yyyy):");

day=sc.nextLine();

if(day.split("/").length!=3)

System.out.print("\n!!!!Bạn đã nhập ngày tháng sai !!!!.");

}while( day.split("/").length!=3);

else

do{

System.out.print("\n nhập ngày kết thúc tìm kiếm (dd/mm/yyyy):");

day=sc.nextLine();

if(day.split("/").length!=3)

System.out.print("\n!!!!Bạn đã nhập ngày tháng sai !!!!.");

}while( day.split("/").length!=3 );

arrday=day.split("/");

int[] ngayTrongThang = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if(Integer.valueOf(arrday[1]) <= 0 || Integer.valueOf(arrday[1]) > 12)

NgayThangNam= -1;

else {

if((((Integer.valueOf(arrday[2]) % 4 == 0) && (Integer.valueOf(arrday[2]) % 100 != 0)) || (Integer.valueOf(arrday[2]) % 400 == 0)))

ngayTrongThang[1]++;

if(Integer.valueOf(arrday[2])<=0)

NgayThangNam= -1;

else if((Integer.valueOf(arrday[0]) > 0 && Integer.valueOf(arrday[0]) <= ngayTrongThang[Integer.valueOf(arrday[1]) - 1]))

NgayThangNam=1;

else NgayThangNam=-1;

}

if(NgayThangNam==-1)

System.out.print("\n !!!!!! ngày tháng năm không chính xác. !!!!!! ");

}while(NgayThangNam==-1);

if(duocNgayBatDau==-1)

{

ngayBatDau=new Date(Integer.valueOf(arrday[2]),Integer.valueOf(arrday[1]),

Integer.valueOf(arrday[0]));

duocNgayBatDau=1;

}

else{

ngayKetThuc=new Date(Integer.valueOf(arrday[2]),Integer.valueOf(arrday[1]),

Integer.valueOf(arrday[0]));

if(ngayKetThuc.before(ngayBatDau))

{

System.out.print("!!!Wrong.Ngày kết thúc tìm kiếm phải sau ngày bắt đầu tìm kiếm.!!!");

i=0;

}

}

}

System.out.printf("\nDanh sách các hóa đơn được lập từ ngày %d/%d/%d đến ngày %d/%d/%d là",

ngayBatDau.getDate(),(ngayBatDau.getMonth()==0)?12:ngayBatDau.getMonth(),

(ngayBatDau.getMonth()==0)?ngayBatDau.getYear()-1:ngayBatDau.getYear(),ngayKetThuc.getDate(),

(ngayKetThuc.getMonth()==0)?12:ngayKetThuc.getMonth(),

(ngayKetThuc.getMonth()==0)?ngayKetThuc.getYear()-1:ngayKetThuc.getYear());

System.out.printf("\n%-10s%-10s%-10s%s","mã HD","mã NV","mã KH","ngày bán");

for (HoaDonBan hd : listDT) {

if(hd.getNgayBan().after(ngayBatDau)&&hd.getNgayBan().before(ngayKetThuc))

{ hd.xuatHD();

System.out.printf("\n%20s Tổng tiền của hóa đơn =%.3f"," ",hd.TinhTongTien());

}

}

}

public void TimHDLapByNV(){

if(this.listDT.isEmpty())

{

System.out.print("\n!!!Danh sách hóa đơn chưa được đọc.!!!");

return;

}

Scanner sc =new Scanner(System.in);

String manv;

System.out.print("\n Nhập mã nhân viên để tìm các hóa đơn do người này lập:");

manv=sc.nextLine();

System.out.printf("\nDanh sách các hóa đơn được lập bời nhân viên có mã \""+manv+"\" là:");

System.out.printf("\n%-10s%-10s%-10s%s","mã HD","mã NV","mã KH","ngày bán");

for (HoaDonBan hd : listDT) {

if(hd.getMaNV().equals(manv))

{ hd.xuatHD();

System.out.printf("\n%20s Tổng tiền của hóa đơn =%.3f"," ",hd.TinhTongTien());

}

}

}

}

* **Chương trình chính**

import java.io.FileNotFoundException;

import java.io.IOException;

import java.text.ParseException;

import java.util.Scanner;

public class MAIN {

public static void main(String[] args) throws ParseException, IOException, FileNotFoundException, ClassNotFoundException, InterruptedException{//throws ParseException

Scanner sc =new Scanner(System.in);

DS\_PERSON listPerSonDocFile =new DS\_PERSON();

DSHD listHDdocfile=new DSHD();//chứa danh sách đọc từ file.

DSSP listSPdocfile=new DSSP();

XuLyMeNu(listPerSonDocFile, listHDdocfile, listSPdocfile);

}

static void XuLyMeNu(DS\_PERSON listPerSonDocFile,DSHD listHDdocfile,DSSP listSPdocfile) throws IOException, FileNotFoundException, ClassNotFoundException{

Scanner sc =new Scanner(System.in);

DS\_PERSON ds\_person=new DS\_PERSON();

DSSP dssp =new DSSP();

DSHD dshd =new DSHD();

int LuaChonChinh;

do{

LuaChonChinh=-1;

menu();

System.out.print("\n Mời bạn nhập lựa chọn:");

try {

LuaChonChinh=sc.nextInt();

sc.nextLine();

} catch (Exception e) {

System.out.print("\n!!!Lỗi: "+e.toString()+" ; phải nhập số nguyên.");

sc.nextLine();

}

switch(LuaChonChinh){

case 1:{ ds\_person.nhap();};break;

case 2:{

if(ds\_person.DSperson.listDT.isEmpty()){

System.out.print("\n Danh sách trống.");

sc.nextLine();

break;

}

ds\_person.xuat(); sc.nextLine();

};break;

case 3:{

if(ds\_person.DSperson.listDT.isEmpty()){

System.out.print("\n Danh sách trống.");

sc.nextLine();

break;

}

ds\_person.ghiFilePerSon();

};break;

case 4:{ listPerSonDocFile.docFilePerSon();listPerSonDocFile.xuat();sc.nextLine();};break;

case 5:{listPerSonDocFile.xuatDSNV();sc.nextLine();};break;

case 6:{listPerSonDocFile.XuatDSKH();sc.nextLine();};break;

case 7:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");

break;

}

System.out.print("\n Tổng lương của các nhân viên là:"+listPerSonDocFile.tongluong());

sc.nextLine();

};break;

case 8:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");

break;

}

listPerSonDocFile.suaInfor();

};break;

case 9:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");

break;

}

listPerSonDocFile.add1PerSon();

};break;

case 10:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");

sc.nextLine();

break;

}

listPerSonDocFile.SortByName();

listPerSonDocFile.xuatDSNV();

sc.nextLine();

};break;

case 11:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");sc.nextLine();

break;

}

listPerSonDocFile.SortByLuong();

listPerSonDocFile.xuatDSNV();

sc.nextLine();

};break;

case 12:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");sc.nextLine();

break;

}

listPerSonDocFile.SortByYear();

listPerSonDocFile.xuatDSNV();

sc.nextLine();

};break;

case 13:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");sc.nextLine();

break;

}

listPerSonDocFile.timNV();

sc.nextLine();

};break;

case 14:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");sc.nextLine();

break;

}

listPerSonDocFile.xoa1PerSon();

listPerSonDocFile.xuatDSNV();

sc.nextLine();

};break;

case 15:{

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file hoặc danh sách nhân viên trống.");sc.nextLine();

break;

}

listPerSonDocFile.sortByNameNSLuong();

listPerSonDocFile.xuatDSNV();

sc.nextLine();

};break;

case 16:{dssp.nhapDSSP(); };break;

case 17:{

if(dssp.listDT.isEmpty()){

System.out.print("\n Danh sách trống.");

sc.nextLine();

break;

}

dssp.xuatSP();

sc.nextLine();

};break;

case 18:{

if(dssp.listDT.isEmpty()){

System.out.print("\n Danh sách trống.");

sc.nextLine();

break;

}

dssp.ghiFile("sanpham.dat");

};break;

case 19:{

listSPdocfile.docfile("sanpham.dat");

listSPdocfile.xuatSP();

sc.nextLine();

};break;

case 21:{

if(listSPdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file sản phẩm.");

sc.nextLine();

break;

}

listSPdocfile.TimSPHet();

sc.nextLine();

};break;

case 22:{

if(listSPdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file sản phẩm.");

sc.nextLine();

break;

}

listSPdocfile.add1SP();

listSPdocfile.ghiFile("sanpham.dat");

listSPdocfile.xuatSP();

sc.nextLine();

};break;

case 23:{

if(listSPdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file sản phẩm.");

sc.nextLine();

break;

}

listSPdocfile.SuaInforSP();

listSPdocfile.ghiFile("sanpham.dat");

listSPdocfile.xuatSP();

sc.nextLine();

};break;

case 24:{

if(listSPdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file sản phẩm,phải đọc file sản phẩm để có dữ liệu đối chiếu.");

sc.nextLine();

break;

}

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file người, phải đọc file để có dữ liệu đối chiếu.");sc.nextLine();

break;

}

dshd.nhapDSHD(listPerSonDocFile, listSPdocfile);

listSPdocfile.ghiFile("sanpham.dat");

};break;

case 25:{

if(dshd.listDT.isEmpty()){

System.out.print("\n Danh sách trống.");

sc.nextLine();

break;

}

dshd.xuatDSHD();

sc.nextLine();

};break;

case 26:{

if(dshd.listDT.isEmpty()){

System.out.print("\n Danh sách trống.");

sc.nextLine();

break;

}

dshd.ghiFile("hoadon.dat");

};break;

case 27:{

listHDdocfile.docfile("hoadon.dat");

listHDdocfile.xuatDSHD();

sc.nextLine();

};break;

case 28:{

if(listHDdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file hóa đơn.");

sc.nextLine();

break;

}

if(listSPdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file sản phẩm,phải đọc file sản phẩm để có dữ liệu đối chiếu.");

sc.nextLine();

break;

}

if(listPerSonDocFile.DSnhanVien.listDT.isEmpty()){

System.out.print("\n !!!Bạn chưa đọc file người, phải đọc file để có dữ liệu đối chiếu.");sc.nextLine();

break;

}

listHDdocfile.add1HD(listPerSonDocFile, listSPdocfile);

listHDdocfile.ghiFile("hoadon.dat");

listSPdocfile.ghiFile("sanpham.dat");

listHDdocfile.xuatDSHD();

sc.nextLine();

};break;

case 29:{

if(listHDdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file hóa đơn.");

sc.nextLine();

break;

}

listHDdocfile.TinhTongTienCuaTatCaHD();

sc.nextLine();

};break;

case 30:{

if(listHDdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file hóa đơn.");

sc.nextLine();

break;

}

listHDdocfile.TimHDtheoNgay();

sc.nextLine();

};break;

case 31:{

if(listHDdocfile.listDT.isEmpty()){

System.out.print("\n Chưa đọc file hóa đơn.");

sc.nextLine();

break;

}

listHDdocfile.TimHDLapByNV();

sc.nextLine();

};break;

case 32:{

listPerSonDocFile.XemThongTinHD\_DaMua(listHDdocfile, listSPdocfile);

sc.nextLine();

};break;

default: break;

}

}while (LuaChonChinh!=0);

}

static void menu(){

System.out.printf("\n%19s"," ");

for (int i = 0; i < 60; i++) {

System.out.print("=");

}

System.out.printf("\n%20s%35s%24s","|","MENU","|");

System.out.printf("\n%19s"," ");

for (int i = 0; i < 60; i++) {

System.out.print("=");

}

System.out.printf("\n%20s\t%-54s%c","|","1.Nhập danh sách Người(Gồm Nhân Viên và Khách hàng).",'|');

System.out.printf("\n%20s\t%-54s%c","|","2.Xuất danh sách Người.",'|');

System.out.printf("\n%20s\t%-54s%c","|","3.Ghi danh sách Người vào file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","4.Đọc danh sách Người từ file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","5.Xuất danh sách Nhân Viên đọc từ file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","6.Xuất danh sách Khách Hàng đọc từ file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","7.Tính tổng lương của các Nhân Viên.",'|');

System.out.printf("\n%20s\t%-54s%c","|","8.Sửa thông tin Nhân Viên theo mã.",'|');

System.out.printf("\n%20s\t%-54s%c","|","9.Thêm 1 Người vào danh sách.",'|');

System.out.printf("\n%20s\t%-54s%c","|","10.Sắp xếp theo tên Nhân Viên.",'|');

System.out.printf("\n%20s\t%-54s%c","|","11.Sắp sếp theo lương Nhân Viên.",'|');

System.out.printf("\n%20s\t%-54s%c","|","12.Sắp xếp theo ngày sinh Nhân Viên.",'|');

System.out.printf("\n%20s\t%-54s%c","|","13.Tìm Nhân Viên theo tên.",'|');

System.out.printf("\n%20s\t%-54s%c","|","14.Xóa 1 Người theo mã.",'|');

System.out.printf("\n%20s\t%-54s%c","|","15.Sắp xếp theo tên, ngày sinh, lương.",'|');

System.out.printf("\n%20s\t%-54s%c","|","------------------------------- ",'|');

System.out.printf("\n%20s\t%-54s%c","|","16.Nhập danh sách Sản Phẩm.",'|');

System.out.printf("\n%20s\t%-54s%c","|","17.Xuất danh sách Sản Phẩm.",'|');

System.out.printf("\n%20s\t%-54s%c","|","18.Ghi danh sách Sản Phẩm vào file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","19.Đọc danh sách Sản Phẩm từ file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","21.Tìm Sản Phẩm đã bán hết.",'|');

System.out.printf("\n%20s\t%-54s%c","|","22.Thêm 1 Sản Phẩm mới.",'|');

System.out.printf("\n%20s\t%-54s%c","|","23.Sửa thông tin Sản Phẩm đã tồn tại.",'|');

System.out.printf("\n%20s\t%-54s%c","|","------------------------------- ",'|');

System.out.printf("\n%20s\t%-54s%c","|","24.Nhập danh sách Hóa Đơn.",'|');

System.out.printf("\n%20s\t%-54s%c","|","25.Xuất danh sách Hóa Đơn.",'|');

System.out.printf("\n%20s\t%-54s%c","|","26.Ghi danh sách Hóa Đơn vào file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","27.Đọc danh sách Hóa Đơn từ file.",'|');

System.out.printf("\n%20s\t%-54s%c","|","28.Nhập thêm 1 bản ghi hóa đơn mới.",'|');

System.out.printf("\n%20s\t%-54s%c","|","29.Xuất tổng tiền của các hóa đơn.",'|');

System.out.printf("\n%20s\t%-54s%c","|","30.Tìm các hóa đơn được lập trong 1 khoảng thời gian.",'|');

System.out.printf("\n%20s\t%-54s%c","|","31.Tìm các hóa đơn được lập bởi nhân viên theo mã NV.",'|');

System.out.printf("\n%20s\t%-54s%c","|","32.Tìm các hóa đơn của một khách hàng nào đó.",'|');

System.out.printf("\n%20s\t%-54s%c","|","0.Thoát.",'|');

System.out.printf("\n%19s"," ");

for (int i = 0; i < 60; i++) {

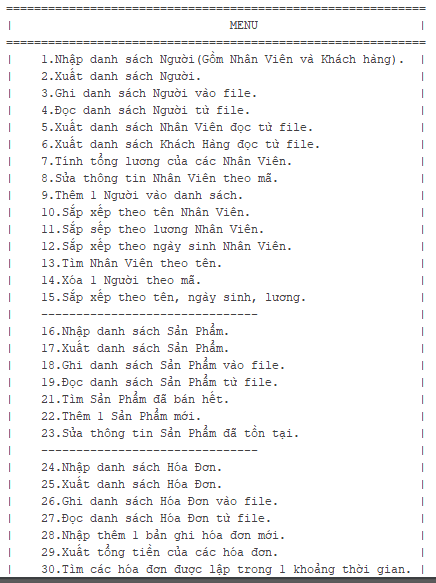
System.out.print("=");

}

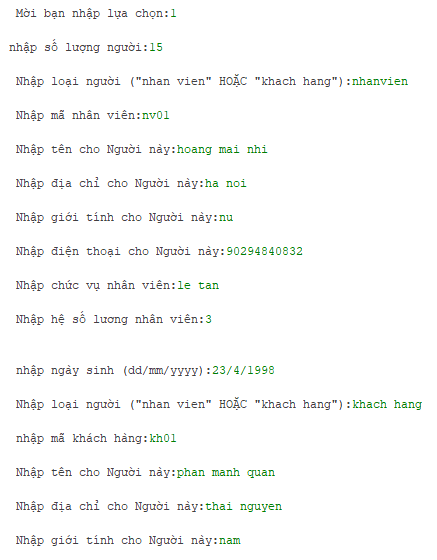
}

}

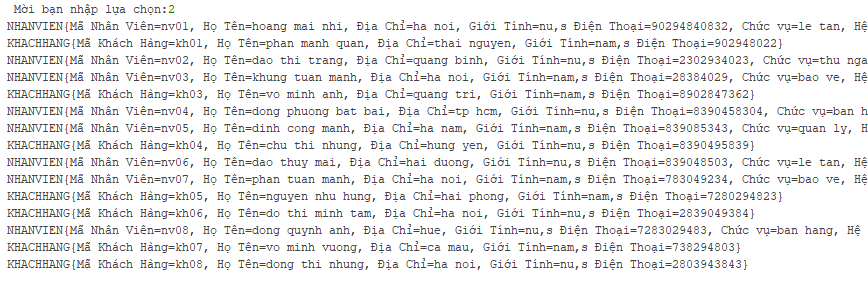
1. **Test**
2. Danh sách các chức năng



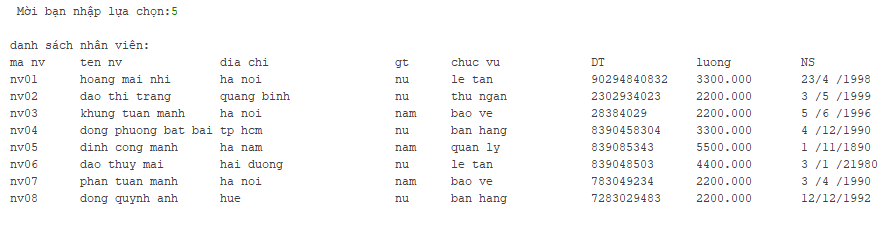
1. Nhập danh sách người



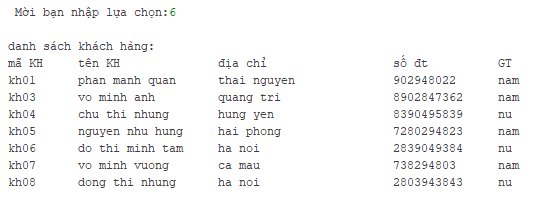
1. Xuất danh sách người



1. Xuất danh sách nhân viên



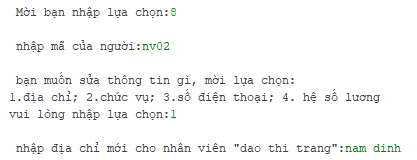
1. Xuất danh sách khách hàng

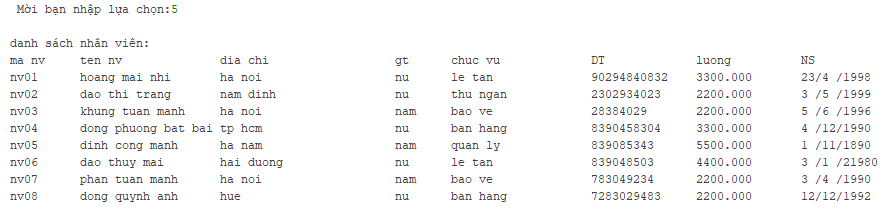


1. Tính tổng lương của các nhân viên

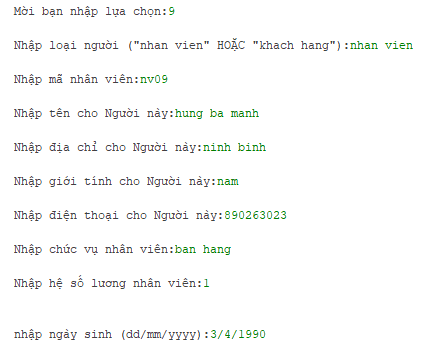


1. Sửa thông tin của nhân viên theo mã

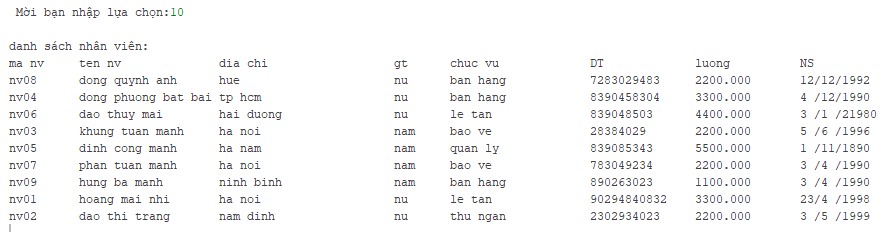




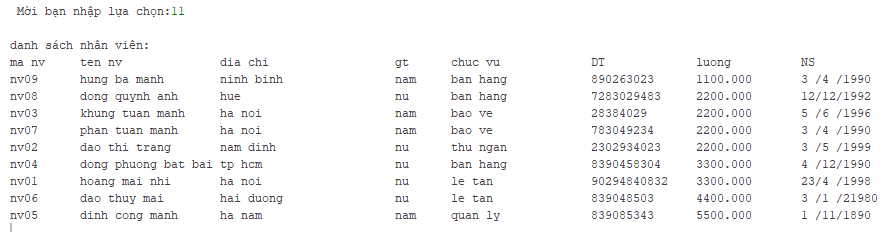
1. Thêm 1 người vào danh sách



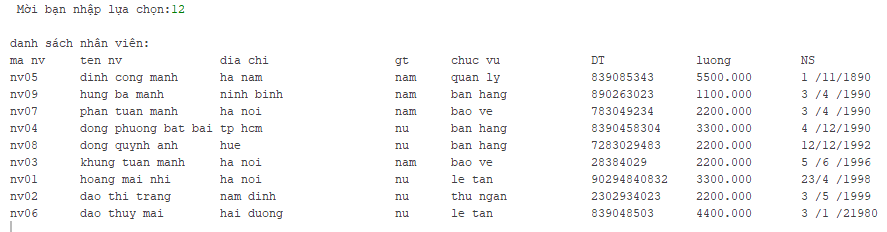
1. Sắp xếp danh sách nhân viên theo tên



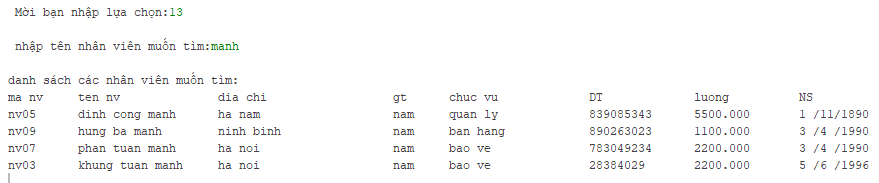
1. Sắp xếp nhân viên theo lương.



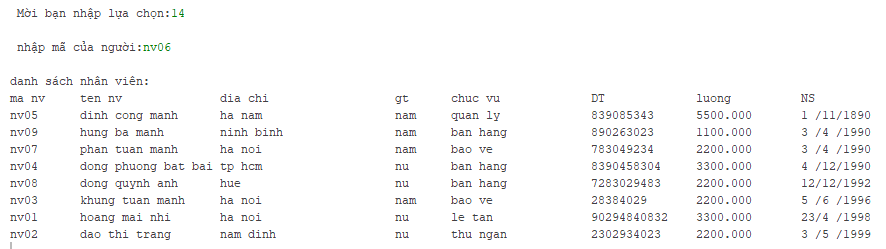
1. Sắp xếp danh sách nhân viên theo ngày sinh



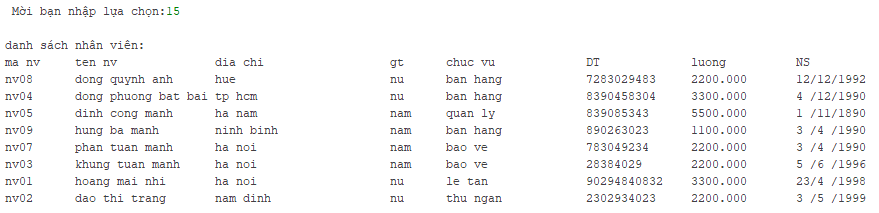
1. Tìm nhân viên theo tên



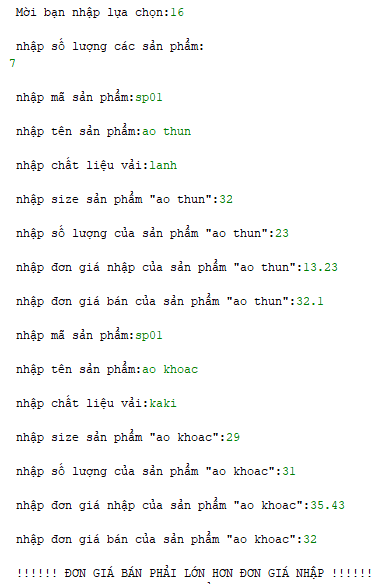
1. Xóa nhân viên theo mã



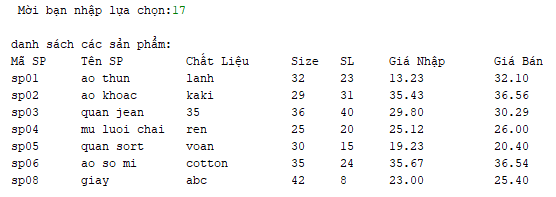
1. Sắp xếp theo tên, ngày sinh, lương.



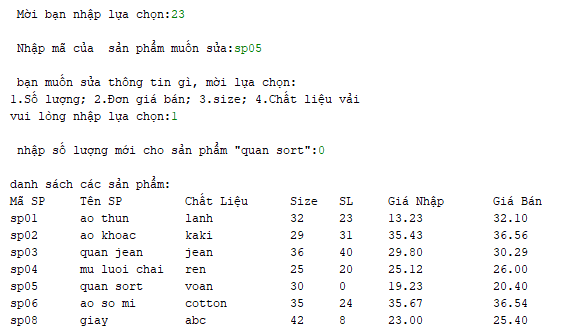
1. Nhập danh sách sản phẩm



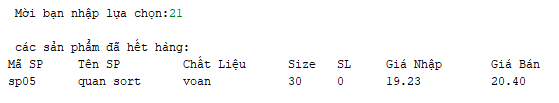
1. Xuất danh sách sản phẩm



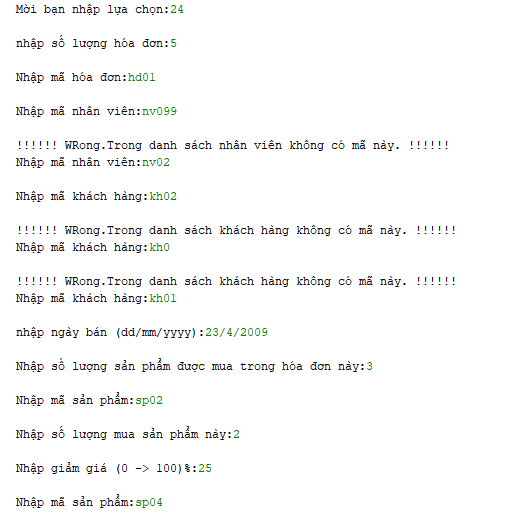
1. Sửa thông tin của sản phẩm



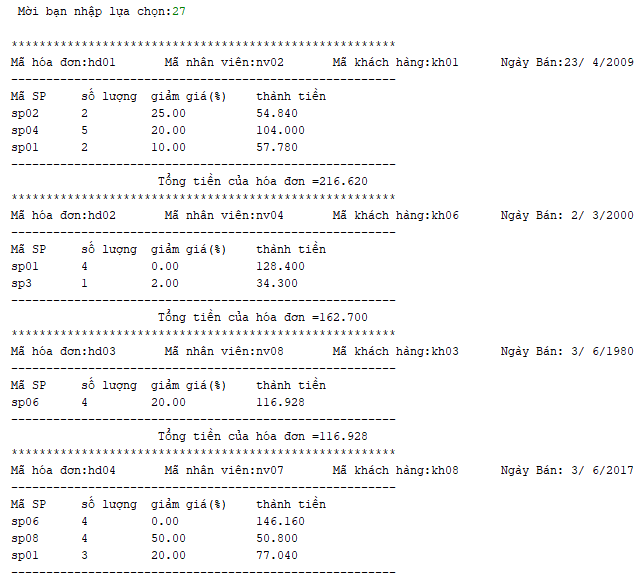
1. Tìm sản phẩm hết hàng

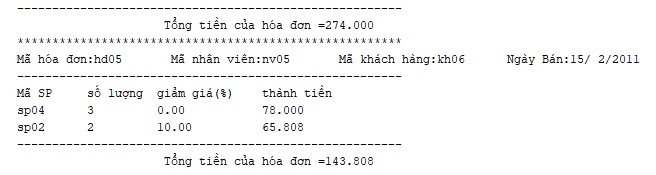


1. Nhập danh sách hóa đơn



1. Xuất danh sách hóa đơn

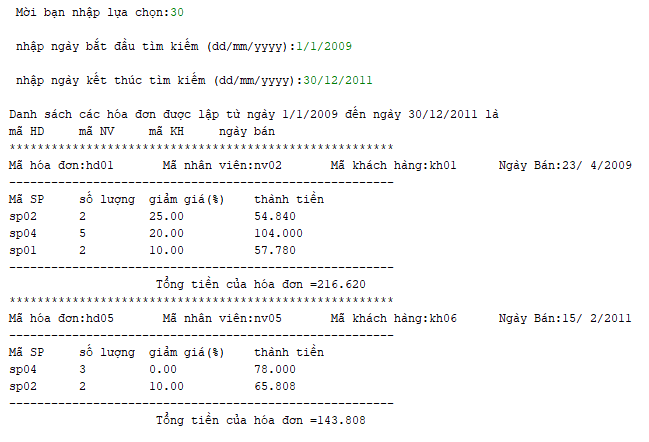




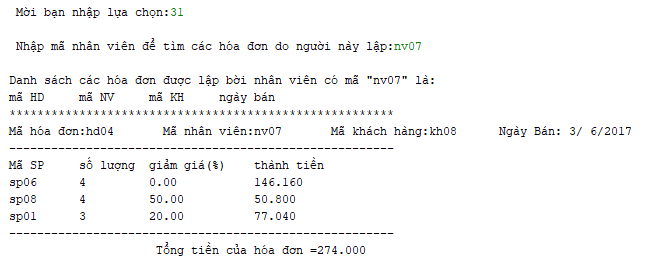
1. Tổng tiền của tất cả hóa đơn



1. Tìm các hóa đơn được lập trong 1 khoảng thời gian



1. Tìm hóa đơn do nhân viên lập theo mã



1. Tìm hóa đơn của khách hàng theo mã hoặc tên của khách hàng

