**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG JAVA**

**ĐỀ TÀI: QUẢN LÝ CỬA HÀNG KINH DOANH NHẬP**

**VẬT LIỆU XÂY DỰNG**

**Giáo viên hướng dẫn: Trịnh Thị Xuân**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Hà Nội, năm 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG JAVA**

**ĐỀ TÀI: QUẢN LÝ DANH SÁCH PHIẾU NHẬP VẬT TƯ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2023**

**LỜI NÓI ĐẦU**

Sau quá trình học tập, rèn luyện kỹ năng, tư duy và cách xử lý các trường hợp trong môn lập trình hướng đối tượng Java, chúng em được cô giáo Trịnh Thị Xuân hướng dẫn rất tận tình. Nhóm chúng em đã ngồi bàn với nhau và cho ra một chương trình hướng đối tượng dựa trên ngôn ngữ lập trình Java. Đề tài của chúng em là Quản Lý Danh Sách Phiếu Nhập Vật Tư.

Nhóm chúng em cũng như toàn thể lớp CNTT 16-06 gửi lời cảm ơn chân thành nhất đến cô giáo Trịnh Thị Xuân, thầy cô khoa Công Nghệ Thông Tin của trường đã đồng hành cùng chúng em trong quãng thời gian qua.

**MỤC LỤC**

**BẢNG CÁC TỪ VIẾT TẮT**

**(Nếu có)**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| **1** | **CSDL** | **Cơ sở dữ liệu** |
| **2** |  |  |

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## 1.1 Lý thuyết biểu đồ lớp và các đối tượng.

### 1.1.1 Lớp Biểu đồ (class diagram)

Ý nghĩa trong lợp trình hướng đối tượng , một nhóm đối tượng chúng thuộc tính và phương thức tạo thành một lớp.Mối tương tác được thể hiện qua mối quan hệ.

Các lớp (bao gồm cả các thuộc tính và phương thức) cùng với các mối quan hệ sẽ tạo thành biểu đồ lớp. Biểu đồ lớp là một biểu đồ dạng mô hình tĩnh nhằm mô tả hướng nhìn tĩnh về một hệ thống bằng các khái niệm lớp, các thuộc tính, phương thức của lớp và mối quan hệ giữa chúng với nhau.

Trong phần này, tài liệu sẽ xem xét các vấn đề liên quan đến biểu diễn sơ đồ lớp trong UML. Cuối phần này sẽ là một bảng tổng kết các ký hiệu UML sử dụng trong sơ đồ lớp.

• Kí hiệu lớp: trong UML, mỗi lớp được biểu diễn bởi hình chữ nhật gồm 3 phần: tên lớp, các thuộc tính và các phương thức.

• Thuộc tính: các thuộc tính trong biểu đồ lớp được biểu diễn theo cấu trúc chung như sau: phạm\_vi tên : kiểu số\_đối\_tượng = mặc\_định (Giá\_ trị\_giới\_hạn )

Trong đó: phạm\_vi cho biết phạm vi truy nhập của thuộc tính. Có ba kiểu xác định thuộc tính phổ biến là:

+: thuộc tính kiểu public #:

thuộc tính kiểu protected Thu Thu Quan li Muon-Tra Sach Cap nhat <> <> Dang nhap Tim kiem -:

thuộc tính kiểu private. ~: thuộc tính được phép truy nhập tới từ các lớp trong cùng package Các phạm vi của thuộc tính có thể được biểu diễn dưới dạng ký hiệu (+, #, -, ~) hoặc biểu diễn dưới dạng các từ khoá (public, protected, private).

Tên: là xâu ký tự biểu diễn tên thuộc tính. kiểu: là kiểu dữ liệu của thuộc tính. số\_đối\_tượng: chỉ ra số đối tượng khai báo cho thuộc tính ứng với một mặc\_định: là giá trị khởi đầu mặc định (nếu có) của thuộc tính.

Giá\_ trị\_giới\_hạn: là giới hạn các giá trị cho thuộc tính (thông tin này không bắt buộc).

Ví dụ một khai báo thuộc tính đầy đủ: purchaseDate:Date[1] ="01-01-2000" (Saturday) • Phương thức (method): các phương thức trong UML được biểu diễn theo cấu trúc chung như sau [UNG]: phạm\_vi tên(danh\_s ách\_tham\_số): kiểu\_trả\_lại { ki ểu\_ph ương\_thức} Trong đó: visibility biểu diễn phạm vi cho phương thức.

Giống như đối với thuộc tính, có ba dạng kiểu xác định cơ bản cho phương thức là:

+: phương thức kiểu public - #:

phương thức kiểu protected - -:

phương thức kiểu private - ~: phương thức được phép truy nhập tới từ các lớp trong cùng package tên là xâu ký tự xác định tên của phương thức.

kiểu\_trả\_lại: chỉ ra kiểu giá trị trả về của phương thức. danh\_sách\_tham\_số: biểu diễn danh sách các tham số trong khai báo của phương thức. Mỗi tham số được biểu diễn dưới dạng chung: tên tham số:

kiểu giá trị = giá trị mặc định. kiểu\_phương\_thức: không bắt buộc, cho biết kiểu phương thức. Phương thức có thể nhận một trong các kiểu đặc biệt sau: abstract: phương thức kiểu trừu tượng query: phương thức kiểu truy vấn.

Ví dụ một khai báo phương thức cho một lớp: generatePurchaseList(prodID:int): String

**Các kiểu lớp UML**

* Lớp thực thể : là lớp đại diện cho các thực thể chứa thông tin về các đối tượng xác định nào đó.

- Lớp biên (lớp giao diện): là lớp nằm ở ranh giới giữa hệ thống với môi trường bên ngoài, thực hiện vai trò nhận yêu cầu trực tiếp từ các tác nhân và chuyển các yêu cầu đó cho các lớp bên trong hệ thống.

- Lớp điều khiển: thực hiện các chức năng điều khiển hoạt động của hệ thống ứng với các chức năng cụ thể nào đó với một nhóm các lớp biên hoặc lớp thực thể xác định.

|  |  |  |
| --- | --- | --- |
| STT | Kiểu lớp | Kí hiệu UML |
|  | Lớp thực thể |  |
| 2 | Lớp biên |  |
| 3 | Lớp điều khiển |  |

-Mối quan hệ Association : Là sự kết hợp giữa các lớp . Trong UML nhằm thể hiện sự ấn định liên kết với nhau giữa lớp với lớp .

Một chiều : Chỉ có một bên

Hai Chiều : Tức là có mua thì có bán , ta có thể suy rộng ra các đối tượng khác nữa

Mối quan hệ khái quát hóa ( Generalization ) : Khái quát giữa các mối quan hệ . Thể hiện tính kế thừa giữa các lớp.

Ta thấy ở bên trên lớp cơ sở là Human và 2 lớp dẫn xuất kế thừa vào là supplier và Employee

-Quan hệ cộng tác là một mối quan hệ một phần của lớp X có thể lấy một số thuộc tính của lớp Y, và lớy X có thể tồn tại độc lập với lớp Y :

Có lớp dẫn xuất là Supplier và lớp đối tượng Item , hai lớp này có mối quan hệ cộng tác và nối với nhau bằng hình con thoi . Lớp supplier lấy những thông tin thuộc tính từ lớp Item gồm name , code , unit ,price , supplier .

-Quan hệ tổng thể ( Composition ) : Một quan hệ biểu diễn tổng thể-bộ phận .

Bảng biểu đồ : Tổng kết các phần mô hình UML được sử dụng trong mô hình

Lớp , ý nghĩa và ký hiệu biểu đồ .

|  |  |  |
| --- | --- | --- |
| Phần tử mô hình | Ý nghĩa | Kí hiệu trong biểu đồ |
| Lớp (class) | Biểu diễn tên lớp, các thuộc tính và phương thức của lớp đó. |  |
| Quan hệ kiểu kết hợp | Biểu diễn quan hệ giữa 2 lớp độc lập, có liên quan đến nhau. |  |
| Quan hệ gộp | Biểu diễn quan hệ kiểu bộ phận – tổng thể. |  |
| Quan hệ khái quát hoá (kế thừa) | Lớp này thừa hưởng các thuộc tính – phương thức của lớp kia. |  |
| Quan hệ phụ thuộc | Các lớp phụ thuộc lẫn nhau trong hoạt động của hệ thống. |  |

Biểu đồ message Tuần Tự .

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại phản hồi | Chức năng | Thể hiện ra hình ảnh |
| 1 | Gọi (call) | Mô tả một lời gọi từ đối tượng này đến đối tượng kia. |  |
| 2 | Trả về (return) | Trả về giá trị ứng với lời gọi |  |
| 3 | Gửi (send) | Gửi một tín hiệu đến một đối tượng |  |
| 4 | Tạo (create) | Tạo một đối tượng |  |
| 5 | Huỷ (destroy) | Huỷ một đối tượng |  |

*Bảng biểu đồ message Tuần tự*

## 1.2 Phát biểu bài toán

Đề tài quản lý xây dựng, được xây dựng dựa trên ý tưởng quản lý vật liệu xây dựng của một công ty XYZ. Hiện nay quy trình quản lý, nhân viên vẫn nhập bằng tay .

Để giải quyết vấn đề trên , giám đốc siêu thị yêu cầu một phần mềm quản lý kho nhập vật tư

### 1.2.1 Yêu cầu bên cửa hàng

Bây giờ em đang xây dựng quản lý kinh doanh cửa hàng xây dựng : đối tượng lớp của em là khách hàng , nhân viên và item

Các lớp khác , nhà kho , hóa đơn

đối tượng quản lý danh sách các sản phẩm , danh sách các nhân viên , danh sách nhà cung cấp

**HÓA ĐƠN Nhập Vật tư(Chưa chỉnh)**

Số Hóa Đơn: 2023.09.0001 – Ngày lập: 01/09/2023

Chủ hộ: Nguyễn Thúy Long Số điện thoại: 0000011111

Căn hộ: 30-01 Email: zyx@gmail.com

Mã số thuế: 8888888888383883 Năm vào ở: 2018

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Mã dịch vụ** | **Tên Dịch vụ** | **Số lượng** | **Đơn giá** | **Thành Tiền** |
| 1 | XeMay | Xe máy tháng | 2 | 50.000 | **100.000** |
| 2 | Oto | Xe ô tô tháng tầng hầm | 1 | 1.100.000 | **1.100.000** |
|  | … | … | … | … | … |
| **Tổng**: | | | 3 |  | **1.200.000** |
| **Số tiền chiết khấu:** | | |  |  | 100.000 |
| **Số tiền phải nộp:** | | |  |  | **1.100.000** |

Bằng chữ: *Một triệu một trăm nghìn đồng*

|  |  |
| --- | --- |
| **Nhân Viên Thu Tiền**  Nguyễn Thị Hà  09877772368  Hant@gmail.com  Kế toán | **Khách Hàng**  (Ký) |

### 1.2.2 Các đối tượng cần quản lý

Nhân viên : name , contact ,code ,shift .

Department (cơ quan): name,address,tankage,inventory .

Item : code,name,unit,price ,supplier

Quản lý : name , contact , code ,shift

Nhà cung cấp :name,contact,listProduct

Hóa đơn : code , date time , supplier , membership , listProduct

Các chức năng xử lý gồm :

-Thêm sản phẩm và lưu danh sách

-Hiển thị sản phẩm

-Xuất file

-Thống kê (Tổng danh sách sản phẩm , nhân viên , sắp xếp danh sách hàng hóa)

-Tìm kiếm sản phẩm

-Cập nhật sản phẩm

Nhà cung cấp(Supplier).

-Thuộc tính : Tên: String.Địa chỉ: String.Contact: String.

Phương thức:

-Container.Getter, Setter.

Nhân viên(Employee).

-Thuộc tính:Tên: String.Mã nhân viên: String.Bộ phận công tác: Warehoure.

Phương thức:

-Container.Get,Setter.

**1.2.3 Quản lý hàng hoá**

Mặt hàng (Item).

Thuộc tính:

-Tên: String.Mã mặt hang: String.Đơn vị: doubleSố lượng: int.Mô tả: String.Đơn vị cung cấp: String.

Phương thức:

-Container.Getter, Setter.getPrice(Tính giá).

Phiếu nhập vật tư (PurchaseOrder).

Thuộc tính:

-Mã phiếu: String.Ngày nhập (Date): String.Nhà cung cấp: Supplier.Nhân viên thực hiện: Employee.Danh sách mặt hang: ArrayList<Item>.

Phương thức:

+ Container.Getter, Setter.getTotalPayment (Tính tổng giá trị build).

Danh sách phiếu nhập (PurchaseOrderList).

Thuộc tính:

-Danh sách phiếu nhập: ArrayList<PurchaseOrder>.

Phương thức:

+Thêm, Xoá phiếu nhập.Tìm kiếm.Tính tổng giá.

Kho hàng (Warehoure).

Thuộc tính:Tên: String.Địa chỉ: String.Sức chứa: int.

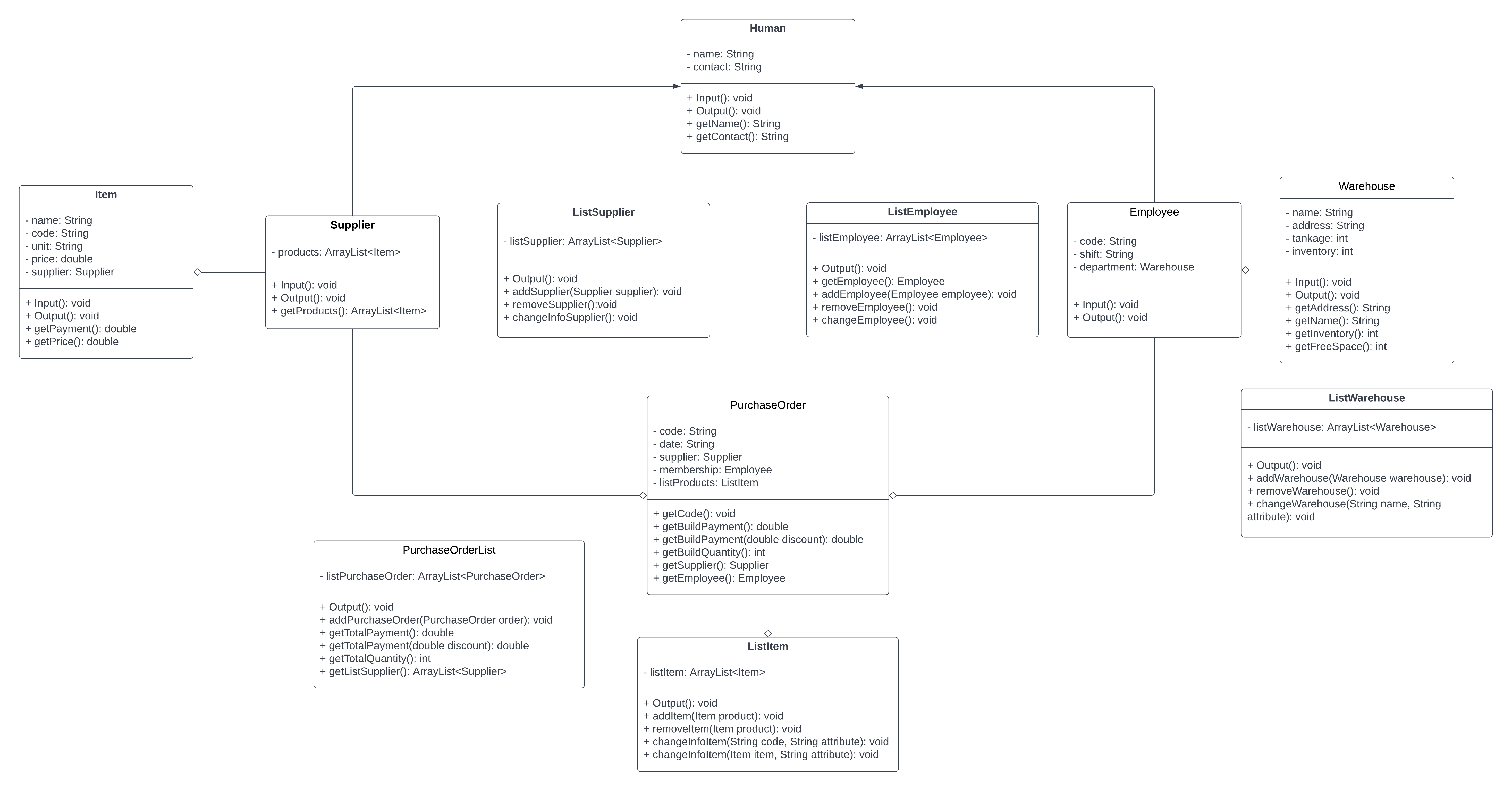
Phương thức:

+Container.Getter, Setter.

# CHƯƠNG 2: Các chức năng và thao tác

## 2.1 Sơ đồ các lớp và bài toán

### 2.1.1Các lớp mô tả bài toán được xây dựng như sau :



Lớp cơ sở ở đây là lớp Person có các thuộc tính : name, phoneNumber

Gồm các phương thức : Input(),output(),getName(),getContact()

Lớp dẫn xuất là Supplier và Emloyee

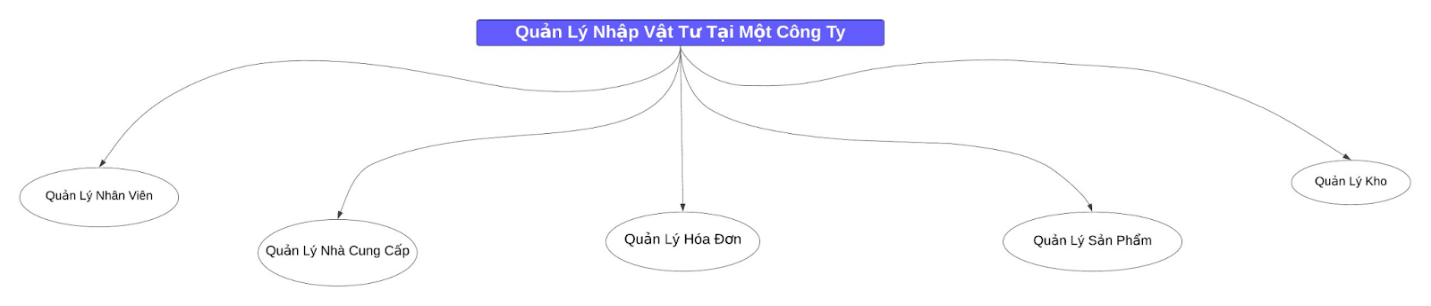
Supplier (cung cấp) có các thuộc tính được kế thừa từ Human

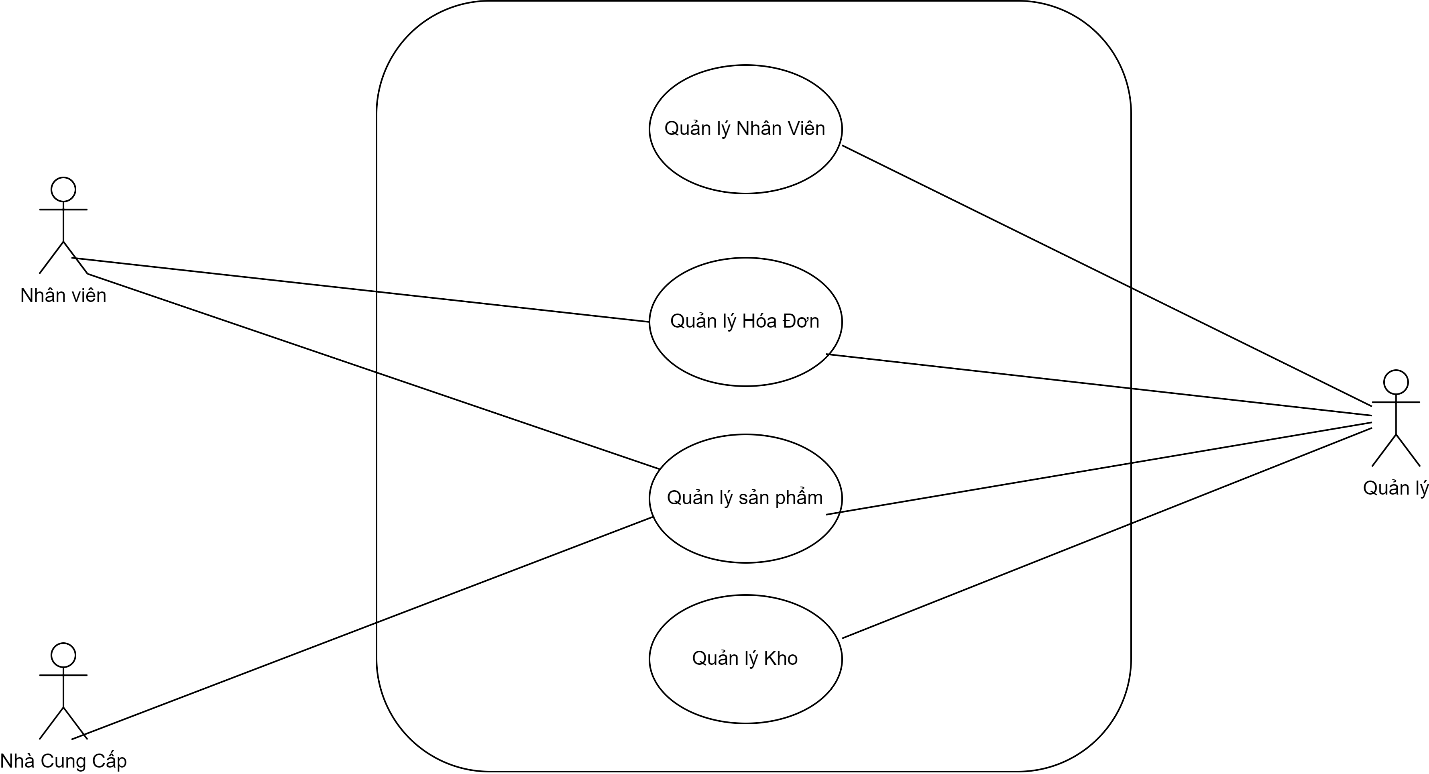
Phương thức : : Input(),output(),getProductList()

Employee (nhân viên) : code , shift(*ca làm)* ,department (địa điểm làm việc )

### 2.1.2 Sơ đồ use case tổng quát

Dưới đây là biểu đồ phân cấp chức năng :





### 2.1.3 Biểu đồ chức năng ( Sequence )

# Chương 3: Cài đặt chương trình

## 3.1 Xây dựng lớp Person

Lớp cơ sở có các thuộc tính chung để các lớp con kế thừa.

Code

package **quanlyvattuxaydung**;

import **java**.**util**.\*;

public class **Person** {

    protected **String** name;

    protected **String** PhoneNumber ;

    public **Person**(**String** name,**String** PhoneNumber){

        this.name = name;

        this.PhoneNumber = PhoneNumber;

    }

    public **Person**(){

    }

    public **Person**(**String** name){

        this.name = name;

    }

    public void **Input**(){

**Scanner** sc = new **Scanner**(**System**.in);

**System**.out.**println**("Nhap ho ten :");

        name = sc.**nextLine**();

**System**.out.**println**("Nhap so dien thoai :");

        PhoneNumber = sc.**nextLine**();

    }

    public void **Output**(){

**System**.out.**printf**("\n%20s",name );

**System**.out.**printf**("%5s",PhoneNumber );

    }

    public **String** **getName**() {

        return name;

    }

    public void **setName**(**String** name) {

        this.name = name;

    }

    public **String** **getPhoneNumber**() {

        return PhoneNumber;

    }

    public void **setPhoneNumber**(**String** phoneNumber) {

        PhoneNumber = phoneNumber;

    }

}

Tại sao cần lớp đấy

Triển khai lớp đấy như nào ( Lường hoạt động , liên kết tới lớp khác , chức năng ra sao )

## 3.2 Lớp Employee

Được kế thừa từ lớp Person , thêm thuộc tính id và address .Lớp này có các phương thức getter của thuộc tính.

package **quanlyvattuxaydung**;

import **java**.**util**.\*;

public class **Employee** extends **Person**{

    private **String** id;

    private **String** address;

*// private String store;*

   public **Employee**(**String** name , **String** PhoneNumber, **String** id,**String** address){

       super(name,PhoneNumber);

       this.id=id;

       this.address = address;

*//    this.store=store;*

   }

    @**Override**

    public void **Input**() {

*// Additional implementation for Employee input*

        super.**Input**();  *// Call the input method of the base class (Person)*

**Scanner** sc = new **Scanner**(**System**.in);

**System**.out.**println**("Nhap thong thin dang nhap:");

        super.**Input**();

**System**.out.**println**("Nhap Dia Chi:");

        address = sc.**nextLine**();

    }

    @**Override**

    public void **Output**() {

        super.**Output**();  *// Call the output method of the base class (Person)*

**System**.out.**println**("Employee ID: " + id);

**System**.out.**println**("Employee Address: " + address);

*// System.out.println("Store: " + store.getName());*

    }

    public **String** **getId**() {

        return id;

    }

    public void **setId**(**String** id) {

        this.id = id;

    }

    public **String** **getAddress**() {

        return address;

    }

    public void **setAddress**(**String** address) {

        this.address = address;

    }

}

## 3.3 Lớp NhaCungCap

Lớp này được kế thừa từ lớp Person , thêm thuộc tính address , và lấy danh sách sản phẩm . Lớp này sẽ gắn kết lớp item , để thêm sản phẩm thì lúc đó người dùng sẽ nhập cả tên nhà cung cấp và số lượng nhà cung cấp .

Nếu quản lý nhà cung cấp thì sẽ phải lấy danh sách bên quản lý sản phẩm thông qua phương thức input của bên quản lý sản phẩm , thì nó đó sẽ đỡ cần phải nhập lại . Hoặc là tôi đã nhập rồi thì thôi không cần nhập nữa.

package **quanlyvattuxaydung**;

import **java**.**util**.\*;

public class **NhaCungCap** extends **Person**{

    protected **String** address; *//*

    private **ArrayList**<**Item**> sanPhams;

    public **NhaCungCap**(**String** name, **String** PhoneNumber, **String** address, **ArrayList**<**Item**> sanPhams) {

        super(name, PhoneNumber);

        this.address = address;

        this.sanPhams = sanPhams;

    }

    public **NhaCungCap**() {

*// Initialization or default constructor*

    }

    public **NhaCungCap**(**String** name) {

        super(name);

    }

    public **String** **getAddress**() {

        return address;

    }

    public void **setAddress**(**String** address) {

        this.address = address;

    }

    public **ArrayList**<**Item**> **getSanPhams**() {

        return sanPhams;

    }

    public void **setSanPhams**(**ArrayList**<**Item**> sanPhams) {

        this.sanPhams = sanPhams;

    }

    public void **Input**(){

**Scanner** sc = new **Scanner**(**System**.in);

**System**.out.**println**("Nhap thong thin dang nhap:");

        super.**Input**();

**System**.out.**println**("Nhap Dia Chi:");

        address = sc.**nextLine**();

**System**.out.**println**("Nhap so luong san pham nha cung cap :");

        int soLuong = sc.**nextInt**();

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

**Item** sanPham = new **Item**();

            sanPham.**Input**(); *//nHAP VAo*

            sanPhams.**add**(sanPham); *//thêm vào sản phẩm*

        }

**System**.out.**println**("Đã được thêm vào");

    }

    public void **Output**(){

**System**.err.**println**("Thong tin nha cung cap :");

        super.**Output**();

**System**.err.**println**("Danh sach san pham:");

        for(**Item** sanPham :sanPhams){

            sanPham.**Output**();

        }

    }

}

## 3.4 Lớp customer

Lớp customer này có các thuộc tính chung của lớp Person , sẽ bổ sung thêm thuộc tính id , sản phẩm mua .

Phương thức : Input() , Output() ,Getter , Đếm số lượng khách hàng khi xây dựng lớp quản lý customer thì cần có thuộc tính của lớp , quản lý danh sách khách hàng đã mua , tính tổng số lượng khách hàng , tìm kiếm khách hàng . Thông tinh khách hàng , số lượng khách hàng đã mua .

package **quanlyvattuxaydung**;

import **java**.**util**.\*;

public class **Customer** extends **Person**{

    private int id;

    private **String** tenSanPham;

    public **Customer**(**String** name , **String** PhoneNumber,int id , **String** tenSanPham){

        super(name,PhoneNumber);

        this.id=id;

        this.tenSanPham = tenSanPham;

    }

    public **Customer**(){

    }

    public int **getId**() {

        return id;

    }

    public void **setId**(int id){

        this.id = id;

    }

    public **String** **getTenSanPham**() {

        return tenSanPham;

    }

    public void **setTenSanPham**(**String** tenSanPham) {

        this.tenSanPham = tenSanPham;

    }

*//nhập*

    @**Override** *// kiểm tra xem có ghi đè hay không*

    public void **Input**(){

        super.**Input**();  *// Call the input method of the base class (Person)*

**Scanner** sc = new **Scanner**(**System**.in);

**System**.out.**println**("Nhap thong thin dang nhap:");

*// super.Input();*

**System**.out.**println**("Nhap id khach hang :");

        id = sc.**nextInt**();

        sc.**nextLine**();

**System**.out.**println**("Nhap ten san pham khach hang mua :");

        tenSanPham = sc.**nextLine**();

    }

    @**Override**

    public void **Output**(){

        super.**Output**();

**System**.out.**print**(" " + id);

**System**.out.**print**("Ten san pham khach hang mua: " + tenSanPham);

    }

}

## 3.5 Lớp Item

Lớp Item có các thuộc tính lần lượt :

name , id ,price , unit , nhaCungCap ( những thông tin ở lớp dữ liệu NhaCungCap) , mieuTa , soLuong .

package **quanlyvattuxaydung**;

import **java**.**util**.\*;

public class **Item**{

    private **String** name;

    private **String** id;

    private double price;

    private **String** unit;

    private **NhaCungCap** nhaCungCap ;

    private **String** mieuTa ;

    private int soLuong;

    public **Item**(**String** name , **String** id,double price,**String** unit,**NhaCungCap** nhaCungCap,**String** mieuTa,int soLuong){

        this.name = name;

        this.id = id;

        this.price = price;

        this.unit = unit;

        this.nhaCungCap = nhaCungCap;

        this.mieuTa = mieuTa;

        this.soLuong = soLuong;

    }

*// public Item(String parts, String parts2, double d, String parts3, NhaCungCap nhaCungCap2, String parts4, int i){ // tạo danh sách rỗng*

*// }*

    public **Item**(){

    }

    public **Item**(**String** name2, **String** id2, double price2, **String** unit2, **String** string, **String** mieuTa2, int soLuong2) {

    }

    public void **Input**() {

**Scanner** scanner = new **Scanner**(**System**.in);

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

        name = scanner.**nextLine**();

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

        id = scanner.**nextLine**();

**System**.out.**print**("Nhap gia tien:");

        price = scanner.**nextDouble**();

        scanner.**nextLine**();

**System**.out.**print**("Nhap don vi: ");

        unit = scanner.**nextLine**(); *// lỗi không nhâppj*

**System**.out.**print**("Nhap nha cung cap : ");

*// nhaCungCap= scanner.nextLine();*

**String** tenNhaCungCap = scanner.**nextLine**();

**NhaCungCap** nhaCungCap= new **NhaCungCap**(tenNhaCungCap);

**System**.out.**print**("Mieu ta: ");

        mieuTa = scanner.**nextLine**();

**System**.out.**print**("Nhap so luong:");

        soLuong = scanner.**nextInt**();

**System**.out.**println**("Thong tin mat hang da duoc cap nhap");

    }

    public void **Output**() {

**System**.out.**println**("Ten san pham: " + name);

**System**.out.**println**("Ma san pham: " + id);

**System**.out.**println**("Don vi: " + unit);

**System**.out.**println**("Gia ca: " + price);

**System**.out.**println**("So luong: " + soLuong);

**System**.out.**println**("Nha cung cap: " + nhaCungCap.**getName**());

    }

    public **String** **getName**() {

        return name;

    }

    public void **setName**(**String** name) {

        this.name = name;

    }

    public **String** **getId**() {

        return id;

    }

    public void **setId**(**String** id) {

        this.id = id;

    }

    public double **getPrice**() {

        return price;

    }

    public void **setPrice**(double price) {

        this.price = price;

    }

    public **String** **getUnit**() {

        return unit;

    }

    public void **setUnit**(**String** unit) {

        this.unit = unit;

    }

    public **NhaCungCap** **getNhaCungCap**() {

        return nhaCungCap;

    }

    public void **setNhaCungCap**(**NhaCungCap** nhaCungCap) {

        this.nhaCungCap = nhaCungCap;

    }

    public **String** **getMieuTa**() {

        return mieuTa;

    }

    public void **setMieuTa**(**String** mieuTa) {

        this.mieuTa = mieuTa;

    }

    public int **getSoLuong**() {

        return soLuong;

    }

    public void **setSoLuong**(int soLuong) {

        this.soLuong = soLuong;

    }

}

## 3.6 Những lớp quản lý

### 3.6.1 Lớp quản lý Item

ListItem () đây là lớp quản lý danh sách sản phẩm vật tư xây dựng

Code:

package **quanlyvattuxaydung**;

import **java**.**io**.**BufferedReader**;

import **java**.**io**.**BufferedWriter**;

import **java**.**io**.**File**;

import **java**.**io**.**FileReader**;

import **java**.**io**.**FileWriter**;

import **java**.**io**.**IOException**;

import **java**.**util**.\*;

public class **ListItems** {

*//static*

**ArrayList**<**Item**> listItem = new **ArrayList**<**Item**>();

    public **ArrayList**<**Item**> **getListItem**() {

        return listItem;

    }

    public **ArrayList**<**Item**> **item**() {

        return listItem;

    }

    public **ListItems**(){

        listItem = new **ArrayList**<**Item**>(); *//CẤP PHÁT BỘ NHỚ*

*// return listItem;*

    }

*//Kiểm tra*

    public void **Output**(){

        if(listItem.**isEmpty**()){

**System**.err.**println**("Danh sach mat hang rong");

        }

        else{

*// duyệt*

            for(**Item** i :listItem){ *// kiểu dữ liệu Item chưa các thuộc tính Item bên trong , và lấy danh sách từ listItem*

                i.**Output**();

            }

        }

    }

    public int **getSoLuong**(){

        return listItem.**size**();

    }

*//get Item*

    public **Item** **getItem**(int soLuong){

        return listItem.**get**(soLuong);

    }

*//remove*

    public void **removeItem**(**Item** i) {

        listItem.**remove**(i);

    }

*//add*

    public void **addItem**(**Item** i){

        listItem.**add**(i);

    }

*//changeInformation*

*// 3 tham số ( Item , thuộc tính , và giá trị đối tượng)*

    public void **changeItem**(**Item** item , **String** attribute , **Object** giaTri){

        for(**Item** sanPham : listItem)   {

*// sửa đổi lấy tên*

            if(sanPham.**getName**().**equals**(item.**getName**())) *// so sánh tham chiếu , giá trị bên trong kế thừa cùng thuộc tính với nhau , so sánh qua cách bộ nhớ*

            {

                switch(attribute){

                    case"name":

                    {

                        if(giaTri instanceof **String**){ *// nếu đối tượng thuộc string*

**String** str = (**String**)giaTri;

                            item.**setName**(str);   *// sửa đổi item thành sanPham*

                        }

                        else{

**System**.out.**println**("Khong co gia tri !");

                        }

                        break;

                    }

                    case"unit":

                        if(giaTri instanceof **String**){

**String** str = (**String**) giaTri;

                            item.**setName**(str);

                        }

                }

            }

        }

    }

*// Thay đổi thông tin sản phẩm*

*// Sap xep*

    private void **QuickSort**(**ArrayList**<**Item**> arr, int left, int right) {

        if (left < right) {

            int pivotIndex = **partition**(arr, left, right);

**QuickSort**(arr, left, pivotIndex - 1);

**QuickSort**(arr, pivotIndex + 1, right);

        }

    }

    private int **partition**(**ArrayList**<**Item**> arr, int left, int right) {

        double pivot = arr.**get**(right).**getPrice**();

        int i = left - 1;

        for (int j = left; j < right; j++) {

            if (arr.**get**(j).**getPrice**() <= pivot) {

                i++;

**Collections**.**swap**(arr, i, j);

            }

        }

**Collections**.**swap**(arr, i + 1, right);

        return i + 1;

    }

    public **ArrayList**<**Item**> **sortPrice**() {

**ArrayList**<**Item**> result = new **ArrayList**<>(listItem);

**QuickSort**(result, 0, result.**size**() - 1);

        return result;

    }

*// ghi file*

*// static*

    public void **writeToFile**(**String** fileName){

*// Item item = new Item("cat","1",90,"USB","Hoa phat","Vat lieu xay dung nha ",2);*

*// listItem.add(item);*

        try(**BufferedWriter**  bw = new **BufferedWriter**(new **FileWriter**(new **File**(fileName))))

        {

            for(**Item** item2: listItem){

**String** dong = item2.**getName**()+","+item2.**getId**()+","+item2.**getPrice**()+","+item2.**getUnit**()+","+item2.**getNhaCungCap**()+","+item2.**getMieuTa**()+","+item2.**getSoLuong**();

                bw.**write**(dong);

                bw.**newLine**();

**System**.out.**println**();

            }

**System**.out.**println**("Luu du lieu thanh cong!");

        }catch(**IOException** e){

            e.**printStackTrace**();

        }

    }

*//Doc tu file*

    public void **ReadFile**(**String** fileName){

        try (**BufferedReader** br = new **BufferedReader**(new **FileReader**(fileName))) {

**String** line;

            while((line =br.**readLine**())!=null){

**String**[] parts = line.**split**(",");

*// Item item = new Item(parts[0],parts[1],Double.parseDouble(parts[2]),parts[3],new NhaCungCap(parts[4]),parts[5],Integer.parseInt(parts[6]));*

**Item** item = new **Item**(parts[0], parts[1], **Double**.**parseDouble**(parts[2]), parts[3],

                new **NhaCungCap**(parts[4]), parts[5], **Integer**.**parseInt**(parts[6]));

                listItem.**add**(item);

            }

        }catch(**IOException** | **NumberFormatException** e){

            e.**printStackTrace**();

        }

    }

*// Gena*

    public static void **main**(**String**[] args) {

*//     // writeToFile("QuanLyVatTuXayDung\\src\\Data\\ListItem.txt");*

*//     ListItems listItem = new ListItems();*

*//     // Generate and add 10 sample Item data*

*//     for (int i = 0; i < 10; i++) {*

*//         Item item = new Item(*

*//                 "Item" + i,*

*//                 String.valueOf(i),*

*//                 10.0 \* (i + 1),*

*//                 "Unit" + i,*

*//                 new NhaCungCap("NCC" + i),*

*//                 "Description" + i,*

*//                 i + 1*

*//         );*

*//         listItem.addItem(item);*

*//     }*

*//     // Write the data to the file*

*//     listItem.writeToFile("ListItem.txt");*

*// Test hàm changeItem*

**ListItems** listItem = new **ListItems**();

        listItem.**addItem**(new **Item**("Muc 1", "ID1", 10.0, "Unit1", new **NhaCungCap** ("NCC1"), "Mieu ta 1", 5));

        listItem.**addItem**(new **Item**("Muc 2", "ID2", 15.0, "Unit2", new **NhaCungCap** ("NCC2"), "Mieu ta 2", 3));

**Item** itemToChange = new **Item**("Muc 1", "ID1", 10.0, "Unit1", ("NCC3"), "Mieu ta 1", 5);

**System**.out.**println**("Danh sach truoc khi thay doi");

        listItem.**Output**();

*// Gọi hàm changeItem để thay đổi tên của itemToChange thành "Mục 1 (Thay doi)"*

        listItem.**changeItem**(itemToChange,"name","Thay doi muc 1");

*// Hiển thị danh sách sau khi thay đổi*

**System**.out.**println**("Danh sach sau khi thay doi:");

        listItem.**Output**();

*// xử lý cách lấy theo danh sách sản phẩm*

*// ArrayList<Item> dsSanPham = listItem.getListItem();*

*// for(Item i :dsSanPham){*

*//     int j = 1;*

*//     System.out.println("San pham thu "+j+"\n"+i);*

*//     j +=1;*

*// }*

*// // in ra từng thuộc tính bên trong*

*// for (Item i : dsSanPham) {*

*//     System.out.println("Ten san pham: " + i.getName());*

*//     System.out.println("Ma san pham: " + i.getId());*

*//     System.out.println("Gia : " + i.getPrice());*

*//     System.out.println("Don vi: " + i.getUnit());*

*//     System.out.println("Nha cung cap: " + i.getNhaCungCap().getName()); // Ví dụ, in tên nhà cung cấp*

*//     System.out.println("Mieu ta: " + i.getMieuTa());*

*//     System.out.println("So luong: " + i.getSoLuong());*

*//     System.out.println("------------------------");*

*// }*

*// // listItem.Output();*

    }

}

### 3.6.2 Lớp quản lý Customer

Đây là lớp quản lý những thông tin của khách hàng , tên , id , số lượng khách hàng .

package **quanlyvattuxaydung**;

import **java**.**util**.\*;

import **java**.**lang**.**reflect**.**Field**;

public class **ListCustomer** {

**ArrayList**<**Customer**> listCustomer;

    public **ListCustomer**(**ArrayList**<**Customer**> listCustomer){

        this.listCustomer = (listCustomer!=null) ? listCustomer : new **ArrayList**<>() ;

    }

    public **ListCustomer**(){

        this.listCustomer = new **ArrayList**<>();

    }

    public **ArrayList**<**Customer**> **getListCustomer**() {

        return listCustomer;

    }

    public void **setListCustomer**(**ArrayList**<**Customer**> listCustomer) {

        this.listCustomer = listCustomer;

    }

*// Thay đổi thông tin*

    public void **updateCustomerInfo**(**Customer** cus, **String** thuocTinh, **Object** giaTri) {

        if (cus == null) {

**System**.out.**println**("Khong tim thay !");

            return;

        } else {

            for (**Customer** customer : listCustomer) {

**System**.err.**println**("Neu khong muon thay doi thuoc tinh nao thi nhap 'n' :");

**String** str = (**String**) giaTri;

                switch (thuocTinh) {

                    case "name":

                        if (customer.**getId**() == cus.**getId**()) {

                            if (giaTri instanceof **String**) {

                                customer.**setName**(str);

                            } else if (str.**equals**("n")) {

                                continue;

                            } else {

**System**.out.**println**("Khong tim thay !");

                                return;

                            }

                        }

                        break;

                    case "id":

                        if (customer.**getId**() == cus.**getId**()) {

                            if (giaTri instanceof **String**) {

                                int id = (int) giaTri;

                                customer.**setId**(id);

                            } else if (str.**equals**("n")) {

                                continue;

                            } else {

**System**.out.**println**("Khong tim thay !");

                                return;

                            }

                        }

                        break;

                    case "PhoneNumber":

*// Xử lý cập nhật PhoneNumber ở đây (tương tự như name và id)*

                        break;

                    case "tenSanPham":

*// Xử lý cập nhật tenSanPham ở đây (tương tự như name và id)*

                        break;

*// Thêm các trường hợp khác cần cập nhật*

                    default:

**System**.out.**println**("Thuoc tinh khong hop le!");

                        return;

                }

            }

        }

    }

*// Lấy số lượng customer*

    public int **soLuongKhachHang**(){

*// Customer customer = new Customer();*

        return listCustomer.**size**();

    }

*// Thêm khách hàng*

    public void **addCustomer**(**Customer** newCustomer){

        if(listCustomer.**contains**(newCustomer))

        {

**System**.out.**println**("Khach hang da ton tai!");

            return;

        }

        listCustomer.**add**(newCustomer);

    }

    public void **removeCustomer**(**Customer** newCustomer){ *// sử dụng iterator lặp qua để không xảy ra lỗi ngạoi lệ*

        for(**Customer** customer :listCustomer){

             if(customer.**getId**()==newCustomer.**getId**())

                 listCustomer.**remove**(customer);

        }

        if (listCustomer.**isEmpty**()) {

**System**.out.**println**("Khong tim thay khach hang co ID: " + newCustomer.**getId**());

        } else {

**System**.out.**println**("Da xoa khach hang ID: " + newCustomer.**getId**());

        }

    }

*// format*

    public void **Output**() {

        if (listCustomer.**isEmpty**()) {

**System**.out.**println**("Danh sach rong");

        } else {

*// Print header*

**System**.out.**printf**("%-20s %-10s %-10s %-10s%n", "Name", "ID", "Phone","Ten san pham");

            for (**Customer** customer : listCustomer) {

**System**.out.**printf**("%-20s %-10d %-15s %-10s%n",

                customer.**getName**(), customer.**getId**(),

                customer.PhoneNumber, customer.**getTenSanPham**());

            }

        }

    }

*// Các update thông tin 2*

    public static void **updateProperty**(**Customer** cus, **String** propertyName, **Object** value) {

        try {

**Field** field = **Customer**.class.**getDeclaredField**(propertyName);

            field.**setAccessible**(true);

*// Check if the types match before updating*

            if (field.**getType**().**isInstance**(value)) {

                field.**set**(cus, value);

            } else {

**System**.out.**println**("Khong the nhan dien gia tri.");

            }

        } catch (**NoSuchFieldException** | **IllegalAccessException** e) {

**System**.out.**println**("Thuoc tinh khong hop le.");

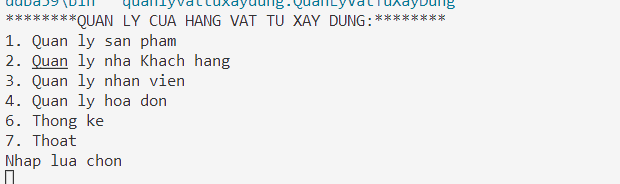
        }

    }

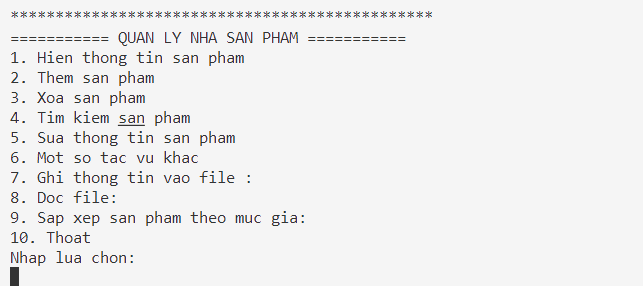
}

# Chương 4 Kiểm thử chương trình

## 4.1 Menu chương trình



### 4.1.1 Menu Quản lý sản phẩm



### 4.1.2 Quản lý thông tin khách hàng

### 