**TÀI LIỆU THIẾT KẾ HƯỚNG ĐỐI TƯỢNG (MÔ HÌNH LỚP)**

Nhóm 17 - Thành viên nhóm

1. Thân Hoàng Thiên Thiên (Nhóm trưởng)
2. Nguyễn Phan Minh Mẫn
3. Võ Ngọc Thúy Vy
4. Đặng Phúc Nguyên

*Tên ứng dụng:* **CHƯƠNG TRÌNH QUẢN LÝ HIỆU THUỐC TÂY**

Thời gian thực hiện: Từ 19/8/2024 đến 04/11/2024 (11 tuần)

**MỤC LỤC**

[1. Mô hình lớp thực thể (Class Diagram) 1](#_Toc180951402)

[2. Đặc tả sơ đồ lớp: 1](#_Toc180951403)

[2.1 Thực thể TaiKhoan 1](#_Toc180951404)

[2.2 Thực thể SanPham: 3](#_Toc180951405)

[2.3 Thực thể NhanVien: 9](#_Toc180951406)

[2.4 Thực thể KhachHang: 18](#_Toc180951407)

[2.5 Thực thể HoaDon: 23](#_Toc180951408)

[2.6 Thực thể ChiTietHoaDon: 32](#_Toc180951409)

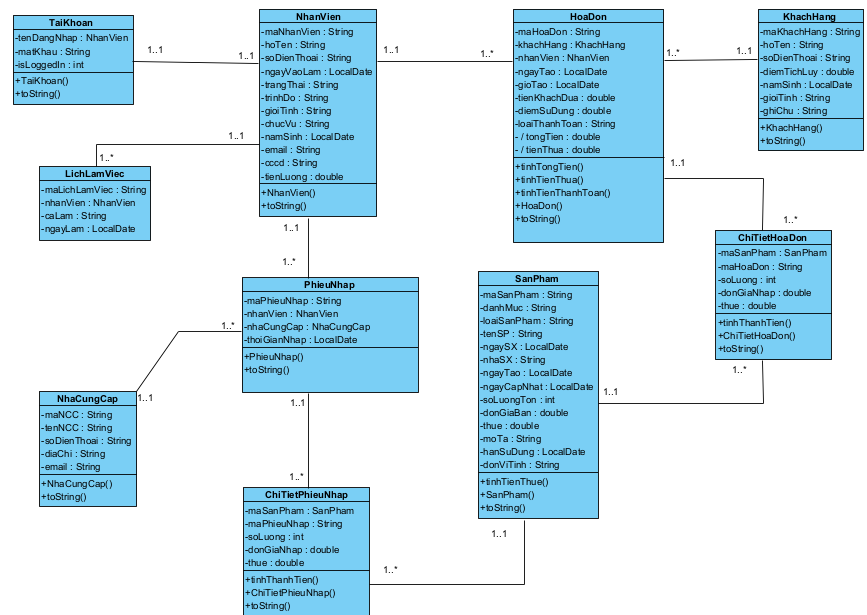
[2.7 Thực thể PhieuNhap: 38](#_Toc180951410)

[2.8 Thực thể NhaCungCap: 42](#_Toc180951411)

[2.9 Thực thể ChiTietPhieuNhap: 46](#_Toc180951412)

[2.10 Thực thể LichLamViec: 50](#_Toc180951413)

# Mô hình lớp thực thể (Class Diagram)



# Đặc tả sơ đồ lớp:

## Thực thể TaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tài khoản** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | tenDangNhap | **NhanVien** | Không được rỗng. | {tenDangNhap} được lấy từ maNhanVien. |
| **1.2** | matKhau | **String** | Mật khẩu mạnh phải dài tối thiểu là 8 ký tự; phải chứa đầy đủ chữ viết thường, chữ viết hoa, ký tự đặc biệt, chữ số.  Không được rỗng. |  |
| **1.3** | isLoggedIn | **Int** | Theo ký tự số:  0: là chưa đăng nhập : là đã đăng nhập |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getTenDangNhap()  setTenDangNhap(NhanVien tenDangNhap) | **NhanVien**  **void** | Không được rỗng. | Throw exception “Tên đăng nhập không được rỗng”. |
| **2.2** | getMatKhau()  setMatKhau(String matKhau) | **String**  **void** | Không được rỗng. | Throw exception “Mật khẩu không được rỗng”. |
| **3** | **Viết các constructor** |  | public TaiKhoan() {} public TaiKhoan(NhanVien tenDangNhap, String matKhau) {  this.tenDangNhap = tenDangNhap;  this.matKhau = matKhau;  }  public TaiKhoan(TaiKhoan original) { this.tenDangNhap = original.tenDangNhap; this.matKhau = original.matKhau; } |  |
| **4** | **Viết phương thức** toString() | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của Tài khoản |

## Thực thể SanPham:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maSP | **String** | Dãy số gồm 5-10 ký tự, duy nhất, Không được rỗng | Mã định danh của sản phẩm |
| **1.2** | tenSP | **String** | Không được rỗng | Tên sản phẩm |
| **1.3** | danhMuc | **String** | Không được rỗng | Danh mục |
|  | loaiSanPham | **String** | Không được rỗng | Loại sản phẩm |
| **1.4** | ngaySX | **LocalDate** | Không được rỗng | Ngày sản xuất |
| **1.5** | nhaSX | **String** | Không được rỗng | Nhà sản xuất |
| **1.6** | ngayTao | **LocalDate** | Tự động phát sinh khi tạo mới | Ngày tạo bản ghi |
| **1.7** | ngayCapNhat | **LocalDate** | Tự động cập nhật khi có thay đổi | Ngày cập nhật cuối |
| **1.8** | soLuongTon | **int** | soLuongTon >= 0  Không được rỗng | Số lượng tồn trong kho |
| **1.9** | donGiaBan | **double** | donGia >= 0  Không được rỗng | Đơn giá của thuốc hoặc thiết bị y tế |
| **1.10** | thue | **float** | 0 <= thue <= 1  Không được rỗng | Thuế |
| **1.11** | hanSuDung | **LocalDate** | Không được rỗng | Hạn sử dụng của thuốc |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | setMaSP(String maSP) | **void** | Phát sinh mã thuốc nếu để trống | Phát sinh tự động mã theo quy luật |
| **2.2** | setDanhMuc(String danhMuc ) | **void** |  |  |
| **2.3** | setTenSP(String tenSP) | **void** | Kiểm tra tham số tenSP không được rỗng | Throw exception "Tên sản phẩm không được rỗng" nếu không hợp lệ |
| **2.4** | setNgaySX(LocalDate ngaySX) | **void** | Kiểm tra tham số ngaySX không được rỗng | Ngày sản xuất không thể trong tương lai |
| **2.5** | setNhaSX(String nhaSX) | **void** | Kiểm tra không rỗng | Throw exception "Nhà sản xuất không được rỗng" nếu không hợp lệ |
| **2.6** | setNgayTao(LocalDate ngayTao) | **void** | Tự động tạo | Thiết lập ngày tạo thuốc |
| **2.7** | setNgayCapNhat(D ngayCapNhat) | **void** | Tự động cập nhật | Thiết lập ngày cập thuốc |
| **2.8** | setSoLuongTon(int soLuongTon) | **void** | soLuongTon >= 0 | Throw exception “Số lượng tồn >= 0” nếu tham số soLuongTon không hợp lệ |
| **2.9** | setDonGiaBan(int donGiaBan) | **void** | donGiaBan >= 0 | Throw exception “Đơn giá bán > 0 ” nếu tham số donGiaBan không hợp lệ |
| **2.10** | setThue(float thue) |  | 0 <= thue <= 1 | Throw exception “0<= Thuế <= 1 ” nếu tham số thue không hợp lệ |
| **2.11** | getMaSP() | **String** |  | Trả về mã sản phẩm |
| **2.12** | getDanhMuc() | **String** |  | Trả về danh mục |
| **2.13** | getTenSP() | **String** |  | Trả về tên sản phẩm |
| **2.14** | getNgaySX() | **LocalDate** |  | Trả về ngày sản xuất |
| **2.15** | getNhaSX() | **String** |  | Trả về nhà sản xuất |
| **2.16** | getNgayTao() | **LocalDate** |  | Trả về ngày tạo |
| **2.17** | getNgayCapNhat() | **LocalDate** |  | Trả về ngày cập nhật |
| **2.18** | getSoLuongTon() | **LocalDate** |  | Trả về số lượng tồn |
| **2.19** | getDonGiaBan() | **double** |  | Trả về đơn giá |
| **2.20** | getThue() | **float** |  | Trả về thuế |
| **2.21** | getHanSuDung() | **LocalDate** |  | Trả về hạn sử dụng |
| **3** | **Viết các constructor** |  | public SanPham() { }  public SanPham(String maSP, String tenSP, LocalDate ngaySX, String nhaSX, LocalDateTime ngayTao, LocalDateTime ngayCapNhat, float soLuongTon, double donGiaBan, float thue, LocalDate hanSuDung) { this.maSP = maSP; this.tenSP = tenSP; this.ngaySX = ngaySX; this.nhaSX = nhaSX; this.ngayTao = ngayTao; this.ngayCapNhat = ngayCapNhat; this.soLuongTon = soLuongTon; this.donGia = donGia; this.thue = thue; this.hanSuDung = hanSuDung; }  public SanPham(SanPham original) { this.maSP = original.maSP; this.tenSP = original.tenSP; this.ngaySX = original.ngaySX; this.nhaSX = original.nhaSX; this.ngayTao = original.ngayTao; this.ngayCapNhat = original.ngayCapNhat; this.soLuongTon = original.soLuongTon; this.donGiaBan = original.donGiaBan; this.thue = original.thue; this.hanSuDung = original.hanSuDung; } |  |
| **4** | **Viết phương thức toString()** | **string** | @Override  public String toString() { return "SanPham{" + "maSP='" + maSP + '\'' + ", tenSP='" + tenSP + '\'' + ", ngaySX=" + ngaySX + ", nhaSX='" + nhaSX + '\'' + ", ngayTao=" + ngayTao + ", ngayCapNhat=" + ngayCapNhat + ", soLuongTon=" + soLuongTon + ", donGiaBan=" + donGiaBan + ", thue=" + thue + ", hanSuDung=" + hanSuDung + '}'; } | Trả về chuỗi chứa giá trị các thuộc tính của sản phẩm |

## Thực thể NhanVien:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maNhanVien | **String** | Không được rỗng.  Mã nhân viên có dạng: MKXXXX với XXXX là 4 chữ số tăng dần. | Tự động phát sinh theo dãy số XXXX tăng dần. |
| **1.2** | hoTen | **String** | Viết hoa chữ cái đầu tiên  Không được rỗng. | Vd: Nguyễn Thị A. |
| **1.3** | email | **String** | Không chứa các ký tự đặc biệt(ngoại trừ ( . ), ( @ ) )  Không được rỗng. | Vd: ngiaoo232@gmail.com. |
| **1.4** | soDienThoai | **String** | Phải đủ 10 ký tự số.  Không được rỗng. | Vd: 0909929929. |
| **1.5** | ngayVaoLam | **LocalDate** | Phải được định dạng: dd/MM/yyyy  Không được rỗng. | Vd: 03/11/2011. |
| **1.6** | trangThai | **String** | Gồm: đang làm, nghỉ việc tạm thời ( như là nghỉ đẻ) , nghỉ việc hẳn, blacklist |  |
| **1.7** | trinhDo | **String** | Chỉ gồm các giá trị: Cao đẳng, Đại học, Cao học. | Trình độ của nhân viên. |
| **1.8** | gioiTinh | **String** | Chỉ gồm các giá trị: Nam, nữ,và khác. | Giới tính của nhân viên. |
| **1.9** | chucVu | **String** | Chỉ gồm: người quản lý, nhân viên |  |
| **1.10** | namSinh | **LocalDate** | Theo định dạng:  dd//MM//yy | Vd: 01/012024 |
| **1.11** | cccd | **String** | Theo định dạng: đủ 12 số | Vd: 012345678912 |
| **1.12** | tienLuong | **Double** | Giá trị tienLuong > 0 |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaNhanVien() | **String** | Kiểm tra không được rỗng. | Throw exception “maNhanVien không hợp lệ” nếu tham số maNhanVien rỗng. |
| 2.2 | setMaNhanVien(string maNhanVien) | **void** | Kiểm tra có đúng định dạng: MKXXXX  XXXX là các chữ số. | Throw exception “maNhanVien không hợp lệ” nếu tham số maNhanVien không hợp lệ. |
| **2.**3 | getHoTen() | **String** | Kiểm tra tham số HoTen không được rỗng. | Throw exception “Họ tên không được rỗng” nếu tham số hoTen rỗng. |
| 2.4 | setHoTen(string hoTen) | **void** | Kiểm tra tham số hoTen có đúng cú pháp không. | Throw exception “Họ tên phải viết hoa chữ cái đầu” nếu tham số hoTen không hợp lệ. |
| **2.**5 | getEmail() | **String** |  | Trả về email của nhân viên. |
| 2.6 | setEmail(string email) | **void** | Không chứa ký tự đặc biệt (ngoại trừ ( . ), ( @ )). Không được rỗng  Theo định dạng: xx@xxxx.com. | Throw exception “Email không hợp lệ ” nếu tham số Email không hợp lệ. |
| **2.7** | getSoDienThoai() | **String** | Kiểm tra phải đủ 10 ký tự số. | Trả về số điện thoại của nhân viên. |
| 2.8 | setSoDienThoai(string soDienThoai) | **void** | Kiểm tra tham số soDienThoai không được rỗng và đủ 10 ký tự số từ 0-9 | Throw exception “ố điện thoại phải đủ 10 ký tự số” nếu tham số SDT rỗng hoặc sai định dạng |
| **2.**9 | getNgayVaoLam() | **LocalDate** | Kiểm tra ngày vào làm đúng theo dạng: dd/MM/yyyy. | Trả về ngày vào làm của nhân viên |
| 2.10 | setNgayVaoLam(string ngayVaoLam) | **void** | Kiểm tra tham số ngayVaoLam đúng theo định dạng dd/MM/yyyy | Throw exception “Ngày vào làm không hợp lệ” nếu tham số ngayVaoLam không đúng định dạng hoặc rỗng |
| **2.**11 | getTrinhDo() | **String** |  | Trả về trình độ của nhân viên |
| 2.12 | setTrinhDo(string trinhDo) | **void** | Kiểm tra tham số trinhDo chỉ mang các giá trị: “Đại học”, “Cao đẳng”, “Cao học” | Throw exception “Trình độ không hợp lệ” nếu tham số trinhDo không đúng định dạng hoặc rỗng |
| **2.13** | getTrangThai() | **String** |  | Trả về trạng thái của nhân viên đang đi làm hay đã nghỉ việc. |
| **2.14** | setTrangThai(String trangThai) | **void** | Gồm: đang làm, nghỉ việc tạm thời ( như là nghỉ đẻ) , nghỉ việc hẳn, blacklist  **True = Đang làm**  **False = Đã nghỉ** | Thiết lập trạng thái đi làm của nhân viên. |
| **2.15** | getGioiTinh() | **String** |  | Trả về giới tính của nhân viên. |
| **2.16** | setGioiTinh(String gioiTinh) | **void** | Kiểm tra tham số GioiTinh chỉ gồm các giá trị: Nam, Nữ, Khác. | Throw exception “Giới tính không hợp lệ” nếu tham số gioiTinh không đúng giá trị hoặc rỗng. |
| **2.17** | getChucVu() | **void** |  | Trả về chức vụ của nhân viên. |
| **2.18** | setChucVu(String chucVu) | **String** | Kiểm tra tham số chức vụ theo định dang: “Người quản lí” hoặc “Nhân viên” | Throw exception “Người quản lý hay Nhân viên” nếu tham số chucVu không đúng giá trị hoặc rỗng. |
| **2.19** | getNamSinh() | **void** |  | Trả về năm sinh của nhân viên. |
| **2.20** | setNamSinh(LocalDate namSinh) | **LocalDate** | Kiểm tra tham số namSinh theo định dang: dd//MM//yyyy | Throw exception “định dạng dd//MM//yyyy” nếu tham số namSinh không đúng giá trị hoặc rỗng. |
| **2.21** | getCccd() | **void** |  | Trả về ký tự số cccd của nhân viên. |
| **2.22** | setCccd(String cccd) | **String** | Kiểm tra định dạng cccd đủ 12 ký tự số | Throw exception “Đủ 12 ký tự số” nếu tham số cccd không đúng giá trị hoặc rỗng. |
| **2.23** | getTienLuong() | **void** |  | Trả về giá trị tienLuong. |
| **2.24** | setTienLuong(double tienLuong) | **Double** | Kiểm tra tienLuong nhập vào có lớn hơn 0 | Throw exception “tienLuong > 0” nếu tham số tienLuong không đúng giá trị hoặc rỗng. |
| **3.** | **Viết các constructor** |  | public NhanVien() {} public NhanVien(String maNhanVien, String hoTen, String email, String soDienThoai, LocalDate ngayVaoLam, String trangThai, String trinhDo, String gioiTinh, LocalDate namSinh) {  this.setMaNhanVien(maNhanVien);  this.setHoTen(hoTen);  this.setEmail(email);  this.setSoDienThoai(soDienThoai);  this.ngayVaoLam = ngayVaoLam;  this.trangThai = trangThai;  this.trinhDo = trinhDo;  this.gioiTinh = gioiTinh;  this.namSinh = namSinh;  }  public NhanVien(NhanVien nv) { this(nv.getMaNhanVien(), nv.getHoTen(), nv.getEmail(), nv.getSoDienThoai(), nv.getNgayVaoLam(), nv.getTrangThai(), nv.getTrinhDo(), nv.getGioiTinh(),nv.getNamSinh())  ; } |  |
| **4.** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của nhân viên |

## Thực thể KhachHang:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Khách hàng** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maKhachHang | **String** | Không được rỗng, là duy nhất | Mã định danh khách hàng |
| **1.2** | hoTen | **String** | Không được rỗng.  Viết hoa chữ cái đầu tiên. | Ví dụ: "Nguyen Van A" |
| **1.3** | soDienThoai | **String** | Phải đủ 10 ký tự số.  Không được rỗng. |  |
| **1.4** | diemTichLuy | **Int** |  | Điểm tích lũy của khách hàng |
| **1.5** | gioiTinh | **String** | Chỉ gồm các giá trị: Nam, nữ,và khác. | Giới tính của khách hàng. |
| **1.6** | namSinh | **LocalDate** | Theo định dạng:  dd//MM//yy | Vd: 01/01/2024 |
| **1.7** | ghiChu | **String** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaKhachHang()  setMaKhachHang(string maKhachHang) | **String**  **void** | Kiểm tra không được rỗng  Kiểm tra có đúng cú pháp | Throw exception “maKhachHang không được rỗng” nếu tham số maKhachHang không hợp lệ.   Throw exception “maKhachHang không hợp lệ” nếu tham số maKhachHang không hợp lệ.  . |
| **2.2** | getHoTen()  setHoTen(string hoTen) | **String**  **void** | Kiểm tra tham số hoTen không được rỗng.  Kiểm tra tham số hoten có viết hoa chữ cái đầu không. | Throw exception “Họ tên không được rỗng” nếu tham số hoTen rỗng  Throw exception “Họ tên phải viết hoa chữ cái đầu” nếu tham số hoTen không hợp lệ. |
| **2.3** | getSoDienThoai()  setSoDienThoai(string soDienThoai) | **String**  **void** | Phải đủ 10 ký tự số.  Không được rỗng. |  |
| **2.4** | getDiemTichLuy()  setDiemTichLuy(int diemTichLuy) | **Int**  **void** |  |  |
| **2.5** | getNamSinh()  setNamSinh(LocalDate namSinh) | **void LocalDate** | Kiểm tra tham số namSinh theo định dang: dd//MM//yyyy | Throw exception “định dạng dd//MM//yyyy” nếu tham số namSinh không đúng giá trị hoặc rỗng. |
| **2.6** | getGhiChu()  setGhiChu(String ghiChu) | **void String** |  |  |
| **3.** | **Viết các constructor** |  | public KhachHang() { }  public KhachHang(String maKhachHang, String hoTen, String soDienThoai, int diemTichLuy) { this.maKhachHang = maKhachHang; this.hoTen = hoTen; this.soDienThoai = soDienThoai; this.diemTichLuy = diemTichLuy; this.namSinh = namSinh;}  public KhachHang(KhachHang original) { this.maKhachHang = original.maKhachHang; this.hoTen = original.hoTen; this.soDienThoai = original.soDienThoai; this.diemTichLuy = original.diemTichLuy; this.namSinh = original.namSinh } |  |
| **4.** | **Viết phương thức toString()** | **String** | @Override public String toString() { return "KhachHang{" + "maKhachHang='" + maKhachHang + '\'' + ", hoTen='" + hoTen + '\'' + ", soDienThoai='" + soDienThoai + '\'' + ", diemTichLuy=" + diemTichLuy + ' + namSinh=" + namSinh + '}'; } | Trả về chuỗi chứa giá trị các thuộc tính của khách hàng. |

## Thực thể HoaDon:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Hoá đơn** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maHoaDon | **String** | Không được rỗng.  Mã hoá đơn có dạng: HDXXXXXXYYYYY bao gồm XXXXXX là thời gian hoá đơn phát sinh (ngày, tháng, 2 số cuối năm), YYYYY là số thứ tự hoá đơn được phát sinh tự động. | Mã hóa đơn |
| **1.2** | khachHang | **KhachHang** | Không được rỗng. | Khách hàng |
| **1.3** | nhanVien | **NhanVien** | Không được rỗng. | Nhân viên |
| **1.4** | Thuoc | **Thuoc** | Không được rỗng | Thuốc |
| **1.5** | ngayTao | **LocalDate** | Không được rỗng  Định dạng dd/mm/yyyy | Ngày tạo |
| **1.6** | tienKhachDua | **double** | Không được là số âm.  Không được rỗng | Tiền khách đưa |
| **1.7** | diemSuDung | **double** | Không được là số âm | Điểm tích lũy sử dụng |
| **1.8** | loaiThanhToan | **String** | Chỉ gồm các giá trị: “Tiền mặt”, “Chuyển khoản” | Phương thức thanh toán |
| **1.9** | gioTao | **LocalDate** | Không được rỗng | Giờ tạo |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaHoaDon() | **String** |  | Lấy ra mã hóa đơn |
| **2.2** | getKhachHang() | **KhachHang** |  | Lấy ra khách hàng |
| **2.3** | getNhanVien() | **NhanVien** |  | Lấy ra nhân viên |
| **2.4** | getThuoc() | **Thuoc** |  | Lấy ra thuốc |
| **2.5** | getNgayTao() | **LocalDate** |  | Lấy ra ngày tạo |
| **2.6** | getGioTao () | **LocalDate** |  | Lấy ra giờ tạo |
| **2.7** | getTienKhachDua() | **double** |  | Lấy ra tiền khách đưa |
| **2.8** | getDiemSuDung() | **double** |  | Lấy ra điểm tích lũy sử |
| **2.9** | getLoaiThanhToan() | **String** |  | Lấy ra loại thanh toán |
| **2.10** | getGhiChu() | **String** |  | Lấy ra ghi chú |
| **2.11** | setMaHoaDon(string maHoaDon) | **void** | Kiểm tra maHoaDon không rỗng và đúng định dạng HDXXXXXXYYYYY bao gồm XXXXXX là thời gian hoá đơn phát sinh (ngày, tháng, 2 số cuối năm), YYYYY là số thứ tự hoá đơn được phát sinh tự động. | Throw exception “Mã hóa đơn không hợp lệ” nếu tham số maHoaDon rỗng và không đúng đinh dạng |
| **2.12** | setKhachHang(KhachHang khachHang) | **void** | Kiểm tra tham số khachHang không rỗng | Throw exception “Khách hàng không hợp lệ” nếu tham số khachHang rỗng |
| **2.13** | setNhanVien(NhanVien nhanVien) | **void** | Kiểm tra tham số nhanVien không rỗng | Throw exception “Nhân viên không hợp lệ” nếu tham số nhanVien rỗng |
| **2.14** | setThuoc(Thuoc thuoc) | **void** | Kiểm tra tham số thuoc không rỗng | Throw exception “Thuốc không hợp lệ” nếu tham số thuoc rỗng |
| **2.15** | setNgayTao (LocalDate ngayTao) | **void** | Kiểm tra tham số ngayTao không rỗng và đúng định dạng dd/mm/yyyy | Throw exception “Ngày tạo không hợp lệ” nếu tham số ngayTao rỗng và không đúng định dạng |
| **2.16** | setGioTao(LocalDate gioTao) | **void** | Kiểm tra tham số gioTao không rỗng và đúng định dạng ss:mm:hh | Throw exception “Giờ tạo không hợp lệ” nếu tham số gioTao rỗng và không đúng định dạng |
| **2.17** | setTienKhachDua(double tienKhachDua) | **void** | Kiểm tra tham số tienKhachDua không rỗng và không được âm | Throw exception “Tiền khách đưa không hợp lệ” nếu tham số tienKhachDua rỗng và là số âm |
| **2.18** | setDiemSuDung(double diemSuDung) | **void** | Kiểm tra tham số diemSuDung không âm | Throw exception “Điểm sử dụng không hợp lệ” nếu tham số diemSuDung là số âm |
| **2.19** | setLoaiThanhToan(String loaiThanhToan) | **void** | Kiểm tra tham số loaiThanhToan không rỗng và chỉ gồm các giá trị: “Tiền mặt”, “Chuyển khoản” | Throw exception “Loại thanh toán không hợp lệ” nếu tham số loaiThanhToan rỗng và không mang giá trị là “Tiền mặt”, “Chuyển khoản” |
| **2.20** | setGhiChu(String ghiChu) | **void** |  |  |
| **3** | **Viết các constructor** |  | public HoaDon() {}  public HoaDon(String maHoaDon, KhachHang khachHang, NhanVien nhanVien, Thuoc thuoc,  LocalDate ngayTao, double tienKhachDua, double diemSuDung, String loaiThanhToan,  List<ChiTietHoaDon>chiTietHoaDon) {  setMaHoaDon(maHoaDon);  setKhachHang(khachHang);  setNhanVien(nhanVien);  setThuoc(thuoc);  setNgayTao(ngayTao);  setTienKhachDua(tienKhachDua);  setDiemSuDung(diemSuDung);  setLoaiThanhToan(loaiThanhToan);  setChiTietHoaDonList(List<ChiTietHoaDonList>)  } |  |
| **4** | **Viết phương thức** toString() | **String** | @Override  public String toString() {  return "HoaDon{" +  "maHoaDon='" + maHoaDon + '\'' +  ", khachHang=" + khachHang +  ", nhanVien=" + nhanVien +  ", thuoc=" + thuoc +  ", ngayTao=" + ngayTao +  ", tienKhachDua=" + tienKhachDua +  ", diemSuDung=" + diemSuDung +  ", loaiThanhToan='" + loaiThanhToan + ", chiTietHoaDonList=" + chiTietHoaDon +  '\'' + '}';  } | Trả về chuỗi chứa giá trị các thuộc tính của Hoá đơn. |

## Thực thể ChiTietHoaDon:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chi tiết hoá đơn** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maSanPham | **SanPham** | Không được rỗng | Mã sản phẩm |
| **1.2** | maHoaDon | **String** | Không được rỗng | Mã hóa đơn |
| **1.3** | soLuong | **int** | Không được là số âm, không được rỗng | Số lượng |
| **1.4** | donGiaNhap | **double** | Phải lớn hơn không, không được rỗng | Đơn giá nhập |
| **1.5** | thue | **double** | Không được là số âm, không được rỗng | Thuế |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaSanPham() | **SanPham** |  | Lấy ra mã sản phẩm |
| **2.2** | setMaSanPham(SanPham sanPham) | **Void** | Kiểm tra tham số maSanPham không rỗng | Throw exception “Mã sản phẩm không được rỗng” nếu tham số maSanPham rỗng và là số âm |
| **2.3** | getMaHoaDon() | **String** |  | Lấy ra mã sản phẩm |
| **2.4** | setMaHoaDon(String maHoaDon) | **Void** | Kiểm tra tham số maHoaDon không rỗng | Throw exception “Mã hóa đơn không được rỗng” nếu tham số maHoaDon rỗng và là số âm |
| **2.5** | getSoLuong() | **int** |  | Lấy ra số lượng |
| **2.6** | setSoLuong(int soLuong) | **void** | Kiểm tra tham số soLuong không được rỗng và không âm | Throw exception “Số lượng không hợp lệ” nếu tham số soLuong rỗng và là số |
| **2.7** | getDonGiaNhap() | **double** |  | Lấy ra đơn giá nhập |
| **2.8** | setMaSanPham(SanPham sanPham) | **void** | Kiểm tra tham số donGiaNhap phải lớn hơn không | Throw exception “Đơn giá nhập phải lớn hơn không” nếu tham số donGiaNhap rỗng và là số âm |
| **2.9** | getThue() | **double** |  | Lấy ra thuế |
| **2.10** | setThue(double thue) | **void** | Kiểm tra tham số thue phải lớn hơn không | Throw exception “Thuế nhập phải lớn hơn không” nếu tham số thue rỗng và là số âm |
| **3** | **Viết các constructor** |  | public ChiTietHoaDon() {}  public ChiTietHoaDon(  SanPham maSanPham,  String maHoaDon,  int soLuong,  double donGiaNhap,  double thue  ) {  setSanPham(sanPham)  setHoaDon(hoaDon); }  public ChiTietHoaDon(ChiTietHoaDon chiTietHoaDon) {  if (chiTietHoaDon != null) {  this.maSanPham = chiTietHoaDon.maSanPham  this.maHoaDon= chiTietHoaDon. maHoaDon  this.donGiaNhap = chiTietHoaDon. donGiaNhap  this.thue = chiTietHoaDon.thue  this.soLuong = chiTietHoaDon.soLuong;}  } |  |
| **4** | **Viết phương thức**  toString() | **String** | @Override  public String toString() {  return "ChiTietHoaDon{" +  "maSanPham=" + maSanPham +  ",maHoaDon=" + maHoaDon+  ", donGiaNhap=" + donGiaNhap +  ", thue=" + thue+  ", soLuong=" + soLuong +  '}';  } | Trả về chuỗi chứa giá trị các thuộc tính của chi tiết hoá đơn. |

## Thực thể PhieuNhap:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maPhieuNhap | **String** | Không được rỗng, định dạng: PNXXXX (XXXX là 4 chữ số) | Mã phiếu nhập |
| **1.2** | nhanVien | **NhanVien** | Không được rỗng | Nhân viên |
| **1.3** | nhaCungCap | **NhaCungCap** | Không được rỗng | Nhà cung cấp |
| **1.4** | thoiGianNhap | **LocalDate** | Không được rỗng | Thời gian |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaPhieuNhap() | **String** |  | Lấy ra mã phiếu nhập |
| **2.2** | getNhanVien() | **NhanVien** |  | Lấy ra nhân viên |
| **2.3** | getNhaCungCap() | **NhaCungCap** |  | Lấy ra nhà cung cấp |
| **2.4** | getThoiGianNhap() | **LocalDate** |  | Lấy ra thời gian nhập |
| **2.5** | setMaPhieuNhap(String maPhieuNhap) | **void** | Kiểm tra tham số maPhieuNhap không rỗng, đúng định dạng: PNXXXX (XXXX là 4 chữ số) | Throw exception “Mã phiếu nhập không hợp lệ” nếu tham số phieuNhap rỗng và không đúng định dạng |
| **2.6** | setNhanVien(NhanVien nhanVien) | **void** | Kiểm tra tham số nhanVien không rỗng | Throw exception “Nhân viên không hợp lệ” nếu tham số nhanVien rỗng |
| **2.7** | setNhaCungCap(NhaCungCap nhaCungCap) | **void** | Kiểm tra tham số nhaCungCap không rỗng | Throw exception “Mã phiếu nhập không hợp lệ” nếu tham số nhaCungCap rỗng |
| **2.8** | setThoiGianNhap(LocalDate thoiGianNhap) | **void** | Kiểm tra tham số thoiGianNhap không rỗng | Throw exception “Thời gian nhập không hợp lệ” nếu tham số thoiGianNhap rỗng |
| **3** | **Viết các constructor** |  | public PhieuNhap() {}  public PhieuNhap(String maPhieuNhap, NhanVien nhanVien, NhaCungCap nhaCungCap, LocalDate thoiGianNhap,  ) { setMaPhieuNhap(maPhieuNhap); setNhanVien(nhanVien); setNhaCungCap(nhaCungCap); setThoiGianNhap(thoiGianNhap);  );}  public PhieuNhap(PhieuNhap phieuNhap) { this(phieuNhap.getMaPhieuNhap(), phieuNhap.getNhanVien(), phieuNhap.getNhaCungCap(), phieuNhap.getThoiGianNhap(),  phieuNhap.getChiTietPhieuNhapList); } |  |
| **4** | **Viết phương thức**  toString() | **String** | @Override  public String toString() {  return "PhieuNhap{" +  "maPhieuNhap='" + maPhieuNhap + '\'' +  ", nhanVien=" + nhanVien +  ", nhaCungCap=" + nhaCungCap +  ", thoiGianNhap=" + thoiGianNhap +  ", chiTietPhieuNhap=" + chiTietPhieuNhapList +  '}';  } | Trả về chuỗi chứa giá trị các thuộc tính của phiếu nhập |

## Thực thể NhaCungCap:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maNCC | **String** | Không được rỗng, định dạng: NCCXXXX (XXXX là 4 chữ số), Là duy nhất | Mã nhà cung cấp |
| **1.2** | tenNCC | **String** | Không được rỗng,viết hoa chữ cái đầu tiên | Tên nhà cung cấp |
| **1.3** | soDienThoai | **String** | Phải đủ 10 ký tự số. | Địa chỉ của nhà cung cấp. |
| **1.4** | diaChi | **String** | Không được rỗng. | Địa chỉ của nhà cung cấp. |
| **1.5** | email | **String** | Không chứa ký tự đặc biệt (ngoại trừ ( . ), ( @ )). Không được rỗng. | Email của nhà cũng cấp |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaNCC() | **String** |  | Lấy mã nhà cung cấp |
| **2.2** | getTenNCC() | **String** |  | Lấy tên nhà cung cấp |
| **2.3** | getSoDienThoai() | **String** |  | Lấy số điện thoại |
| **2.4** | getDiaChi() | **String** |  | Lấy địa chỉ nhà cung cấp |
| **2.5** | getEmail() | **String** |  | Lấy email nhà cung cấp |
| **2.6** | setMaNCC(String maNCC) | **void** | Kiểm tra tham số maNCC không được rỗng và không bị trùng, đúng định dạng NCCXXXX (XXXX là 4 chữ số) | Throw exception “Mã nhà cung cấp không hợp lệ” nếu tham số maNCC rỗng hoặc không đúng định dạng |
| **2.7** | setTenNCC(String tenNCC) | **void** | Kiểm tra tham số tenNCC không rỗng và phải viết hoa chữ cái đầu | Throw exception “Tên nhà cung cấp không hợp lệ” nếu tham số tenNCC rỗng, không đúng định dạng |
| **2.8** | setSoDienThoai(String soDienThoai) | **void** | Kiểm tra tham số soDienThoai không được rỗng và chỉ chứa đủ 10 ký tự số | Throw exception “Số điện thoại nhà cung cấp không hợp lệ” nếu tham số soDienThoai rỗng, không đúng định dạng |
| **2.9** | setEmail(String email) | **void** | Kiểm tra tham số email không được rỗng và Không chứa ký tự đặc biệt (ngoại trừ ( . ), ( @ )) | Throw exception “Email nhà cung cấp không hợp lệ” nếu tham số email rỗng, không đúng định dạng |
| **2.10** | setDiaChi(String diaChi) | **void** | Kiểm tra tham số diaChi không được rỗng | Throw exception “Mã nhà cung cấp không hợp lệ” nếu tham số diaChi rỗng |
| **3** | **Viết các constructor** |  | public NhaCungCap() {}  public NhaCungCap(String maNCC, String tenNCC, String soDienThoai, String diaChi, String email) { setMaNCC(maNCC); setTenNCC(tenNCC); setSoDienThoai(soDienThoai); setDiaChi(diaChi); setEmail(email); }  public NhaCungCap(NhaCungCap nhaCungCap) { this(nhaCungCap.getMaNCC(), nhaCungCap.getTenNCC(), nhaCungCap.getSoDienThoai(), nhaCungCap.getDiaChi(), nhaCungCap.getEmail()); } |  |
| **4** | **Viết phương thức** toString() | **String** | @Override public String toString() { return "NhaCungCap{" + "maNCC='" + maNCC + '\'' + ", tenNCC='" + tenNCC + '\'' + ", soDienThoai='" + soDienThoai + '\'' + ", diaChi='" + diaChi + '\'' + ", email='" + email + '\'' + '}'; } | Trả về chuỗi chứa giá trị các thuộc tính của nhà cung cấp |

## Thực thể ChiTietPhieuNhap:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maSanPham | **SanPham** | Không được rỗng | Sản phẩm |
| **1.2** | maPhieuNhap | **String** | Không được rỗng | Phiếu nhập |
| **1.3** | soLuong | **int** | Không được rỗng | Số lượng |
| **1.4** | donGiaNhap | **double** | Không được rỗng | Đơn giá |
| **1.5** | thue | **double** | Không được rỗng | Thuế |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaSanPham() | **SanPham** |  | Lấy giá trị của mã sản phẩm |
| **2.2** | getMaPhieuNhap | **String** |  | Lấy giá trị của mã phiếu nhập |
| **2.3** | getSoLuong() | **int** |  | Lấy ra số lượng |
| **2.4** | getDonGia() | **double** |  | Lấy ra đơn giá |
| **2.5** | getThue() | **double** |  | Lấy ra thuế |
| **2.6** | setMaSanPham(PhieuNhap maSanPham) | **void** | Kiểm tra tham số maSanPham có rỗng hay không | Throw exception “Mã sản phẩm không hợp lệ” nếu tham số maSanPham rỗng |
| **2.7** | setMaPhieuNhap(String phieuNhap) | **void** | Kiểm tra tham số maPhieuNhap có rỗng hay không | Throw exception “Phiếu nhập không hợp lệ” nếu tham số maPhieuNhap rỗng |
| **2.8** | setSoLuong(int soLuong) | **void** | Kiểm tra tham số soLuong có rỗng hay không | Throw exception “Số lượng không hợp lệ” nếu tham số soLuong rỗng |
| **2.9** | setDonGiaNhap(double donGiaNhap) | **void** | Kiểm tra tham số donGia có rỗng hay không | Throw exception “Đơn giá không hợp lệ” nếu tham số donGia rỗng |
| **2.10** | setThue(double thue) | **void** | Kiểm tra tham số thue có rỗng hay không | Throw exception “Thuế không hợp lệ” nếu tham số thue rỗng |
| **3** | **Viết các constructor** |  | public ChiTietPhieuNhap() {}  public ChiTietPhieuNhap(  maSanPham SanPham  maPhieuNhap String,  int soLuong, double donGiaNhap, double thue) {  this.maSanPham = maSanPham;  this.maPhieuNhap = maPhieuNhap;  this.soLuong = soLuong;  this.donGiaNhap = donGiaNhap;  this.thue = thue;  }  public ChiTietPhieuNhap(ChiTietPhieuNhap chiTietPhieuNhap) {  this(  chiTietPhieuNhap.getMaPhieuNhap(),  chiTietPhieuNhap.getMaHoaDon (),  chiTietPhieuNhap.getSoLuong(),  chiTietPhieuNhap.getDonGiaNhap(),  chiTietPhieuNhap.getThue());  } |  |
| **4** | **Viết phương thức** toString() | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của chi tiết phiếu nhập |

## Thực thể LichLamViec:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| **1.1** | maLichLamViec | **String** | Không được rỗng, là duy nhất | Mã định danh lịch làm việc |
| **1.2** | nhanVien | **String** | Không được rỗng | Tên nhân viên |
| **1.3** | caLam | **String** | Nha | Ca làm việc |
| **1.4** | ngayLam | **LocalDate** |  | Ngày làm việc |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| **2.1** | getMaLichLamViec() | **String** |  | Lấy mã thiết bị y tế |
| **2.2** | setMaLichLamViec(String maLichLamViec) | **void** | Kiểm tra | Thiết lập mã thiết bị y tế |
| **2.3** | getNhanVien() | **String** |  | Lấy tên loại thiết bị y tế |
| **2.4** | setNhanVien(String nhanVien) | **void** | Không được rỗng | Throw exception “Nhân viên không hợp lệ” nếu tham số nhanVien rỗng |
| **2.5** | getCaLam() | **String** |  | Lấy ca làm việc |
| **2.6** | setCaLam(String caLam) | **void** | Tham số caLam gồm các giá trị là: “Ca 1”, “Ca 2”  Không được rỗng | Throw exception “Ca làm việc không hợp lệ” nếu tham số ngayLam rỗng hoặc sai giá trị |
| **2.7** | getNgayLam() | **LocalDate** |  | Lấy ngày làm việc |
| **2.8** | setNgayLam(LocalDate ngayLam) | **void** | Kiểm tra tham số ngayLam đúng định dạng dd/mm/yyyy | Throw exception “Ngày làm không hợp lệ” nếu tham số ngayLam rỗng hoặc sai định dạng |
| **3** | **Viết các constructor** |  |  |  |
| **4** | **Viết phương thức** toString() | **String** | public LichLamViec(){}  public LichLamViec(String maLichLamViec, String nhanVien, String caLam, LocalDate ngayLam) { this.maLichLamViec = maLichLamViec; this.nhanVien = nhanVien; this.caLam = caLam; this.ngayLam = ngayLam; }  public LichLamViec(LichLamViec other) { this.maLichLamViec = other.maLichLamViec; this.nhanVien = other.nhanVien; this.caLam = other.caLam; this.ngayLam = other.ngayLam; } | Trả về chuỗi chứa giá trị các thuộc tính của loại thiết bị y tế |

***Link nhật ký:*** [***https://github.com/Nooba2612/tai-lieu-ptud***](https://github.com/Nooba2612/tai-lieu-ptud)