

Gestión de Productos Agroalimentarios

Alumno: Francisco López Velázquez

Materia: Programación Orientada a Objetos 2

Profesor: Dr. Carlos Hugo García C.

Fecha: 30 de noviembre de 2024

Código Fuente en GitHub:

https://github.com/f-lopez-velazquez/P00/tree/f5c5bb3cedf14b8b91c380fb0e8041c7c5b6492d/Gestion_Productos_Agroalimentarios

Resumen:

Este reporte describe el desarrollo de un programa en Java para gestionar productos de una empresa agroalimentaria. Se implementan conceptos de herencia, abstracción e interfaces, aplicados a tres tipos de productos: frescos, refrigerados y congelados.

Introducción

El objetivo de este proyecto es crear un programa en lenguaje Java que gestione tres tipos de productos: **frescos**, **refrigerados** y **congelados**. Los productos tienen características comunes, como *fecha de caducidad* y *número de lote*, y propiedades específicas según el tipo. Se utiliza:

- **Herencia:** Para compartir atributos y métodos comunes entre las clases.
- **Abstracción:** Mediante una clase base abstracta (**Producto**) y una interfaz (**InformacionProducto**).
- **Polimorfismo:** Para implementar el método `mostrarInformacion()` en cada subclase.

Estructura del Programa

El programa está dividido en las siguientes clases e interfaz:

- **Producto (abstracta):** Clase base con atributos comunes.
- **InformacionProducto (interfaz):** Define el método `mostrarInformacion()`.
- **ProductoFresco:** Extiende **Producto** e implementa **InformacionProducto**.
- **ProductoRefrigerado:** Extiende **Producto** e implementa **InformacionProducto**.
- **ProductoCongelado:** Extiende **Producto** e implementa **InformacionProducto**.
- **Test:** Clase principal que instancia objetos y muestra sus datos.

Código Fuente

A continuación, se presenta el código fuente del programa:

Clase Producto

```
1 public abstract class Producto {
2     private String fechaCaducidad;
3     private int numeroLote;
4
5     public Producto(String fechaCaducidad, int numeroLote) {
6         this.fechaCaducidad = fechaCaducidad;
7         this.numeroLote = numeroLote;
8     }
9
10    public String getFechaCaducidad() {
11        return fechaCaducidad;
12    }
13 }
```

```

14     public void setFechaCaducidad(String fechaCaducidad) {
15         this.fechaCaducidad = fechaCaducidad;
16     }
17
18     public int getNumeroLote() {
19         return numeroLote;
20     }
21
22     public void setNumeroLote(int numeroLote) {
23         this.numeroLote = numeroLote;
24     }
25
26     public abstract void mostrarInformacion();
27 }

```

Interfaz InformacionProducto

```

1 public interface InformacionProducto {
2     void mostrarInformacion();
3 }

```

Clase ProductoFresco

```

1 public class ProductoFresco extends Producto implements
    InformacionProducto {
2     private String fechaEnvasado;
3     private String paisOrigen;
4
5     public ProductoFresco(String fechaCaducidad, int numeroLote,
6                             String fechaEnvasado, String paisOrigen) {
7         super(fechaCaducidad, numeroLote);
8         this.fechaEnvasado = fechaEnvasado;
9         this.paisOrigen = paisOrigen;
10    }
11
12    @Override
13    public void mostrarInformacion() {
14        System.out.println("Producto Fresco:");
15        System.out.println("Fecha de Caducidad: " + getFechaCaducidad());
16        System.out.println("Numero de Lote: " + getNumeroLote());
17        System.out.println("Fecha de Envasado: " + fechaEnvasado);
18        System.out.println("País de Origen: " + paisOrigen);
19    }
20 }

```

Clase ProductoRefrigerado

```

1 public class ProductoRefrigerado extends Producto implements
  InformacionProducto {
2     private String codigoOrganismo;
3
4     public ProductoRefrigerado(String fechaCaducidad, int numeroLote,
5                               String codigoOrganismo) {
6         super(fechaCaducidad, numeroLote);
7         this.codigoOrganismo = codigoOrganismo;
8     }
9
10    @Override
11    public void mostrarInformacion() {
12        System.out.println("Producto Refrigerado:");
13        System.out.println("Fecha de Caducidad: " + getFechaCaducidad());
14        System.out.println("Numero de Lote: " + getNumeroLote());
15        System.out.println("Codigo del Organismo de Supervision: " +
16                            codigoOrganismo);
17    }
18 }

```

Clase ProductoCongelado

```

1 public class ProductoCongelado extends Producto implements
  InformacionProducto {
2     private double temperaturaCongelacion;
3
4     public ProductoCongelado(String fechaCaducidad, int numeroLote,
5                               double temperaturaCongelacion) {
6         super(fechaCaducidad, numeroLote);
7         this.temperaturaCongelacion = temperaturaCongelacion;
8     }
9
10    @Override
11    public void mostrarInformacion() {
12        System.out.println("Producto Congelado:");
13        System.out.println("Fecha de Caducidad: " + getFechaCaducidad());
14        System.out.println("Numero de Lote: " + getNumeroLote());
15        System.out.println("Temperatura de Congelacion Recomendada: " +
16                            temperaturaCongelacion + " C");
17    }
18 }

```

Clase Test

```

1 public class Test {
2     public static void main(String[] args) {
3         ProductoFresco fresco = new ProductoFresco("2024-12-15", 101, "
4             2024-11-01", "Mexico");
5         ProductoRefrigerado refrigerado = new ProductoRefrigerado("
6             2024-12-30", 102, "ORGAN-12345");
7     }
8 }

```

```

5         ProductoCongelado congelado = new ProductoCongelado("2025-01-20",
6             103, -18.0);
7
8         fresco.mostrarInformacion();
9         System.out.println();
10        refrigerado.mostrarInformacion();
11        System.out.println();
12        congelado.mostrarInformacion();
13    }

```

Funcionamiento del Programa

Al ejecutar el programa, la clase `Test` instancia objetos de los tres tipos de productos y llama al método `mostrarInformacion()` para imprimir los datos. La salida esperada es:

Producto Fresco:

Fecha de Caducidad: 2024-12-15

Número de Lote: 101

Fecha de Envasado: 2024-11-01

País de Origen: México

Producto Refrigerado:

Fecha de Caducidad: 2024-12-30

Número de Lote: 102

Código del Organismo de Supervisión: ORGAN-12345

Producto Congelado:

Fecha de Caducidad: 2025-01-20

Número de Lote: 103

Temperatura de Congelación Recomendada: -18.0°C

Conclusión

Este proyecto permitió aplicar conceptos fundamentales de la Programación Orientada a Objetos, como herencia, abstracción e interfaces, para desarrollar un programa funcional y estructurado.