

Trabajo Practico 8: Interfaces y Excepciones en JavaParte 1: Interfaces en un sistema de E-commerce

Main:

```
package ParteUnoInterfaces;

public class Main {

    public static void main(String[] args) {
        Cliente c1 = new Cliente("Cliente 1");
        Cliente c2 = new Cliente("Cliente 2");

        Pedido pedido = new Pedido("Pendiente", c1);
        Pedido pedido2 = new Pedido("Pendiente", c2);

        Producto p1 = new Producto("Computadora", 4500000);
        Producto p2 = new Producto("Notebook", 6000000);
        Producto p3 = new Producto("Silla Escritorio", 220000);

        pedido.agregarProducto(p1);
        pedido.agregarProducto(p3);

        pedido.notificarEstado("PROCESANDO");

        double total = pedido.calcularTotal();

        PagoConPayPal pago = new PagoConPayPal();

        pago.procesarPago(total);

        pedido.notificarEstado("PAGO REALIZADO");

        pedido2.agregarProducto(p2);
        pedido2.notificarEstado("PROCESANDO");

        double total2 = pedido2.calcularTotal();

        PagoConTarjeta pago2 = new PagoConTarjeta();

        pago2.procesarPago(total2);

        pedido2.notificarEstado("PAGO REALIZADO");
    }
}
```

Cliente:

```
package ParteUnoInterfaces;

public class Cliente implements Notificable{

    private String nombre;

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

    @Override
    public void notificarCamboDeEstado(String nuevoEstado) {
        System.out.println(nombre + " estado de su pedido: " + nuevoEstado);
    }

}
```

Pago con PayPal:

```
package ParteUnoInterfaces;

public class PagoConPayPal implements PagoConDescuento{

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

    @Override
    public void procesarPago(double monto) {
        double total = aplicarDescuento(monto);
        System.out.println("Total con descuento: " + total);
        System.out.println("Pago realizado");
    }

}
```

Pago con Descuento:

```
package ParteUnoInterfaces;

public class PagoConTarjeta implements Pago{

    @Override
    public void procesarPago(double monto) {
        System.out.println("Total: " + monto);
        System.out.println("Pago realizado");
    }

}
```

Pedido:

```
package ParteUnoInterfaces;

import java.util.ArrayList;

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

    public Pedido(String estado, Cliente cliente) {
        this.estado = estado;
        this.cliente = cliente;
        this.productos = new ArrayList();
    }

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

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

        return total;
    }

    public void notificarEstado(String nuevoestado) {
        this.estado = nuevoestado;
        cliente.notificarCamboDeEstado(nuevoestado);
    }
}
```

Notificable:

```
package ParteUnoInterfaces;

public interface Notificable {
    public void notificarCamboDeEstado(String nuevoEstado);
}
```

Pagable:

```
package ParteUnoInterfaces;

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

Pago:

```
package ParteUnoInterfaces;

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

Producto:

```
package ParteUnoInterfaces;

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

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

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

Pago con tarjeta:

```
package ParteUnoInterfaces;

public class PagoConTarjeta implements Pago{

    @Override
    public void procesarPago(double monto) {
        System.out.println("Total: " + monto);
        System.out.println("Pago realizado");
    }

}
```

Build:

```
--- exec:3.1.0:exec (default-cli) @ TP8 ---
Cliente 1 estado de su pedido: PROCESANDO
Total con descuento: 4012000.0
Pago realizado
Cliente 1 estado de su pedido: PAGO REALIZADO
Cliente 2 estado de su pedido: PROCESANDO
Total: 6000000.0
Pago realizado
Cliente 2 estado de su pedido: PAGO REALIZADO
```

```
-----
BUILD SUCCESS
-----
```

Parte 2: Ejercicios sobre Excepciones

Conversión Numero a Cadena:

```
package ParteDosExcepciones;

import java.util.Scanner;

public class ConversionNumeroACadena {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Ingrese un texto que pueda ser convertido a entero");
        String textoAConvertir = scan.nextLine();

        try {
            int textoConvertido = Integer.parseInt(textoAConvertir);
            System.out.println("El numero convertido es: " + textoConvertido);
        } catch (NumberFormatException ex) {
            System.out.println("El texto ingresado no es valido.");
        }
    }

}
```

División:

```
package ParteDosExcepciones;

import java.util.Scanner;

public class Division {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        double num1, num2;

        System.out.println("Ingrese dos numeros");
        System.out.println("Numero 1: ");
        num1 = scan.nextDouble();

        System.out.println("Numero 2:");
        num2 = scan.nextDouble();

        try {
            if (num2 == 0) {
                throw new ArithmeticException("No se puede dividir por 0");
            }
            double resultado = num1 / num2;
            System.out.println("Resultado de la division: " + resultado);
        } catch (ArithmeticException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Edad Invalida:

```
package ParteDosExcepciones;

public class EdadInvalida extends RuntimeException{

    public EdadInvalida() {
    }

    public EdadInvalida(String message) {
        super(message);
    }

    public EdadInvalida(String message, Throwable cause) {
        super(message, cause);
    }

    public EdadInvalida(Throwable cause) {
        super(cause);
    }
}
```

Excepción Edad 2

```
package ParteDosExcepciones;

import java.util.Scanner;

public class ExcepcionEdad2 {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.println("Ingrese una edad");
        int edad = Integer.parseInt(scan.nextLine());
        if (edad <= 0 || edad >= 100) {
            throw new EdadInvalida("Edad invalida");
        }
    }
}
```

Lectura del archivo txt:

```
package ParteDosExcepciones;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class LecturaArchivo {

    public static void main(String[] args) throws IOException {
        Scanner scan = new Scanner(System.in);

        System.out.println("Ingrese el nombre del archivo txt: ");
        String nombre = scan.nextLine();

        try {
            File archivo = new File(nombre);
            BufferedReader br = new BufferedReader(new FileReader(archivo));
            System.out.println(br.readLine());
        } catch (FileNotFoundException e) {
            System.out.println("El archivo no se encontro.");
        } catch (IOException ex) {
            System.out.println("Error IO");
            System.out.println(ex.getMessage());
        }
    }
}
```

Ubicación Archivo txt:

```
package ParteDosExcepciones;

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

public class UbicacionArchivoTXT {

    public static void main(String[] args) {
        File archivo = new File // Ubicacion del archivo esto varia segun el lugar donde se ejecute;
        try(BufferedReader br = new BufferedReader(new FileReader(archivo))){
            System.out.println(br.readLine());
        } catch(IOException ex) {
            System.out.println("Error de E/S: " + ex.getMessage());
        }
    }
}
```