

## Đề bài: Chương trình kiểm tra thực phẩm.

---

Viết lớp **ThucPham** mô tả một hàng hóa là hàng thực phẩm trong kho của một siêu thị, có các thuộc tính:

- Mã hàng (không cho phép sửa, không được để rỗng).
- Tên hàng (không được để rỗng), đơn giá (>0).
- Ngày sản xuất và ngày hết hạn (ngày không được để rỗng, ngày hết hạn phải sau ngày sản xuất).

Ràng buộc chặt chẽ các ràng buộc trên các trường dữ liệu. Nếu dữ liệu không hợp lệ thì gán giá trị mặc định cho phép tương ứng của trường đó.

- Tạo 1 constructor có đầy đủ tham số, 1 constructor có tham số là mã hàng.
- Viết các phương thức setters/getters.
- Viết phương thức kiểm tra một hàng thực phẩm đã hết hạn chưa?
- Phương thức toString, trả về chuỗi chứa thông tin của hàng thực phẩm. Trong đó: Định dạng đơn giá có phân cách hàng nghìn. Định dạng kiểu ngày là dd/MM/yyyy.

Viết hàm Main để thực hiện các yêu cầu trên.

### // Thực phẩm

```
import java.text.NumberFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;

public class ThucPham {
    //Khai báo các thuộc tính cần thiết cho bài toán
    private int maHang;
    private String tenHang;
    private double donGia;
    private Date nSX, hSD;

    //Khởi tạo constructor có tham số
    public ThucPham(int maHang, String tenHang, double donGia, Date nSX, Date
hSD) {
        this.maHang = maHang;
        this.tenHang = tenHang;
```

```

        this.donGia = donGia;
        this.nSX = nSX;
        this.hSD = hSD;
    }

    //Khởi tạo constructor mặc định
    public ThucPham() {
    }

    //-----begin getter and setter-----
    public int getMaHang() {
        return maHang;
    }

    public void setMaHang(int maHang) {
        this.maHang = maHang;
    }

    public String getTenHang() {
        return tenHang;
    }

    public void setTenHang(String tenHang) {
        this.tenHang = tenHang;
    }

    public double getDonGia() {
        return donGia;
    }

    public void setDonGia(double donGia) {
        this.donGia = donGia;
    }

    public Date getnSX() {
        return nSX;
    }

    public void setnSX(Date nSX) {
        this.nSX = nSX;
    }

    public Date gethSD() {
        return hSD;
    }

    public void sethSD(Date hSD) {
        this.hSD = hSD;
    }

    //-----end getter and setter-----
    @Override
    //khởi tạo phương thức toString
    public String toString() {
        //sử dụng phương thức Locale để biến đổi theo tiền tệ việt nam
        Locale localeVN = new Locale("vi", "VN");

```

```

        NumberFormat numberFormat =
NumberFormat.getCurrencyInstance(localeVN);
        String str = numberFormat.format(donGia);
        // sử dụng phương thức SimpleDateFormat để biến đổi ngày tháng năm
theo dạng "dd/MM/yyyy"
        SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("dd/MM/yyyy");
        String str1 = simpleDateFormat.format(nSX);
        String str2 = simpleDateFormat.format(hSD);
        return "Thông tin về thực phẩm: \n" +
                "Mã hàng : " + maHang +
                "\nTên hàng : '" + tenHang +
                "\nĐơn giá : " + str +
                "\nNgày sản xuất : " + str1 +
                "\nHạn sử dụng : " + str2
                ;
    }

    //Khởi tạo phương thức để nhập năm tháng ngày sản xuất
    public void setNSX(int year, int month, int day) {
        Calendar calendar = Calendar.getInstance();
        calendar.set(year, month - 1, day);
        this.nSX = calendar.getTime();
    }

    //Khởi tạo phương thức để nhập hạn sử dụng
    public void setHSD(int year, int month, int day) {
        Calendar calendar = Calendar.getInstance();
        calendar.set(year, month - 1, day);
        this.hSD = calendar.getTime();
    }

    //khởi tạo phương thức kiểm tra tên hàng không được để trống
    public boolean kiemTraTenHang(boolean k) {
        if (this.tenHang == "" || this.tenHang.isEmpty()) {
            System.out.println("Tên hàng không được để trống : ");
        } else {
            k = false;
        }
        return k;
    }

    //khởi tạo hàm kiểm tra ngày hết hạn không được nhỏ hơn ngày sản xuất
    public boolean kiemTraNgay(boolean t) {
        if (this.getnSX().compareTo(this.gethSD()) < 0) {
            t = false;
        } else {
            System.out.println("Ngày hết hạn không được nhỏ hơn ngày sản xuất
: ");
        }
        return t;
    }

    //khởi tạo phương thức kiểm tra hạn sử dụng của sản phẩm đã hết hạn hay
còn hạn
    public void kiemTraHSD() {

```

```

        Date today = new Date();
        today.getTime();
        SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("dd/MM/yyyy");
        String st = simpleDateFormat.format(today);
        if (this.gethSD().compareTo(today) < 0) {
            System.out.println("Hôm nay là ngày " + st + ", hàng hóa đã hết
hạn ");
        } else {
            System.out.println("Hôm nay là ngày " + st + ", hàng hóa vẫn còn
hạn ");
        }
    }
}

```

## // Main

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ThucPham tp = new ThucPham();
        boolean kt = true;
        boolean th = true;
        System.out.println("Nhập mã hàng : ");
        tp.setMaHang(scanner.nextInt());
        scanner.nextLine();
        do {
            System.out.println("Nhập tên hàng : ");
            tp.setTenHang(scanner.nextLine());
        } while (tp.kiemTraTenHang(kt));
        System.out.println("Nhập đơn giá : ");
        tp.setDonGia(scanner.nextDouble());
        do {
            System.out.println("Nhập năm, tháng, ngày sản xuất : ");
            tp.setNSX(scanner.nextInt(), scanner.nextInt(),
scanner.nextInt());
            System.out.println("nhập năm, tháng, ngày hết hạn : ");
            tp.setHSD(scanner.nextInt(), scanner.nextInt(),
scanner.nextInt());
        } while (tp.kiemTraNgay(th));
        System.out.println(tp);
        tp.kiemTraHSD();
    }
}

```