**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**



**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO ĐỒ ÁN MÔN LẬP TRÌNH HĐT**

**ĐỀ TÀI: QUẢN LÝ KHO THỰC PHẨM**

Giảng viên hướng dẫn : **Ths. Phạm Trọng Huynh**

Nhóm : **10**

Sinh viên thực hiện : **Nguyễn Đình Châu(1050080090),**

**Trịnh Gia Bảo(1050080089)**

**Phan Nguyễn Bảo Khang(1050080102)**

Lớp : **10\_ĐH\_CNTT3**

Khoá **: 10**

***TP. Hồ Chí Minh, tháng 04 năm 2023***

MỤC LỤC

[I. Phát biểu bài toán. 3](#_Toc131581031)

[II. Xác định các lớp của bài toán. 3](#_Toc131581032)

[III. Mô tả thuật toán thao tác. 4](#_Toc131581033)

[IV. Cài đặt bài toán. 6](#_Toc131581034)

[V. Kiểm thử lập bộ test. 20](#_Toc131581035)

# Phát biểu bài toán.

Nước Việt Nam, một đất nước phát triển về nông nghiệp thì việc quản lý thực phẩm là một công việc vô cung khó khan và phức tạp. Khi nó đòi hỏi phải nhận vào một lường dữ liệu vô cùng lớn tại các kho quản lý thực phẩm từ nhỏ đến lơn. Đây cũng là đề tài mà chũng em sẽ giải đáp trong những mục bên dưới đây.

Thực phẩm: mỗi loại thực phẩm khi đưa vào gồm có Mã sản phẩm(duy nhất), Tên sẩn phẩm, Số lượng, Gía, Ngày sẩn xuất, Hạn sử dụng.

# Xác định các lớp của bài toán.

1. **Class ThucPham.**

* Thuộc tính: String MaSP, TenSP, SoLuong, Gia, NSX, HSD.
* Phương thức:
* Contructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString: trả về chuỗi đại diện của ThucPham.

1. **Class DongGoi.**

* Thuộc tính: String ChatLieu.
* Phương thức:
* Contructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString: trả về chuỗi đại diện của DongGoi.

1. **Class Tuoi.**

* Thuộc tính: String LoaiRau.
* Phương thức:
* Contructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString: trả về chuỗi đại diện của Tuoi.

1. **Class NuocUong.**

* Thuộc tính: String LoaiNuoc.
* Phương thức:
* Contructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString: trả về chuỗi đại diện của NuocUong.

1. **Class DanhSachTP.**

* Thuộc tính: ArrayList<ThucPham> Menu.
* Phương thức:
* NhapSP() : thêm Thực phẩm, có thể là Thực Phẩm Đóng Gói, Thực Phẩm Tươi hoặc Nước Uống.
* XuatDSThucPham(): xuất danh sách các Thực Phẩm vừa được thêm vào Menu.
* xoaThucPham(): xóa Thực Phẩm theo mã sản phẩm.
* sapxepMaSanPhamGiamDan(): sắp xếp giảm dần theo số lượng của Thực Phẩm.
* sapxepMaSanPhamTangDan(): sắp xếp tăng dần theo số lượng của Thực Phẩm.
* TimKiemTP(): tiềm kiếm sản phẩm theo mã sản phẩm.
* Tong(): tính tổng số lượng số lượng của theo từng loại Thực Phẩm có trong kho.
* GiaCao(): xuất những sản phẩm có giá thành cao trên 15000Đ.
* SoNgaySD(): số ngày sử dụng của từng sản phẩm đã nhập vào.

# Mô tả thuật toán thao tác.

Diagram

Description automatically generated

1. **Chỉnh sửa thông tin thực phẩm.**

Thuật toán:

* Nhập: nhập vào thông tin của đối tượng mới theo từng loại thực phẩm được chọn.
* Xuất: xuất ra tất cả các thông tin của thực phẩm vừa nhập gọi toString.
* Xóa: Trong Arraylist dùng phép remove.

1. **Hiện thị tăng giảm theo số lượng thực phẩm.**

Thuật toán: giảm dần

* Sử dụng Collections Sort trong ArrayList<ThucPham>
* Với phương thức int compare( ThucPham tp1, ThucPham tp2)
* Nếu sub= tp1.getSoLuong()<tp2.getSoLuong()

return 1

* Nếu sub= tp1.getSoLuong()>tp2.getSoLuong()

return -1

* Nếu sub= tp1.getSoLuong()=tp2.getSoLuong()

return 0

Thuật toán: Tăng dần.

* Sử dụng Collections Sort trong ArrayList<ThucPham>
* Với phương thức int compare( ThucPham tp1, ThucPham tp2)
* Nếu sub= tp1.getSoLuong()>tp2.getSoLuong()

return 1

* Nếu sub= tp1.getSoLuong()<tp2.getSoLuong()

return -1

* Nếu sub= tp1.getSoLuong()=tp2.getSoLuong()

return 0

1. **Hiện thị Thực Phẩm được tìm kiếm, Input:Tên sản phẩm.**

Thuật toán:

* Trong Arraylist<ThucPham>, tìm ra ThucPham có tên giống với tên sản phẩm được nhập vào bằng phép indexof() > = 0.
* Sử dụng vòng lặp foreach, lấy tên sản phẩm của ThucPham.
* Nếu có tên giống với tên sản phẩm nhập vào sẽ xuất ra màn hình.

1. **Tính tổng số lượng sẩn phẩm của từng loại thực phẩm.**

Thuật toán:

* Trong Arraylist<ThucPham>, lọc ra ThucPham là DongGoi, Tuoi và NuocUong bằng phép instance of.
* Sử dụng vòng lặp foreach, lấy ra Soluong trong Menu.
* Tình tổng số lượng theo từng loại thực phẩm.

1. **Hiện ra các thực phẩm có giá thành trên 15000.**

Thuật toán:

* Sử dụng vòng lặp foreach, lấy ra giá của từng loại sản phẩm có trong Menu.
* Sử dụng if để so sánh giá nếu lớn hơn 15000.
* Nếu thực phẩm đó có giá lớn hơn 15000 thì xuất ra màn hình.

1. **Tính số ngày sử dụng của từng loại thực phẩm.**

Thuật toán:

* Sử dụng vòng lặp foreach, lấy ra ngày sản xuất và hạn sử dụng của từng loại sản phẩm có trong Menu.
* Lấy Hạn sủ dụng – Ngày sản xuất của sản phẩm để biết được số ngày sử dụng của sẩn phẩm là bao lâu.

# Cài đặt bài toán.

1. **Class ThucPham.**

package quanlykho;

import java.util.Calendar;

import java.util.Date;

import java.util.Objects;

public class ThucPham implements Comparable<ThucPham>{

private String MaSP;

private String TenSP;

private int Soluong;

private double gia;

private int NSX;

private int HSD;

public ThucPham(){

}

public ThucPham(String MaSP, String TenSP,int Soluong, double gia,int NSX,int HSD){

this.MaSP = MaSP;

this.TenSP =TenSP;

this.Soluong = Soluong;

this.gia = gia;

this.NSX = NSX;

this.HSD = HSD;

}

// Tạo các phương thức get/setter

public String getMaSP() {

return MaSP;

}

public void setMaSP(String MaSP) {

this.MaSP = MaSP;

}

public String getTenSP() {

return TenSP;

}

public void setTenSP(String TenSP) {

this.TenSP = TenSP;

}

public int getSoluong() {

return Soluong;

}

public void setSoluong(int Soluong) {

this.Soluong = Soluong;

}

public double getGia() {

return gia;

}

public void setGia(double gia) {

this.gia = gia;

}

public int getNSX() {

return NSX;

}

public void setNSX(int NSX) {

this.NSX = NSX;

}

public int getHSD() {

return HSD;

}

public void setHSD(int HSD) {

this.HSD = HSD;

}

@Override

public String toString() {

return "ThucPham{" + "MaSP=" + MaSP + ", TenSP=" + TenSP + ", Soluong=" + Soluong + ", gia=" + gia + ", NSX=" + NSX + ", HSD=" + HSD + '}';

}

public void ThucPham(String MaSP){

this.MaSP = MaSP;

}

public void SuDung(){

System.out.println("Huong dan bao quan: ");

}

public void SoNgaySD(){

System.out.println("San pham su dung trong.");

}

public ThucPham(String MaSP){

this.MaSP = MaSP;

}

@Override

public int compareTo(ThucPham o) {

return this.MaSP.compareTo(o.MaSP);

}

@Override

public int hashCode() {

return Objects.hash(MaSP, TenSP);

}

@Override

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null) {

return false;

}

if (getClass() != obj.getClass()) {

return false;

}

ThucPham other = (ThucPham) obj;

return Objects.equals(MaSP, other.MaSP);

}

}

1. **Class DongGoi.**

public class DongGoi extends ThucPham{

private String ChatLieu;

public DongGoi(){

}

public DongGoi(String ChatLieu) {

this.ChatLieu = ChatLieu;

}

public DongGoi(String ChatLieu, String MaSP, String TenSP, int Soluong, double gia, int NSX, int HSD) {

super(MaSP, TenSP, Soluong, gia, NSX, HSD);

this.ChatLieu = ChatLieu;

}

public String getChatLieu() {

return ChatLieu;

}

public void setChatLieu(String ChatLieu) {

this.ChatLieu = ChatLieu;

}

@Override

public String toString() {

return super.toString()+"DongGoi{" + "ChatLieu=" + ChatLieu + '}';

}

@Override

public void SoNgaySD(){

int SN = this.getHSD()- this.getNSX();

System.out.println("So ngay bao quan cua san pham dong goi nay la: " + SN);

}

@Override

public void SuDung(){

System.out.println("Bao quan duoc trong thoi gian dai.");

}

}

1. **Class Tuoi.**

public class Tuoi extends ThucPham{

private String Loai;

public Tuoi() {

}

public Tuoi(String Loai) {

this.Loai = Loai;

}

public Tuoi(String Loai, String MaSP, String TenSP, int Soluong, double gia, int NSX, int HSD) {

super(MaSP, TenSP, Soluong, gia, NSX, HSD);

this.Loai = Loai;

}

public String getLoai() {

return Loai;

}

public void setLoai(String Loai) {

this.Loai = Loai;

}

@Override

public String toString() {

return super.toString()+"Tuoi{" + "Loai=" + Loai + '}';

}

@Override

public void SoNgaySD(){

int SN = this.getHSD()- this.getNSX();

System.out.println("So ngay bao quan cua san pham tuoi nay la: " + SN);

}

@Override

public void SuDung(){

System.out.println("Bao quan trong thoi gian ngan.");

}

}

1. **Class NuocUong.**

public class NuocUong extends ThucPham {

private String LoaiNuoc;

public NuocUong(){

}

public NuocUong(String LoaiNuoc) {

this.LoaiNuoc = LoaiNuoc;

}

public NuocUong(String LoaiNuoc, String MaSP, String TenSP, int Soluong, double gia, int NSX, int HSD) {

super(MaSP, TenSP, Soluong, gia, NSX, HSD);

this.LoaiNuoc = LoaiNuoc;

}

public String getLoaiNuoc() {

return LoaiNuoc;

}

public void setLoaiNuoc(String LoaiNuoc) {

this.LoaiNuoc = LoaiNuoc;

}

@Override

public String toString() {

return super.toString()+"NuocUong{" + "LoaiNuoc=" + LoaiNuoc + '}';

}

@Override

public void SoNgaySD(){

int SN = this.getHSD()- this.getNSX();

System.out.println("So ngay bao quan cua san pham nuoc uong nay la: " + SN);

}

@Override

public void SuDung(){

System.out.println("Bao quan theo thoi han va nhiet do phu hop.");

}

}

1. **Class DanhSachTP.**

package quanlykho;

import java.util.Scanner;

import java.util.ArrayList;

import java.util.Collection;

import java.util.Collections;

import java.util.Comparator;

import quanlykho.ThucPham;

/\*\*

\*

\* @author DINH CHAU

\*/

public class DanhSachTP {

private ArrayList<ThucPham> Menu;

Scanner sc = new Scanner(System.in);

// Khai báo 1 cái arraylist

int chon =0;

public DanhSachTP(){

super();

Menu = new ArrayList<ThucPham>();

}

public void NhapSP(int n)

{

do {

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

System.out.println("0: Thoat !!!");

System.out.println("1: Thuc pham Dong Goi.");

System.out.println("2: Thuc pham Tuoi.");

System.out.println("3: Nuoc uong.");

System.out.println("Nhap lua chon cu ban: ");

chon = sc.nextInt();sc.nextLine();

switch (chon){

case 1:

{

String MaSP;

String TenSP;

int Soluong;

double gia;

String ChatLieu;

int NSX;

int HSD;

System.out.print("Nhap Ma san pham: "); MaSP = sc.nextLine();

System.out.print("Nhap ten san pham: "); TenSP = sc.nextLine();

System.out.print("Nhap so luong san pham: "); Soluong = sc.nextInt();

System.out.print("Nhap gia san pham: "); gia = sc.nextDouble();

System.out.print("Nhap chat lieu dong goi: "); ChatLieu = sc.next();

System.out.print("Nhap NSX: "); NSX = sc.nextInt();sc.nextLine();

System.out.print("Nhap HSD: "); HSD = sc.nextInt();sc.nextLine();

DongGoi dg = new DongGoi(MaSP, TenSP,ChatLieu, Soluong, gia, NSX, HSD);

Menu.add(dg);

}

break;

case 2:

{

String MaSP;

String TenSP;

int Soluong;

double gia;

String Loai;

int NSX;

int HSD;

System.out.print("Nhap Ma san pham: "); MaSP = sc.nextLine();

System.out.print("Nhap ten san pham: "); TenSP = sc.nextLine();

System.out.print("Nhap so luong san pham: "); Soluong = sc.nextInt();

System.out.print("Nhap gia san pham: "); gia = sc.nextDouble();

System.out.print("Nhap loai rau cu qua: "); Loai = sc.next();

System.out.print("Nhap NSX: "); NSX = sc.nextInt();sc.nextLine();

System.out.print("Nhap HSD: "); HSD = sc.nextInt();sc.nextLine();

Tuoi t = new Tuoi(MaSP, TenSP,Loai, Soluong, gia, NSX, HSD);

Menu.add(t);

}

break;

case 3:

{

String MaSP;

String TenSP;

int Soluong;

double gia;

String LoaiNuoc;

int NSX;

int HSD;

System.out.print("Nhap Ma san pham: "); MaSP = sc.nextLine();

System.out.print("Nhap ten san pham: "); TenSP = sc.nextLine();

System.out.print("Nhap so luong san pham: "); Soluong = sc.nextInt();

System.out.print("Nhap gia san pham: "); gia = sc.nextDouble();

System.out.print("Nhap loai nuoc uong: "); LoaiNuoc = sc.next();

System.out.print("Nhap NSX: "); NSX = sc.nextInt();sc.nextLine();

System.out.print("Nhap HSD: "); HSD = sc.nextInt();sc.nextLine();

NuocUong nu = new NuocUong(MaSP, TenSP,LoaiNuoc, Soluong, gia, NSX, HSD);

Menu.add(nu);

}

break;

}

}

} while (chon != 0);

}

public void XuatDSThucPham(){

for(ThucPham thucPham : Menu){

System.out.println(thucPham.toString());

}

}

public boolean xoaThucPham(String MaSP){

for (ThucPham tp : Menu) {

if (tp.getMaSP().equals(MaSP)) {

return this.Menu.remove(tp);

}

}

return false;

}

public void sapxepMaSanPhamGiamDan() {

Collections.sort(this.Menu, new Comparator<ThucPham>() {

@Override

public int compare(ThucPham tp1, ThucPham tp2) {

if(tp1.getSoluong()<tp2.getSoluong()) {

return 1;

}else if(tp1.getSoluong()>tp2.getSoluong()) {

return -1;

}else {

return 0;

}

}

});

}

public void sapxepMaSanPhamTangDan() {

Collections.sort(this.Menu, new Comparator<ThucPham>() {

@Override

public int compare(ThucPham tp1, ThucPham tp2) {

if(tp1.getSoluong()>tp2.getSoluong()) {

return 1;

}else if(tp1.getSoluong()<tp2.getSoluong()) {

return -1;

}else {

return 0;

}

}

});

}

public void TimKiemTP(String ten){

for(ThucPham tp : Menu){

if(tp.getTenSP().indexOf(ten) >= 0){

System.out.println(tp);

}

}

}

public void Tong(){

int DG = 0;

int T = 0;

int NU = 0;

for (ThucPham tp : Menu) {

if (tp instanceof DongGoi) {

DG += tp.getSoluong();

} else if (tp instanceof Tuoi) {

T += tp.getSoluong();

}else if (tp instanceof NuocUong) {

NU += tp.getSoluong();

}

}

System.out.println("Tong so luong thuc pham dong goi trong kho la: " + DG);

System.out.println("Tong so luong thuc pham tuoi trong kho la: : " + T);

System.out.println("Tong so luong nuoc uong trong kho la: : " + NU);

}

public void GiaCao(){

for(ThucPham tp : Menu){

if(tp.getGia() > 15000){

System.out.println(tp.toString());

}

}

}

public void SoNgaySD(){

for(ThucPham tp : Menu){

tp.SoNgaySD();

}

}

}

1. **Class QuanLyKho(class Main).**

package quanlykho;

import java.util.Scanner;

import java.util.Date;

import quanlykho.ThucPham;

import quanlykho.DanhSachTP;

public class QuanLyKho {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

DanhSachTP dstp = new DanhSachTP();

int chon = 0;

ThucPham t = new ThucPham();

t.SuDung();

t = new Tuoi();

t.SuDung();

t = new DongGoi();

t.SuDung();

t = new NuocUong();

t.SuDung();

do {

System.out.println("-----------DANH SACH THUC PHAM----------");

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

System.out.println("0. Thoat khoi man hinh nhap.\n"

+ "1. Them Thuc Pham vao Menu.\n"

+ "2. Xuat menu thuc pham.\n"

+ "3. Xoa mot san pham trong menu. \n"

+ "4. Sap xep san pham giam dan theo so luong.\n"

+ "5. Sap xep san pham tang dan theo so luong.\n"

+ "6. Tim kiem san pham trong menu theo ten. \n"

+ "7. Tong so luong cua san pham trong kho. \n"

+ "8. San pham co gia thanh lon hon 15000. \n"

+ "9. So ngay su dung cua san pham. \n");

System.out.print("Lua chon cua ban la: ");

chon = sc.nextInt();

sc.nextLine();

if(chon == 0){}

else if(chon ==1){

dstp.NhapSP(chon);

}

else if(chon ==2){

dstp.XuatDSThucPham();

}

else if(chon ==3){

System.out.print("Nhap vao ma thuc pham ma ban muon xoa: ");

String MaSP = sc.nextLine();//sc.nextLine();

ThucPham tp = new ThucPham(MaSP);

System.out.println("Xoa thuc pham: " + dstp.xoaThucPham(MaSP));

}

else if(chon ==4){

dstp.sapxepMaSanPhamGiamDan();

dstp.XuatDSThucPham();

}

else if(chon ==5){

dstp.sapxepMaSanPhamTangDan();

dstp.XuatDSThucPham();

}

else if(chon ==6){

System.out.print("Nhap ten thuc pham muon tim: "); String TenSP = sc.next();

System.out.println("Thuc pham tim duoc la: ");

dstp.TimKiemTP(TenSP);

}

else if(chon == 7){

dstp.Tong();

}

else if(chon == 8){

dstp.GiaCao();

}

else if(chon == 9){

dstp.SoNgaySD();

}

} while (chon !=0);

}

}

# Kiểm thử lập bộ test.

Dữ liệu DongGoi:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã Sản Phẩm | Tên Sản Phẩm | Chất Liệu | Số Lượng | Gía | NSX | HSD |
| DG01 | Mi Goi | Nhua | 5 | 5000 | 22-3-2021 | 29-6-2023 |
| DG02 | Thit Hop | Kim Loai | 10 | 20000 | 21-5-2022 | 22-7-2023 |
| DG03 | Ca Hop | Kim Loai | 12 | 22000 | 6-5-2021 | 26-6-2023 |
| DG04 | Banh | Giay | 29 | 35000 | 6-9-2022 | 24-5-2023 |

Dữ liệu Tuoi:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã Sản Phẩm | Tên Sản Phẩm | Loại  Rau, củ, quả | Số Lượng | Gía | NSX | HSD |
| TU01 | Rau Cai | Rau | 12 | 5000 | 25-5-2021 | 9-6-2021 |
| TU02 | Ca Rot | Củ | 100 | 12000 | 24-1-2023 | 2-2-2023 |
| TU03 | Qua Cam | Qủa | 150 | 21000 | 21-6-2021 | 2-9-2023 |
| TU04 | Ca Chua | Qủa | 230 | 17000 | 2-2-2023 | 5-3-2023 |

Dữ liệu NuocUong:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã Sản Phẩm | Tên Sản Phẩm | Loại  Nước Uống | Số Lượng | Gía | NSX | HSD |
| NU01 | Pepsi | Nước Có Ga | 150 | 18000 | 21-6-2021 | 2-3-2024 |
| NU02 | Aquafina | Nước Lọc | 600 | 12000 | 21-3-2023 | 2-6-2025 |
| NU03 | Nuoc Cam | Nước Không Ga | 240 | 19000 | 21-6-2023 | 2-9-2025 |

1. Text

   Description automatically generatedText

   Description automatically generated with low confidenceText

   Description automatically generatedNhập thông tin thực phẩm.
2. Hiện thông tin thực phẩm.

Text

Description automatically generated

1. Xóa thực phẩm khỏi danh sách.

Text, letter

Description automatically generated

1. Sắp xếp thực phẩm giảm dần theo số lượng.

Text

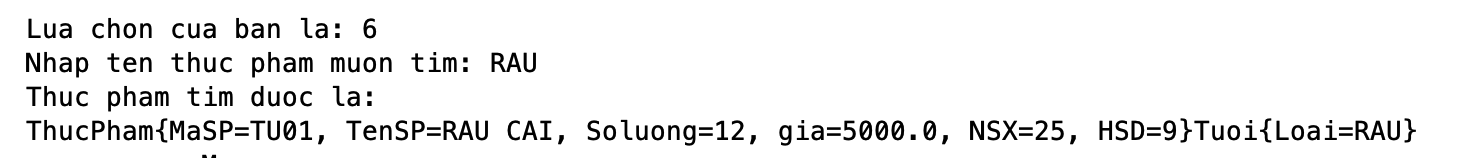
Description automatically generated

1. Sắp xếp thực phẩm tăng dần theo số lượng.

Text

Description automatically generated

1. Tìm kiếm thưc phẩm trong danh sách.

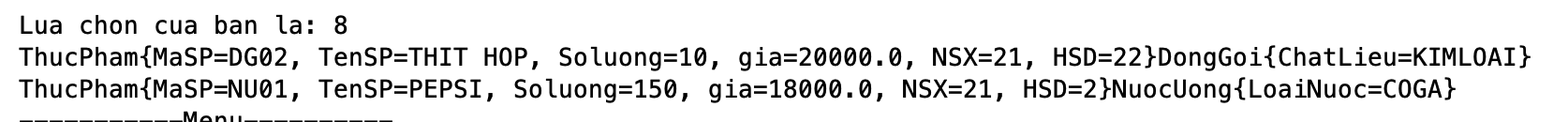


1. Tổng số lượng sản phẩm có trong kho.

Text

Description automatically generated

1. Hiện thực phẩm có giá thành cao hơn 15000đ.



1. Số ngày sử dụng của từng sản phẩm.

Text

Description automatically generated