PROGRAMACIÓN II

TP 8: Interfaces y Excepciones en Java

Repositorio Github: https://github.com/AguP10/UTN-TUPaD-P2

Alumno: Palacios Fernando Agustin.

Parte 1: Interfaces en un sistema de E-commerce

1. Crear una interfaz Pagable con el método calcularTotal()

2. Clase Producto: tiene nombre y precio, implementa Pagable.

3. Clase Pedido: tiene una lista de productos, implementa Pagable y calcula el total del pedido.

4. Ampliar con interfaces Pago y PagoConDescuento para distintos medios de pago (TarjetaCredito, PayPal), con métodos procesarPago(double) y aplicarDescuento(double).

5. Crear una interfaz Notificable para notificar cambios de estado. La clase Cliente implementa dicha interfaz y Pedido debe notificarlo al cambiar de estado.

Interfaz Pagable

```
package tp8;

public interface Pagable {
    double calcularTotal();
}
```

Clase Producto

```
package tp8;

public class Producto implements Pagable {
    private String nombre;
    private double precio;

    public Producto(String nombre, double precio) {
        this.nombre = nombre;
```

```
        this.precio = precio;
    }

    @Override
    public double calcularTotal() {
        return precio;
    }

    public String getNombre() {
        return nombre;
    }

    public double getPrecio() {
        return precio;
    }
}
```

Interfaz Pago

```
package tp8;

public interface Pago {
    void procesarPago(double monto);
}
```

Interfaz PagoConDescuento

```
package tp8;

public interface PagoConDescuento extends Pago {
    double aplicarDescuento(double monto);
}
```

Clase TarjetaCredito

```
package tp8;

public class TarjetaCredito implements PagoConDescuento {
    private String titular;

    public TarjetaCredito(String titular) {
        this.titular = titular;
    }

    @Override
    public void procesarPago(double monto) {
        System.out.println("Procesando pago con tarjeta de crédito de " + titular + ": $" + monto);
    }

    @Override
    public double aplicarDescuento(double monto) {
        double descuento = monto * 0.15;
        return monto - descuento;
    }
}
```

Clase PayPal

```
package tp8;

public class PayPal implements Pago {
    private String usuario;

    public PayPal(String usuario) {
        this.usuario = usuario;
    }

    @Override
```

```
    public void procesarPago(double monto) {
        System.out.println("Procesando pago con PayPal (" + usuario + "): $" + monto);
    }
}
```

Interface Notificable

```
package tp8;

public interface Notificable {
    void notificarCambio(String mensaje);
}
```

Clase Cliente

```
package tp8;

public class Cliente implements Notificable {
    private String nombre;
    private String email;

    public Cliente(String nombre, String email) {
        this.nombre = nombre;
        this.email = email;
    }

    @Override
    public void notificarCambio(String mensaje) {
        System.out.println("Notificación para " + nombre + ": " + mensaje);
    }

    public String getNombre() {
        return nombre;
    }
```

```
    public String getEmail() {
        return email;
    }
}
```

Clase Pedido

```
package tp8;

import java.util.ArrayList;
import java.util.List;

public class Pedido implements Pagable {
    private List<Producto> productos;
    private String estado;
    private Cliente cliente;

    public Pedido(Cliente cliente) {
        this.productos = new ArrayList<>();
        this.estado = "Pendiente";
        this.cliente = cliente;
    }

    public void agregarProducto(Producto p) {
        productos.add(p);
    }

    @Override
    public double calcularTotal() {
        double total = 0;
        for (Producto p : productos) {
            total += p.calcularTotal();
```

```
        }
        return total;
    }

    public void cambiarEstado(String nuevoEstado) {
        this.estado = nuevoEstado;
        cliente.notificarCambio("El pedido cambió a estado: " + nuevoEstado);
    }

    public String getEstado() {
        return estado;
    }
}
```

Main

```
package tp8;

public class Main {
    public static void main(String[] args) {
        Cliente cliente = new Cliente("Pepe", "pepe123@gmail.com");
        Pedido pedido = new Pedido(cliente);

        pedido.agregarProducto(new Producto("Mouse", 5000));
        pedido.agregarProducto(new Producto("Teclado", 12000));

        System.out.println("Total pedido: $" + pedido.calcularTotal());

        TarjetaCredito tarjeta = new TarjetaCredito("Pepe");
        double totalConDescuento = tarjeta.aplicarDescuento(pedido.calcularTotal());
        tarjeta.procesarPago(totalConDescuento);
       pedido.cambiarEstado("Enviado");
    }
}
```

Parte 2: Ejercicios sobre Excepciones

1. División segura

○ **Solicitar dos números y dividirlos. Manejar ArithmeticException si el divisor es cero**

```
package tp8;

import java.util.Scanner;

public class Tp8 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Ingrese el dividendo: ");
            int a = sc.nextInt();
            System.out.print("Ingrese el divisor: ");
            int b = sc.nextInt();

            int resultado = a / b;
            System.out.println("Resultado: " + resultado);
        } catch (ArithmeticException e) {
            System.out.println("Error: no se puede dividir por cero.");
        }
    }

}
```

2. Conversión de cadena a número

○ **Leer texto del usuario e intentar convertirlo a int. Manejar NumberFormatException si no es válido.**

```
package tp8;

import java.util.Scanner;
```

```
public class Tp8 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese un número en texto: ");
        String texto = sc.nextLine();

        try {
            int numero = Integer.parseInt(texto);
            System.out.println("Número convertido: " + numero);
        } catch (NumberFormatException e) {
            System.out.println("Error: el texto ingresado no es un número válido.");
        }
    }

}
```

3. Lectura de archivo

○ **Leer un archivo de texto y mostrarlo. Manejar FileNotFoundException si el archivo no existe.**

```
package tp8;

import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;

public class Tp8 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el nombre del archivo: ");
        String nombreArchivo = sc.nextLine();
```

```
        try {
            File archivo = new File(nombreArchivo);
            try (Scanner lector = new Scanner(archivo)) {
                System.out.println("Contenido del archivo:");
                while (lector.hasNextLine()) {
                    System.out.println(lector.nextLine());
                }
            }
        } catch (FileNotFoundException e) {
            System.out.println("Error: el archivo no existe ");
        }
    }

}
```

4. Excepción personalizada

○ Crear EdadInvalidaException. Lanzarla si la edad es menor a 0 o mayor a 120. Capturarla y mostrar mensaje.

Excepción

```
package tp8;

public class EdadInvalidaException extends Exception {
    public EdadInvalidaException(String mensaje) {
        super(mensaje);
    }
}
```

Main

```
package tp8;

import java.util.Scanner;
```

```
public class Tp8 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese su edad: ");
        int edad = sc.nextInt();

        try {
            validarEdad(edad);
            System.out.println("Edad válida: " + edad);
        } catch (EdadInvalidaException e) {
            System.out.println("Error: " + e.getMessage());
        }

    }

    private static void validarEdad(int edad) throws EdadInvalidaException {
        if (edad < 0 || edad > 120) {
            throw new EdadInvalidaException("La edad debe estar entre 0 y 120 años.");
        }
    }
}
```

5. Uso de try-with-resources

○ **Leer un archivo con BufferedReader usando try-with-resources. Manejar IOException correctamente.**

```
package tp8;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
```

```
public class Tp8 {

    public static void main(String[] args) {
        String rutaArchivo = "texto.txt";

        try (BufferedReader lector = new BufferedReader(new FileReader(rutaArchivo))) {
            String linea;
            System.out.println("Contenido del archivo:");

            while ((linea = lector.readLine()) != null) {
                System.out.println(linea);
            }

        } catch (IOException e) {
            System.out.println("Ocurrió un error al leer el archivo: " + e.getMessage());
        }

    }
}
```