PROGRAMACIÓN II

TP 8: Interfaces y Excepciones en Java

Alumno: ANGELELLI, Rodrigo Martin
Comision N°7
Link al repositorio: https://github.com/roangelelli/Programacion2_TUPAD_Angelelli_Rodrigo

OBJETIVO GENERAL: Desarrollar habilidades en el uso de interfaces y manejo de excepciones en Java para fomentar la modularidad, flexibilidad y robustez del código. Comprender la definición e implementación de interfaces como contratos de comportamiento y su aplicación en el diseño orientado a objetos. Aplicar jerarquías de excepciones para controlar y comunicar errores de forma segura. Diferenciar entre excepciones comprobadas y no comprobadas, y utilizar bloques try, catch, finally y throw para garantizar la integridad del programa. Integrar interfaces y manejo de excepciones en el desarrollo de aplicaciones escalables y mantenibles.

Parte 1: Interfaces en un sistema de E-commerce

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

```java
package tp8.e.commerce;
public interface Pagable {
    double calcularTotal();
}
```

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

```java
package tp8.e.commerce;
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;
    }

    @Override
    public String toString() {
        return nombre + " - $" + precio;
    }
}
```

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

```java
package tp8.e.commerce;
import java.util.ArrayList;
import java.util.List;

public class Pedido implements Pagable {

    private List<Producto> productos = new ArrayList<>();
    private String estado = "CREADO";
    private Cliente cliente;

    public Pedido(Cliente cliente) {
        this.cliente = cliente;
    }

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

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

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

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

```java
package tp8.e.commerce;
public interface Pago {
    void procesarPago(double monto);
}
```

```java
package tp8.e.commerce;
public interface PagoConDescuento {
    double aplicarDescuento(double monto);
}
```

```java
package tp8.e.commerce;
public interface Notificable {
    void notificar(String mensaje);
}
```

```java
package tp8.e.commerce;
public class TarjetaCredito implements Pago, PagoConDescuento {

    @Override
    public void procesarPago(double monto) {
        System.out.println("Pago procesado con Tarjeta de Crédito: $" + monto);
    }

    @Override
    public double aplicarDescuento(double monto) {
        return monto * 0.90; // 10% de descuento
    }
}
```

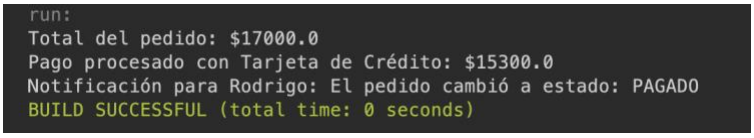
```java
package tp8.e.commerce;
public class Paypal implements Pago, PagoConDescuento {

    @Override
    public void procesarPago(double monto) {
        System.out.println("Pago procesado con PayPal: $" + monto);
    }

    @Override
    public double aplicarDescuento(double monto) {
        return monto * 0.95; // 5% de descuento
    }
}
```

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

```java
package tp8.e.commerce;
public class Cliente implements Notificable {

    private String nombre;

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

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

    @Override
    public String toString() {
        return nombre;
    }
}
```

Main:

```java
package tp8.e.commerce;
public class TP8ECommerce {
    public static void main(String[] args) {
        Cliente cliente = new Cliente("Rodrigo");

        Pedido pedido = new Pedido(cliente);

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

        double total = pedido.calcularTotal();
        System.out.println("Total del pedido: $" + total);

        TarjetaCredito tarjeta = new TarjetaCredito();
        double totalConDescuento = tarjeta.aplicarDescuento(total);

        tarjeta.procesarPago(totalConDescuento);

        pedido.cambiarEstado("PAGADO");
    }

}
```



```
run:
Total del pedido: $17000.0
Pago procesado con Tarjeta de Crédito: $15300.0
Notificación para Rodrigo: El pedido cambió a estado: PAGADO
BUILD SUCCESSFUL (total time: 0 seconds)
```

Parte 2: Ejercicios sobre Excepciones

1. **División Segura:**
   a. Solicitar dos números y dividirlos. Manejar ArithmeticException si el divisor es cero.

```java
package divisionsegura;
import java.util.Scanner;

public class DivisionSegura {
    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.");
        }
    }
}
```

```
run:
Ingrese el dividendo: 25
Ingrese el divisor: 5
Resultado: 5
BUILD SUCCESSFUL (total time: 7 seconds)
```

```
run:
Ingrese el dividendo: 30
Ingrese el divisor: 0
Error: No se puede dividir por cero.
BUILD SUCCESSFUL (total time: 2 seconds)
```

2. **Conversión de cadena a número:**
   a. Leer texto del usuario e intentar convertirlo a int. Manejar NumberFormatException si no es válido.

```java
package tp8.conversioncadenaanumero;
import java.util.Scanner;
public class TP8ConversionCadenaANumero {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            System.out.print("Ingrese un número: ");
            String texto = sc.nextLine();

            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.");
        }
    }
}
```

```
run:
Ingrese un número: 8
Número convertido: 8
BUILD SUCCESSFUL (total time: 1 second)
```

```
run:
Ingrese un número: 8.5
Error: El texto ingresado no es un número válido.
BUILD SUCCESSFUL (total time: 2 seconds)
```

3. **Lectura de archivo:**
   a. Leer un archivo de texto y mostrarlo. Manejar FileNotFoundException si el archivo no existe.

```java
package tp8.lecturaarchivo;

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

public class TP8LecturaArchivo {

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

        System.out.print("Ingrese la ruta del archivo: ");
        String ruta = sc.nextLine();

        try {
            File archivo = new File(ruta);
            Scanner lector = new Scanner(archivo);

            System.out.println("Contenido del archivo:");
            while (lector.hasNextLine()) {
                System.out.println(lector.nextLine());
            }

            lector.close();

        } catch (FileNotFoundException e) {
            System.out.println("Error: El archivo no existe.");
        }
    }
}
```

```
run:
Ingrese la ruta del archivo: /Users/roangelelli/Desktop/TextoPrueba.rtf
Contenido del archivo:
{\rtf1\ansi\ansicpg1252\cocoartf2867
\cocoatextscaling0\cocoaplatform0{\fonttbl\f0\fswiss\fcharset0 Helvetica;}
{\colortbl;\red255\green255\blue255;}
{\*\expandedcolortbl;;}
\paperw11900\paperh16840\margl1440\margr1440\vieww11520\viewh8400\viewkind0
\pard\tx720\tx1440\tx2160\tx2880\tx3600\tx4320\tx5040\tx5760\tx6480\tx7200\tx7920\tx8640\pardirnatural\partightenfactor0

\f0\fs24 \cf0 Texto de prueba - TP 8 Lectura de archivo - materia programacion 2}
BUILD SUCCESSFUL (total time: 2 minutes 30 seconds)
```

```
run:
Ingrese la ruta del archivo:  /Users/roangelelli/Desktop/TextoPrueba2.rtf
Error: El archivo no existe.
BUILD SUCCESSFUL (total time: 15 seconds)
```

4. **Excepción personalizada:**
   a. Crear EdadInvalidaException. Lanzarla si la edad es menor a 0 o mayor a 120. Capturarla y mostrar mensaje.

```java
package tp8.edadinvalida;
public class EdadInvalidaException extends Exception {

    public EdadInvalidaException(String mensaje) {
        super(mensaje);
    }
}
```

```java
package tp8.edadinvalida;
import java.util.Scanner;
public class TP8EdadInvalida {

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

     try {
         System.out.print("Ingrese una edad: ");
         int edad = sc.nextInt();

         if (edad < 0 || edad > 120) {
             throw new EdadInvalidaException("Edad fuera de rango permitido.");
         }

         System.out.println("Edad válida: " + edad);

     } catch (EdadInvalidaException e) {
         System.out.println("Error: " + e.getMessage());
     }
    }
}
```

```
run:
Ingrese una edad: 16
Edad válida: 16
BUILD SUCCESSFUL (total time: 1 second)
```

```
run:
Ingrese una edad: -9
Error: Edad fuera de rango permitido.
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
run:
Ingrese una edad: 126
Error: Edad fuera de rango permitido.
BUILD SUCCESSFUL (total time: 1 second)
```

5. **Uso de try-with-resources:**
   a. Leer un archivo con BufferedReader usando try-with-resources. Manejar IOException correctamente

```java
package tp8.lecturacontryresources;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class TP8LecturaConTryResources {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese la ruta del archivo: ");
        String ruta = sc.nextLine();

        try (BufferedReader br = new BufferedReader(new FileReader(ruta))) {

            String linea;
            System.out.println("Contenido del archivo:");
            while ((linea = br.readLine()) != null) {
                System.out.println(linea);
            }

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

```
run:
Ingrese la ruta del archivo: /Users/roangelelli/Desktop/TextoPrueba.rtf
Contenido del archivo:
{\rtf1\ansi\ansicpg1252\cocoartf2867
\cocoatextscaling0\cocoaplatform0{\fonttbl\f0\fswiss\fcharset0 Helvetica;}
{\colortbl;\red255\green255\blue255;}
{\*\expandedcolortbl;;}
\paperw11900\paperh16840\margl1440\margr1440\vieww11520\viewh8400\viewkind0
\pard\tx720\tx1440\tx2160\tx2880\tx3600\tx4320\tx5040\tx5760\tx6480\tx7200\tx7920\tx8640\pardirnatural\partightenfactor0

\f0\fs24 \cf0 Texto de prueba - TP 8 Lectura de archivo - materia programacion 2}
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
run:
Ingrese la ruta del archivo: /Users/roangelelli/Desktop/TextoPrueba2.rtf
Error al leer el archivo: /Users/roangelelli/Desktop/TextoPrueba2.rtf (No such file or directory)
BUILD SUCCESSFUL (total time: 12 seconds)
```