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

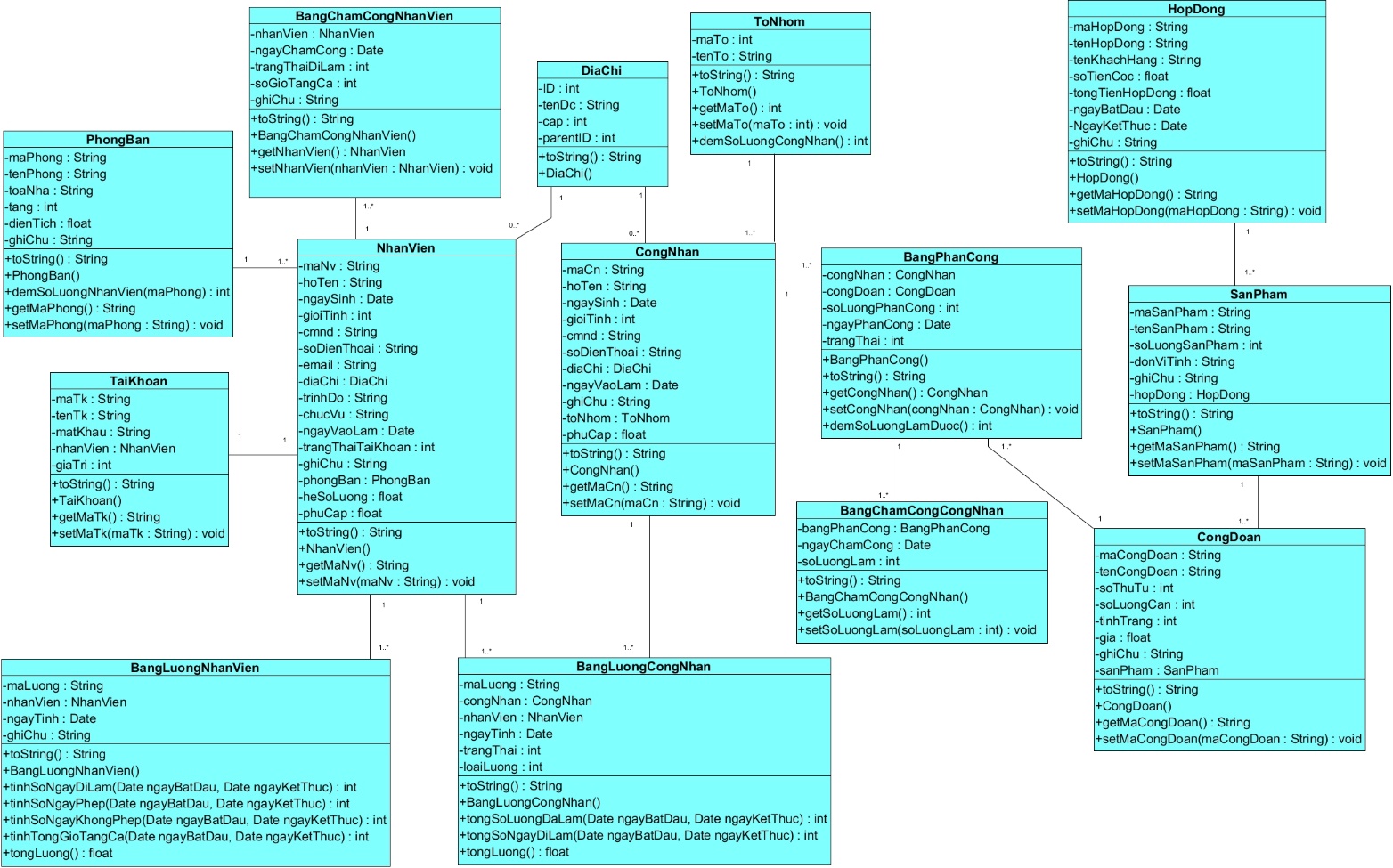
Nhóm 07 - Thành viên nhóm

1. Nguyễn Thành An (Nhóm trưởng)
2. Trần Vũ Minh Nhật
3. Nguyễn Hồng Quân
4. Nguyễn Tuấn Hùng

*Tên ứng dụng:* **QUẢN LÝ LƯƠNG SẢN PHẨM**

Thời gian thực hiện: Từ 22/08/2023 đến 30/10/2023 (10 tuần)

# **Mô hình lớp thực thể (Class Diagram)**

****

Hình 1‑1: Mô hình lớp thực thể

# **Đặc tả sơ đồ lớp:**

## Thực thể CongDoan

Table 1: Thực thể công đoạn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maCongDoan | **String** | Không được trống, gồm 8 kí tự trong đó 6 kí tự là số (\*) từ 0 đến 9 theo định dạng CD\*\*\*\*\*\* | Khóa chính, phát sinh mã tự động |
| 1.2 | tenCongDoan | **String** | Không được bỏ trống, không chứa kí tự đặc biệt \_\*@/ và không chứa kí tự số |  |
| 1.3 | soThuTu | **int** | Phải > 0 |  |
| 1.4 | soLuongCan | **int** | Phải > 0 |  |
| 1.5 | tinhTrang | **int** | Có 3 trạng thái: chưa hoàn thành, đang làm, hoàn thành. |  |
| 1.6 | gia | **float** | Phải >= 0 |  |
| 1.7 | ghiChu | **String** |  |  |
| 1.8 | sanPham | **SanPham** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | getMaCongDoan() | **String** |  |  |
| 2.2 | getTenCongDoan() | **String** |  |  |
| 2.3 | getSoThuTu() | **int** |  |  |
| 2.4 | getSoLuongCan() | **int** |  |  |
| 2.5 | getTinhTrang() | **int** |  |  |
| 2.6 | getGia() | **float** |  |  |
| 2.7 | getGhiChu() | **String** |  |  |
| 2.8 | getSanPham() | **SanPham** |  |  |
| 2.9 | setMaCongDoan(String maCongDoan) | **void** | Không được trống, gồm 8 kí tự trong đó 6 kí tự là số (\*) từ 0 đến 9 theo định dạng CD\*\*\*\*\*\* | Phát sinh tự động, throw exception “Mã công đoạn phải theo dạng CD\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9 |
| 2.10 | setTenCongDoan(String tenCongDoan) | **void** | Không được bỏ trống, không chứa kí tự đặc biệt \_\*@/ và không chứa kí tự số | Throw exception “Tên công đoạn không được rỗng và không chứa các kí tự đặc biệt \_\*@/” nếu tham số tenCongDoan không hợp lệ |
| 2.11 | setSoThuTu(int soThuTu) | **void** | Phải > 0 | Không được bỏ trống, throw exception “Số thứ tự phải > 0” nếu tham số soThuTu không hợp lệ |
| 2.12 | setSoLuongCan(int soLuongCan) | **void** | Phải > 0 | Không được bỏ trống, throw exception “Số lượng cần phải > 0” nếu tham số soLuongCan không hợp lệ |
| 2.13 | setTinhTrang(int tinhTrang) | **void** |  |  |
| 2.14 | setGia(float gia) | **void** | Phải >= 0 | Không được bỏ trống, throw exception “Giá tiền của công đoạn phải >= 0” nếu tham số gia không hợp lệ |
| 2.15 | setGhiChu(String ghiChu) | **void** |  |  |
| 2.16 | setSanPham(SanPham sanPham) | **void** | Kiểm tra than số sp không được null | throw exception “MaSP không không được null” nếu tham số sp không hợp lệ |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | CongDoan() |  |  | Contructor mặc nhiên, phát sinh tự động |
| 3.2 | CongDoan(String maCongDoan) |  |  |  |
| 3.3 | CongDoan(String maCongDoan, String tenCongDoan, int soThuTu, int soLuongCan, int tinhTrang, float gia, String ghiChu, SanPham sanPham) |  |  | Contructor đầy đủ tham số, phát sinh tự động |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của công đoạn |

## Thực thể SanPham:

Table 2: Thực thể sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maSanPham | **String** | Không được trống, gồm 8 kí tự trong đó 6 kí tự là số (\*) từ 0 đến 9 theo định dạng SP\*\*\*\*\*\* | Khóa chính, phát sinh mã tự động |
| 1.2 | tenSanPham | **String** | Không được bỏ trống, không chứa kí tự đặc biệt \_\*@/ và không chứa kí tự số |  |
| 1.3 | soLuongSanPham | **int** | Phải > 0 |  |
| 1.4 | donViTinh | **String** | Cái |  |
| 1.5 | ghiChu | **String** |  |  |
| 1.6 | hopDong | **HopDong** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | getMaSanPham() | **String** |  |  |
| 2.2 | getTenSanPham() | **String** |  |  |
| 2.3 | getSoLuongSanPham() | **int** |  |  |
| 2.4 | getDonViTinh() | **String** |  |  |
| 2.5 | getGhiChu() | **String** |  |  |
| 2.6 | getHopDong() | **HopDong** |  |  |
| 2.7 | setMaSanPham(String maSanPham) | **void** | Không được trống, gồm 8 kí tự trong đó 6 kí tự là số (\*) từ 0 đến 9 theo định dạng SP\*\*\*\*\*\* | Phát sinh tự động, throw exception “Mã sản phẩm phải theo dạng SP\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9 |
| 2.8 | setTenSanPham(String tenSanPham) | **void** | Không được bỏ trống, không chứa kí tự đặc biệt \_\*@/ và không chứa kí tự số | Throw exception “Tên sản phẩm không được rỗng và không chứa các kí tự đặc biệt \_\*@/” nếu tham số tenSanPham không hợp lệ |
| 2.9 | setSoLuongSanPham(int soLuongSanPham) | **void** | Phải > 0 | Throw exception “Số lượng sản phẩm phải >0” nếu tham số soLuongSanPham không hợp lệ |
| 2.10 | setDonViTinh(String donViTinh) | **void** | Cái |  |
| 2.11 | setGhiChu(String ghiChu) | **void** |  |  |
| 2.12 | setHopDong (HopDong hopDong) | **void** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | SanPham() |  |  | Contructor mặc nhiên, phát sinh tự động |
| 3.2 | SanPham(String maSanPham) |  |  |  |
| 3.3 | SanPham(String maSanPham, String tenSanPham, int soLuongSanPham, String donViTinh, String ghiChu, HopDong hopDong) |  |  | Contructor đầy đủ tham số, phát sinh tự động |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của sản phẩm |

## Thực thể ToNhom:

Table 3: Thực thể tổ nhóm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maTo | **int** |  | Khóa chính, phát sinh mã tự động |
| 1.2 | tenTo | **String** | Không được bỏ trống, không chứa kí tự đặc biệt \_\*@/ và không chứa kí tự số | Throw exception “Tên tổ không được rỗng và không chứa các kí tự đặc biệt \_\*@/” nếu tham số tenTo không hợp lệ |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | getMaTo() | **int** |  |  |
| 2.2 | getTenTo | **String** |  |  |
| 2.3 | setMaTo(int maTo) | **void** |  | Phát sinh mã tăng tự động |
| 2.4 | setTenTo(String tenTo) | **void** | Không được bỏ trống, không chứa kí tự đặc biệt \_\*@/ và không chứa kí tự số | throw exception “Tên tổ không không được rỗng” nếu tham số tento không hợp lệ |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | ToNhom() |  |  | Contructor mặc nhiên, phát sinh tự động |
|  | ToNhom(int maTo) |  |  |  |
|  | ToNhom(int maTo, String tenTo) |  |  | Contructor đầy đủ tham số, phát sinh tự động |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của tổ nhóm |

## Thực Thể BangChamCongNhanVien:

Table 4: thực thể chấm công nhân viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | ngayChamCong | **Date** | Không rỗng |  |
| 1.2 | trangThaiDiLam | **Int** | Không rỗng | 1 là có, 2 là không |
| 1.3 | soGioTangCa | **int** | Không rỗng |  |
| 1.4 | nhanVien | **NhanVien** |  |  |
| 1.5 | ghiChu | **String** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | setNgayChamCong (Date NgayChamCong) | **void** | Kiểm tra tham số ngayChamCong,không được rỗng | throw exception “Ngày Chấm Công không hợp lệ” nếu tham số ngayChamCong không hợp lệ |
| 2.2 | setTrangThaiDiLam(int TrangThaiDiLam) | **void** | Kiểm tra tham số trangThaiDiLam,không được rỗng | throw exception “Ca Làm không hợp lệ” nếu tham số trangThaiDiLam không hợp lệ |
| 2.3 | setGioTangCa(int gioTangCa) | **void** | Kiểm tra tham số gioTangCa không được rỗng | throw exception “Giờ tăng ca không hợp lệ” nếu tham số gioTangCa rỗng |
| 2.4 | setNhanVien(NhanVien nhanVien) | **void** |  |  |
| 2.5 | setGhiChu(String ghiChu) | **void** |  |  |
| 2.6 | getNgayChamCong() | **Date** |  |  |
| 2.7 | getTrangThaiDiLam() | **Int** |  |  |
| 2.8 | getSoGiaTangCa() | **int** |  |  |
| 2.9 | getNhanVien() | **NhanVien** |  |  |
| 2.10 | getGhiChu() | **String** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | **BangChamCongNhanVien()** | **Constructor** |  | Contructor mặc nhiên |
| 3.2 | **BangChamCongNhanVien(**NhanVien nhanVien, Date ngayChamCong, Int trangThaiDiLam, Int soGioTangCa, String ghiChu**)** | **Constuctor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của bảng chấm công nhân viên |

## Thực thể BangChamCongCongNhan

Table 5: thực thể bảng chấm công công nhân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
|  | maChamCong | **String** | Không được rỗng, gồm 8 ký tự theo dạng CCCN\*\*\*\* (với \* là các kí tự số từ 0 đến 9) |  |
|  | congNhan | **CongNhan** |  |  |
|  | ngayChamCong | **Date** | Phải trước hoặc bằng ngày hiện tại |  |
|  | soLuongLam | **int** | Không được rỗng, phải >= 0 |  |
|  | trangThai | **int** |  | 1 là hoàn thanh, 2 là chưa hoàn thành |
|  | congDoan | **CongDoan** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
|  | setMaChamCong | **void** | Kiểm tra tham số maChamCong không được rỗng, gồm 8 ký tự theo dạng CCCN\*\*\*\* (với \* là các kí tự số từ 0 đến 9) | Phát sinh mã tự động Throw exception “Mã bảng lương phải theo dạng CCCN\*\*\*\* với \* là các kí tự số \* đầu tiền từ 0 đến 9 |
|  | setCongNhan | **void** | Kiểm tra tham số congNhan không được rỗng | Throw exception “Công nhân không được để trống “ khi người dùng bỏ trống |
|  | setNgayChamCong | **void** | Kiểm tra tham số ngayChamCong không được để trống | Throw exception “Ngày chấm công không hợp lệ” nếu tham số ngayChamCong không hợp lệ. |
|  | setSoLuongLam | **void** | Kiểm tra tham số soLuongLam phải >=0 | Throw exception “số lượng phải >= 0” khi người dùng nhập vào số < 0 |
|  | setTrangThai | **void** | Kiểm tra tham số trangThai không được rỗng | Throw exception “Trạng thái không hợp lệ” khi tham số trangThanh không hợp lệ. |
|  | setCongDoan | **void** | Kiểm tra tham số congDoan không được rỗng | Throw exception “Công đoạn không hợp lệ” khi người dùng nhập vào không hợp lệ |
|  | getMaChamCong() | **String** |  |  |
|  | getCongNhan() | **CongNhan** |  |  |
|  | getNgayChamCong() | **Date** |  |  |
|  | getSoLungLam() | **Int** |  |  |
|  | getTrangThai() | **Int** |  |  |
|  | getCongDoan() | **CongDoan** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | BangChamCongCongNhan() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | BangChamCongCongNhan (String maChamCong, CongNhan congNhan, Date ngayChamCong, int soLuongLam, int trangThanh, CongDoan congDoan) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của bảng chấm công công nhân |

## Thực Thể HopDong

Table 6: thực thể hợp đồng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
|  | maHopDong | **String** | Không được rỗng, gồm 8 ký tự theo dạng HD\*\*\*\*\*\* (với \* là các kí tự số từ 0 đến 9) | Khóa chính |
|  | tenHopDong | **String** | Không được rỗng |  |
|  | tenKhachHang | **String** | Khoong được rỗng |  |
|  | soTienCoc | **Float** | Không được rỗng, phải >= 0 |  |
|  | tongTienHopDong | **Float** | Không được rỗng phải >= 0 |  |
|  | ngayBatDau | **Date** | Phải trước hoặc bằng ngày hiện tại |  |
|  | ngayKetThuc | **Date** | Phải sau ngày hiện tại |  |
|  | ghiChu | **String** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
|  | setMaHopDong(String maHopDong) | **void** | Kiểm tra tham số maHopDong không được rỗng, gồm 8 ký tự theo dạng HD\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9. | Phát sinh tự động, Throw exception “Mã hợp đồng phải theo dạng HD\*\*\*\*\*\* với \* là các kí tự số x đầu tiền từ 0 đến 9. |
|  | setTenHopDong(String tenHopDong) | **void** | Kiểm tra tham số tenHopDong không được rỗng | Throw exception “Tên hợp đồng không được để trống “ khi người dùng bỏ trống |
|  | setTenKhachHang(String tenKhachHang) | **void** | Kiểm tra tham số tenKhachHang không được để trống | Throw exception “tên khách hàng không được để rỗng” khi người dùng bỏ trống. |
|  | SetSoTienCoc(double soTienCoc) | **void** | Kiểm tra tham số soTienCoc không được để trống, tiền cọc phải >= 0 | Throw exception “Tiền cọc phải >= 0” khi người dùng nhập vào số < 0 |
|  | setTongTienHopDong(double tongTien) | **void** | Kiểm tra tham số tổng tongTien không được < 0 | Throw exception “Tổng tiền phải >= 0” |
|  | setNgayBatDau(Date ngayKiKet) | **void** | Kiểm tra tham số ngayBatDau phải là ngày trước hoặc ngày hiện tại | Throw exception “Ngày bắt đầu phải là ngày trước hoặc bằng hiện tại” khi người dùng nhập vào ngày sau hiện tại |
|  | setNgayKetThuc(Date ngayKetThuc) | **void** | Kiểm tra tham số ngayKetThuckhông được được trước hoặc bằng ngày hiện tại | Throw exception “ngày kết thúc phải sau ngày hiện tại” khi tham số là ngày hiện tại trở vê trước. |
|  | setGhiChu(String ghiChu) | **void** |  |  |
|  | getMaHopDong | **String** |  |  |
|  | getTenHopDong() | **String** |  |  |
|  | getTenKhachHang() | **String** |  |  |
|  | getSoTienCoc() | **Float** |  |  |
|  | getTongTienHopDong() | **Float** |  |  |
|  | getNgayBatDau() | **Date** |  |  |
|  | getNgayKetThuc() | **Date** |  |  |
|  | getGhiChu() | **String** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | HopDong() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | HopDong(String maHopDong, String tenHopDong, String tenKhachHang, float soTienCoc, float tongTienHopDong, Date ngayBatDau, Date ngayKetThuc, String ghiChu) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của hợp đồng |

## Thực thể BangLuongNhanVien

Table 7: thực thể bảng lương nhân viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maLuong | **String** | Không được rỗng, gồm 8 ký tự theo dạng L\*\*\*\*\*\* (với \* là các kí tự số từ 0 đến 9) | Khóa chính |
| 1.2 | nhanVien | **NhanVien** | Không được rỗng |  |
| 1.3 | ngayTinh | **Date** | Không được rỗng |  |
| 1.4 | ghiChu | **String** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | setMaLuong(String maLuong) | **void** | Kiểm tra tham số maLuong không được rỗng, gồm 8 ký tự theo dạng L\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9. | Phát sinh tự động, Throw exception “Mã lương phải theo dạng L\*\*\*\*\*\* với \* là các kí tự số x đầu tiền từ 0 đến 9. |
| 2.2 | setNhanVien(String NhanVien) | **void** | Kiểm tra tham số nhanVien không được rỗng | Throw exception “Nhân viên không được để trống “ khi người dùng bỏ trống |
| 2.3 | setNgayTinh(String ngayTinh) | **void** | Kiểm tra tham số ngayTinh không được để trống | Throw exception “ngayTinh không được để rỗng” khi người dùng bỏ trống. |
| 2.4 | setGhiChu(String ghiChu) | **void** |  |  |
| 2.5 | getMaLuong | **String** |  |  |
| 2.6 | getNhanVien() | **NhanVien** |  |  |
| 2.7 | getNgayTinh() | **Date** |  |  |
| 2.8 | getGhiChu() | **String** |  |  |
| 2.9 | tinhSoNgayDiLam(Date ngayBatDau, Date ngayKetThuc) | **int** |  | Trả về tổng số ngày đi làm trong khoảng thời gian theo tham số truyền |
| 2.10 | tinhSoNgayPhep(Date ngayBatDau, Date ngayKetThuc) | **int** |  | Trả về tổng số ngày phép trong khoảng thời gian theo tham số truyền |
| 2.11 | tinhSoNgayKhongPhep(Date ngayBatDau, Date ngayKetThuc | **int** |  | Trả về tổng số ngày không phép trong khoảng thời gian theo tham số truyền |
| 2.12 | tinhTongGioTangCa(Date ngayBatDau, Date ngayKetThuc) | **int** |  | Trả về tổng số giờ tăng ca trong khoảng thời gian theo tham số truyền |
| 2.13 | tongLuong() | **int** |  | Tính tổng lương |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | BangLuongNhanVien() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | BangLuongNhanVien(String maLuong, NhanVien nhanVien, Date ngayTinh, String ghiChu) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của bảng lương nhân viên |

## Thực thể BangLuongCongNhan

Table 8: thực thể bảng lương công nhân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maLuong | **String** | Không được rỗng, gồm 8 ký tự theo dạng L\*\*\*\*\*\* (với \* là các kí tự số từ 0 đến 9) | Khóa chính |
| 1.2 | congNhan | **CongNhan** | Không được rỗng |  |
| 1.3 | nhanVien | **NhanVien** | Khoong được rỗng |  |
| 1.4 | ngayTinh | **Date** | Không được rỗng, phải >= 0 |  |
| 1.5 | trangThai | **int** | Phải sau ngày hiện tại |  |
| 1.6 | loaiLuong | **int** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | setMaLuong(String maLuong) | **void** | Kiểm tra tham số maLuong không được rỗng, gồm 8 ký tự theo dạng L\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9. | Phát sinh tự động, Throw exception “Mã hợp đồng phải theo dạng L\*\*\*\*\*\* với \* là các kí tự số x đầu tiền từ 0 đến 9. |
| 2.2 | setCongNhan(CongNhan congNhan) | **void** | Kiểm tra tham số congNhan không được rỗng | Throw exception “congNhan không được để trống “ khi người dùng bỏ trống |
| 2.3 | setNhanVien(NhanVien nhanVien) | **void** | Kiểm tra tham số nhanVien không được để trống | Throw exception “Nhân viên không được để rỗng” khi người dùng bỏ trống. |
| 2.4 | setNgayTinh(double ngayTinh) | **void** | Kiểm tra tham số ngayTinh không được để trống | Throw exception “ngayTinh phải >= 0” khi người dùng nhập vào số < 0 |
| 2.5 | setTrangThai(int trangThai) | **void** | Kiểm tra tham số trạng thái không được rỗng | Throw exception “Trạng thái không được rỗng” khi người dùng để trống |
| 2.6 | setLoaiLuong(int loaiLuong) | **void** | Kiểm tra tham số loaiLuong không được để trống | Throw exception “loại lương không được để trống” khi người dùng để trống |
| 2.7 | getMaLuong() | **String** |  |  |
| 2.8 | getCongNhan() | **CongNhan** |  |  |
| 2.9 | getNhanVien() | **NhanVien** |  |  |
| 2.10 | getNgayTinh() | **Date** |  |  |
| 2.11 | getTrangThai() | **Int** |  |  |
| 2.12 | getLoaiLuong() | **Int** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | BangLuongCongNhan() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | BangLuongCongNhan(String maLuong, CongNhan congNhan, NhanVien nhanVien, Date ngayTinh, int trangThai, int loaiLuong) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của bảng lương công nhân |

## Thực thể DiaChi

Table 9: thực thể địa chỉ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | ID | **int** | Không được rỗng | Khóa chính |
| 1.2 | tenDc | **String** | Không được rỗng |  |
| 1.3 | cap | **int** | Không được rỗng |  |
| 1.4 | ParentID | **int** | Khoong được rỗng |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | setID(int ID) | **void** |  |  |
| 2.2 | setTenDc(String tenDc) | **void** | Kiểm tra tham số tenDC không được rỗng | Throw exception “Tên địa chỉ không được để trống “ khi người dùng bỏ trống |
| 2.3 | setCap (int cap) | **void** | không được để trống |  |
| 2.4 | setParentID(int parentID) | **void** |  |  |
| 2.5 | getID() |  |  |  |
| 2.6 | getTenDc() |  |  |  |
| 2.7 | getCap() |  |  |  |
| 2.8 | getParentID() |  |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | DiaChi() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | DiaChi(int ID, String tenDc, int cap,int ParentID) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của địa chỉ |

## Thực Thể CongNhan

Table 10: thực thể công nhân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maCn | **String** | Không được rỗng, gồm 8 ký tự theo dạng CN\*\*\*\*\*\* (với \* là các kí tự số từ 0 đến 9). | Khóa chính. |
| 1.2 | hoTen | **String** | Không được rỗng |  |
| 1.3 | ngaySinh | **Date** | Không được rỗng. Số tuổi phải lớn hơn 18. |  |
| 1.4 | gioiTinh | **Int** | Không được rỗng. 0 là nam, 1 là nữ. |  |
| 1.5 | cmnd | **String** | Không được rỗng. Tối đa 12 số. |  |
| 1.6 | sdt | **String** | Không được rỗng. Phải đủ 10 số. |  |
| 1.7 | diaChi | **DiaChi** | Không được rỗng. |  |
| 1.8 | ngayVaoLam | **Date** | Phải bé hơn hoặc bằng ngày hiện tại. |  |
| 1.9 | ghiChu | **String** |  |  |
| 1.10 | toNhom | **ToNhom** | Không để rỗng. |  |
| 1.11 | phuCap | **float** | phuCap >= 0 |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
|  | setMaCN(String maCn) | **void** | Kiểm tra tham số maCn không được rỗng, gồm 8 ký tự theo dạng CN\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9. | Phát sinh tự động, Throw exception “Mã công nhân phải theo dạng CN\*\*\*\*\*\* với \* là các kí tự số x đầu tiền từ 0 đến 9. |
|  | setHoTen(String hoTen) | **void** | Kiểm tra tham số hoTen không được rỗng | Throw exception “Tên công nhân không được để trống “ khi người dùng bỏ trống. |
|  | setNgaySinh(Date ngaySinh) | **void** | Kiểm tra tham số ngaySinh phải bé hơn hoặc bằng ngày hiện tại | Throw exception “Ngày sinh phải bé hơn hoặc bằng ngày hiện tại ” nhập ngày sau ngày hiện tại. |
|  | setGioiTinh(int gioiTinh) | **void** | Kiểm tra tham số gioiTinh không để trống và không nhập khác 0 và 1 | Throw exception “Không được để trống” khi người dùng bỏ trống và “Nhập 0 là nam, 1 là nữ” khi nhập khác 0 và 1. |
|  | setCmnd(String cmnd) | **void** | Kiểm tra tham số cmnd không được để trống, chỉ được 9 hoặc 12 số và phải là duy nhất. | Throw exception “Không được để trống” khi người nhập để trống, “Cmnd chỉ có 9 hoặc 12 số” khi người nhập nhập các kí tự không phải số hay nhập khác 9 hoặc 12 chữ số, “Cmnd bị trùng” khi người nhập nhập thông tin cmnd đã bị trùng. |
|  | setSoDienThoai(String soDienThoai) | **void** | Kiểm tra tham số soDienThoai không được để trống và số chữ số phải bằng 10. | Throw exception “Số điện thoại không được để trống” khi người không nhập và “Số điện thoại phải 10 chữ số” khi người nhập nhập các kí tự không phải số hoặc nhập khác 10 chữ số. |
|  | setDiaChi(DiaChi diaChi) | **void** | Kiểm tra tham số diaChi không được để trống. | Throw exception “Địa chỉ không được để trống”. |
|  | setNgayVaoLam(Date ngayVaoLam) | **void** | Kiểm tra tham số ngayVaoLam phải là ngày trước hoặc ngày hiện tại. | Throw exception “Ngày vào làm phải là ngày trước hoặc bằng hiện tại” khi người dùng nhập vào ngày sau hiện tại. |
|  | setGhiChu(String ghiChu) | **void** |  |  |
|  | setToNhom(ToNhom toNhom) | **void** | Kiểm tra tham số toNhom không được để trống. | Throw exception “Tổ không được để trống” khi người nhập để trống. |
|  | setPhuCap(float phuCap) | **void** | phuCap >= 0 |  |
|  | getPhuCap() | **float** |  |  |
|  | getMaCN() | **String** |  |  |
|  | getHoTen() | **String** |  |  |
|  | getNgaySinh() | **Date** |  |  |
|  | getGioiTinh() | **Int** |  |  |
|  | getCmnd() | **String** |  |  |
|  | getSoDienThoai() | **String** |  |  |
|  | getDiaChi() | **DiaChi** |  |  |
|  | getNgayVaoLam() | **Date** |  |  |
|  | getGhiChu() | **String** |  |  |
|  | getToNhom () | **ToNhom** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | CongNhan() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | CongNhan(String maCN, String hoTen, Date ngaySinh, int gioiTinh, String cmnd, String soDienThoai, DiaChi diaChi, Date ngayVaoLam, String ghiChu, ToNhom toNhom, float phuCap) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của công nhân |

## Thực Thể NhanVien

Table 11: thực thể nhân viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maNv | **String** | Không được rỗng, gồm 8 ký tự theo dạng NV\*\*\*\*\*\* (với \* là các kí tự số từ 0 đến 9). | Khóa chính. |
| 1.2 | hoTen | **String** | Không được rỗng |  |
| 1.3 | ngaySinh | **Date** | Không được rỗng. Số tuổi phải lớn hơn 18. |  |
| 1.4 | gioiTinh | **Int** | Không được rỗng. 0 là nam, 1 là nữ. |  |
| 1.5 | cmnd | **String** | Không được rỗng. Tối đa 12 số. |  |
| 1.6 | sdt | **String** | Không được rỗng. Phải đủ 10 số. |  |
| 1.7 | email | **String** |  |  |
| 1.8 | diaChi | **DiaChi** | Không được rỗng. |  |
| 1.9 | trinhDo | **String** |  |  |
| 1.10 | chucVu | **String** | Không được rỗng. |  |
| 1.11 | ngayVaoLam | **Date** | Phải bé hơn hoặc bằng ngày hiện tại. |  |
| 1.12 | trangThaiTaiKhoan | **int** | Không được rỗng. 0 là không có tài khoản, 1 là đã có tài khoản. |  |
| 1.13 | ghiChu | **String** |  |  |
| 1.14 | phongBan | **PhongBan** | Không để rỗng. |  |
| 1.15 | heSoLuong | **float** | Không để rỗng. |  |
| 1.16 | phuCap | **float** | phuCap >= 0 |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
|  | setMaNv(String maNv) | **void** | Kiểm tra tham số maNv không được rỗng, gồm 8 ký tự theo dạng NV\*\*\*\*\*\* với \* là các kí tự số từ 0 đến 9. | Phát sinh tự động, Throw exception “Mã hợp đồng phải theo dạng NV\*\*\*\*\*\* với \* là các kí tự số x đầu tiền từ 0 đến 9. |
|  | setHoTen(String hoTen) | **void** | Kiểm tra tham số hoTen không được rỗng | Throw exception “Tên nhân viên không được để trống “ khi người dùng bỏ trống. |
|  | setNgaySinh(Date ngaySinh) | **void** | Kiểm tra tham số ngaySinh phải bé hơn hoặc bằng ngày hiện tại | Throw exception “Ngày sinh phải bé hơn hoặc bằng ngày hiện tại ” nhập ngày sau ngày hiện tại. |
|  | setGioiTinh(int gioiTinh) | **void** | Kiểm tra tham số gioiTinh không để trống và không nhập khác 0 và 1 | Throw exception “Không được để trống” khi người dùng bỏ trống và “Nhập 0 là nam, 1 là nữ” khi nhập khác 0 và 1. |
|  | setCmnd(String cmnd) | **void** | Kiểm tra tham số cmnd không được để trống, chỉ được 9 hoặc 12 số và phải là duy nhất. | Throw exception “Không được để trống” khi người nhập để trống, “Cmnd chỉ có 9 hoặc 12 số” khi người nhập nhập các kí tự không phải số hay nhập khác 9 hoặc 12 chữ số, “Cmnd bị trùng” khi người nhập nhập thông tin cmnd đã bị trùng. |
|  | setSoDienThoai(String soDienThoai) | **void** | Kiểm tra tham số soDienThoai không được để trống và số chữ số phải bằng 10. | Throw exception “Số điện thoại không được để trống” khi người không nhập và “Số điện thoại phải 10 chữ số” khi người nhập nhập các kí tự không phải số hoặc nhập khác 10 chữ số. |
|  | setEmail(String email) | **void** |  |  |
|  | setDiaChi(DiaChi diaChi) | **void** | Kiểm tra tham số diaChi không được để trống. | Throw exception “Địa chỉ không được để trống”. |
|  | setTrinhDo(String trinhDo) | **void** |  |  |
|  | setChucVu(String chucVu) | **void** | Kiểm tra tham số chức vụ không được để trống. | Throw exception “Chức vụ không được để trống”. |
|  | setNgayVaoLam(Date ngayVaoLam) | **void** | Kiểm tra tham số ngayVaoLam phải là ngày trước hoặc ngày hiện tại. | Throw exception “Ngày vào làm phải là ngày trước hoặc bằng hiện tại” khi người dùng nhập vào ngày sau hiện tại. |
|  | setTrangThaiTaiKhoan(int trangThaiTaiKhoan) | **void** | Kiểm tra tham số trangThaiTaikhoan không được để trống và phải bằng 0 hoặc 1. | Throw exception “Trạng thái tài khoản không được để trống” khi người nhập để trống và “Phải bằng 0 nếu chưa có tài khoản va 1 nếu đã có tài khoản” khi người nhập nhập số khác 0 và 1. |
|  | setGhiChu(String ghiChu) | **void** |  |  |
|  | setPhongBan(CaLam caLam) | **void** | Kiểm tra tham số phongBan không được để trống. | Throw exception “Phòng ban không được để trống” khi người nhập để trống. |
|  | setHeSoLuong(float heSoLuong) | **void** | Kiểm tra tham số heSoLuong không được để trống. | Throw exception “Hệ số lương không được để trống” khi người nhập để trống. |
|  | setPhuCap(float phuCap) | **void** | phuCap >= 0 |  |
|  | getPhuCap() | **float** |  |  |
|  | getMaNv() | **String** |  |  |
|  | getHoTen() | **String** |  |  |
|  | getNgaySinh() | **Date** |  |  |
|  | getGioiTinh() | **Int** |  |  |
|  | getCmnd() | **String** |  |  |
|  | getSoDienThoai() | **String** |  |  |
|  | getEmail() | **String** |  |  |
|  | getDiaChi() | **DiaChi** |  |  |
|  | getTrinhDo() | **String** |  |  |
|  | getChucVu() | **String** |  |  |
|  | getNgayVaoLam() | **Date** |  |  |
|  | getTrangThaiTaiKhoan() | **int** |  |  |
|  | getGhiChu() | **String** |  |  |
|  | getPhongBan() | **PhongBan** |  |  |
|  | getHeSoLuong() | **float** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | NhanVien() | **Constructor** |  | Constructor mặc nhiên |
| 3.2 | NhanVien(String maNv, String hoTen, Date ngaySinh, int gioiTinh, String cmnd, String soDienThoai, DiaChi diaChi, String trinhDo, String chucVu, Date ngayVaoLam, String ghiChu, PhongBan phongBan, float heSoLuong, float phuCap) | **Constructor** |  | Constructor có đầy đủ tham số |
| **4** | **Viết phương thức toString()** | **String** |  | Trả về chuỗi chứa giá trị các thuộc tính của nhân viên |

## Thực thể TaiKhoan:

Table 12: thực thể tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maTk | **String** | Khóa chính, là mã của nhân viên. |  |
| 1.2 | tenTk | **String** | Không được để rỗng. |  |
| 1.3 | maKhau | **String** | Không được để rỗng, không sử dụng các kí tự đặc biệt và phải đủ 8 kí tự. |  |
| 1.4 | nhanVien | **NhanVien** | Không được để rỗng. |  |
| 1.5 | giaTri | **int** | Không được để rỗng. Các giá trị sẽ chia thành các loại tài khoản khác nhau. | 0 là admin, 1 là người quản lý nhân sự, 2 là nhân viên kế toán, 3 là quản lí dự án hợp đồng, 4 là nhân viên chấm công. |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
|  | setMaTk(String maTk) | **void** | Mã tài khoản là mã nhân viên |  |
|  | setTenTk(String hoTen) | **void** | Kiểm tra tham số tenTK không được rỗng. | Throw exception “Tên tài khoản không được để trống “ khi người dùng nhập ngày sau ngày hiện tại |
|  | setMatKhau(String matKhau) | **void** | Kiểm tra tham số matKhau không được rỗng, không sử dụng các kí tự đặc biệt và phải đủ 8 kí tự. | Throw exception “Mật khẩu không được để rỗng. ” khi người dùng bỏ trống, “Bao gồm 8 kí tự, chữ in hoa, chữ thường và số.” khi người dùng nhập quá 8 kí tự hoặc dùng các kí tự đặc biệt |
|  | setNhanVien(NhanVien nhanVien) | **void** | Kiểm tra tham số nhanVien không để trống. | Throw exception “Không được để trống” khi người dùng bỏ trống. |
|  | setGiaTri(int giaTri) | **void** | Kiểm tra tham số giaTri không được để trống và phải nhập trong khoảng từ 0 tới 4. | Throw exception “Không được để trống” khi người nhập để trống, “Nhập số từ 0 đến 4, 0 là admin, 1 là người quản lý nhân sự, 2 là nhân viên kế toán, 3 là quản lí dự án hợp đồng, 4 là nhân viên chấm công.” khi người nhập nhập các kí tự không phải số trong khoảng từ 0 đến 4. |
|  | getMaTk() | **String** |  |  |
|  | getTenTk() | **String** |  |  |
|  | getMatKhau() | **String** |  |  |
|  | getNhanVien() | **NhanVien** |  |  |
|  | getGiaTri() | **int** |  |  |
|  | demSoLuongNhanVien(maPhong) | **int** |  | Trả về tổng số lượng nhân viên theo mã phòng |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | TaiKhoan() |  |  | Contructor mặc nhiên, phát sinh tự động |
| 3.2 | TaiKhoan(String maTk, String matKhau) |  |  |  |
| 3.3 | TaiKhoan(String maTk, String tenTk, String matKhau, NhanVien nhanVien, int giaTri) |  |  | Contructor đầy đủ tham số, phát sinh tự động |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của ca làm |

## Thực thể PhongBan:

Table 13: thực thể phòng ban

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | maPhong | **String** | Gồm 4 ký tự theo dạng PB\*\*\* (với \* là các kí tự số từ 0 đến 9). | Khóa chính. Phát sinh tự động. |
| 1.2 | tenPhong | **String** | Không được để rỗng. |  |
| 1.3 | toaNha | **String** | Không được để rỗng |  |
| 1.4 | tang | **Int** | Không được để rỗng |  |
| 1.5 | dienTich | **float** | dienTich >= 0 |  |
| 1.6 | ghiChu | **String** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | setMaPhong(String maPhong) | **void** | Phát sinh tự động. Gồm 4 ký tự theo dạng PB\*\*\* (với \* là các kí tự số từ 0 đến 9). |  |
| 2.2 | setTenPhong(String hoTen) | **void** | Kiểm tra tham số tenPhong không được rỗng. | Throw exception “Tên phòng ban không được để trống “ khi người dùng để rỗng. |
| 2.3 | setGhiChu(NhanVien nhanVien) | **void** |  |  |
| 2.3 | getMaPhong() | **String** |  |  |
| 2.5 | getTenPhong() | **String** |  |  |
| 2.6 | getGhiChu() | **String** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | PhongBan() |  |  | Contructor mặc nhiên, phát sinh tự động |
| 3.2 | TaiKhoan(String maPhong, string tenPhong, int sucChua, String ghiChu) |  |  |  |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của ca làm |

## Thực thể BangPhanCong:

Table 14: thực thể phòng ban

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| **1** | **Khai báo thuộc tính:** |  |  |  |
| 1.1 | congNhan | **CongNhan** |  | Khóa chính. |
| 1.2 | congDoan | **CongDoan** |  | Khóa chính |
| 1.3 | soLuongPhanCong | **int** | Không được để rỗng |  |
| 1.4 | ngayPhanCong | **Date** | Không được để rỗng |  |
| 1.5 | trangThai | **int** |  |  |
| **2** | **Viết các phương thức getter, setter** |  |  |  |
| 2.1 | setCongNha(CongNhan cn) | **void** |  |  |
| 2.2 | setCongDoan(CongDoan cd) | **void** |  |  |
| 2.3 | setSoLuongPhanCong(int soLuongPC) | **void** |  |  |
| 2.4 | setNgayPhanCong | **Void** |  |  |
| 2.5 | setTrangThai | **void** |  |  |
| 2.6 | getCongNhan() | **CongNhan** |  |  |
| 2.7 | getCongDoan() | **CongDoan** |  |  |
| 2.8 | getNgayPhanCong() | **Date** |  |  |
| 2.9 | getSoLuongPhanCong | **Int** |  |  |
| 2.10 | getTrangThai | **int** |  |  |
| **3** | **Viết các constructor** |  |  |  |
| 3.1 | BangPhanCong() |  |  | Contructor mặc nhiên, phát sinh tự động |
| 3.2 | BangPhanCong (CongNhan cn, CongDoan cd, int soLuongPC, Date ngayPC, int trangThai) |  |  |  |
| **4** | **Viết phương thức toString()** | **String** |  | trả về chuỗi chứa giá trị các thuộc tính của bảng phân công |

[Nhật ký nhóm 07](https://drive.google.com/drive/folders/1w7y2ZYwOdNdSnu3DLV7FgWHf84T4-gFC?usp=sharing)