PROGRAMACIÓN II

Trabajo Práctico 8: Interfaces y Excepciones en Java

Alumno: Gonzalo Barrios.

GitHub: https://github.com/Goonza88/Programacion-II

Caso Practico 1:

Interfaz Pagable:

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

Clase Producto:

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

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

Clase Pedido:

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

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

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

    @Override
    public double calcularTotal() {
        double total = 0;
        System.out.println("\nPedido de compra: ");
        for (Producto p : productos) {
            System.out.println("Producto: " + p.getNombre() + ". Precio: " + p.getPrecio() + ".");
            total += p.getPrecio();
        }
        return total;
    }

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

Interfaz Pago:

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

Interfaz PagoConDescuento:

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

Clase PagoConTarjeta:

```java
public class PagoConTarjeta implements Pago {

    @Override
    public void procesarPago(double total) {
        System.out.println("\nEl total a pagar es: " + total + ".\nRealizando pago... " + "   Pago realizado.");
    }
}
```

Clase PagoConPayPal:

```java
public class PagoConPayPal implements PagoConDescuento {

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

    @Override
    public void procesarPago(double total) {
        double totalDescuento = aplicarDescuento(total);
        System.out.println("\nEl precio con el descuento es: " + totalDescuento + "..\nRealizando pago... " + "   Pago realizado.")
    }
}
```

Interfaz Notificable:

```java
public interface Notificable {
    void notificarCambioEstado(String estado);
}
```

Clase Cliente:

```java
public class Cliente implements Notificable {
    private String nombre;

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

    @Override
    public void notificarCambioEstado(String nuevoEstado) {
        System.out.println("\nCliente: " + nombre + ".\nEstado del pedido: " + nuevoEstado);
    }
}
```

Main y Resultados:

```java
public class InterfacesYExcepciones {
    public static void main(String[] args) {
        Cliente Juan = new Cliente("Juan");
        Cliente Maria = new Cliente("Maria");

        Pedido P1 = new Pedido(Juan, "Pendiente");
        Pedido P2 = new Pedido(Maria, "Pendiente");

        Producto A = new Producto("Silla", 2500.50);
        Producto B = new Producto("Mesa", 15000.00);
        Producto C = new Producto("Cortina", 3250.50);
```

```
        P1.agregarProducto(A);
        P1.agregarProducto(B);

        P1.notificarEstado("Iniciando pedido.");

        double total = P1.calcularTotal();

        PagoConTarjeta pago = new PagoConTarjeta();

        pago.procesarPago(total);

        P1.notificarEstado("Pedido completado.");
```

```
Output ×
InterfacesYExcepciones (run) ×   Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre
run:

Cliente: Juan.
Estado del pedido: Iniciando pedido.

Pedido de compra:
Producto: Silla. Precio: 2500.5.
Producto: Mesa. Precio: 15000.0.

El total a pagar es: 17500.5.
Realizando pago...    Pago realizado.

Cliente: Juan.
Estado del pedido: Pedido completado.
```

```
        P2.agregarProducto(C);

        P2.notificarEstado("Iniciando pedido.");

        double total2 = P2.calcularTotal();

        PagoConPayPal pago2 = new PagoConPayPal();

        pago2.procesarPago(total2);
    }
}
```

```
Output ×
InterfacesYExcepciones (run) ×   Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatri
Cliente: Maria.
Estado del pedido: Iniciando pedido.

Pedido de compra:
Producto: Cortina. Precio: 3250.5.

El precio con el descuento es: 2762.925..
Realizando pago...    Pago realizado.
```

Caso Practico 2:

1. División Segura:

```
System.out.println("Division Segura:");
double num, num2;

System.out.println("\nIngrese dos numeros para dividirlos:\nPrimer numero: ");
num = scan.nextDouble();

System.out.println("Segundo numero: ");
num2 = scan.nextDouble();

try {
    if (num2 == 0) {
        throw new ArithmeticException("Error. No es posible dividir por 0.");
    }
    double resultado = num / num2;
    System.out.println("El resultado es: " + resultado);
} catch (ArithmeticException ae) {
    System.out.println(ae.getMessage());
}
```

Caso Correcto:

```
Division Segura:

Ingrese dos numeros para dividirlos:
Primer numero:
10
Segundo numero:
5
El resultado es: 2.0
```

Caso Erróneo:

```
Division Segura:

Ingrese dos numeros para dividirlos:
Primer numero:
10
Segundo numero:
0
Error. No es posible dividir por 0.
```

2. Conversión de cadena a número:

```
System.out.println("\nConversion de cadena a numero:");
System.out.println("\nPorfavor ingrese un texto para convertirlo a Entero(int): ");
String texto = scan.nextLine();

try {
    int textoAInt = Integer.parseInt(texto);
    System.out.println("Texto convertido a Entero: " + textoAInt);
} catch (NumberFormatException nfe) {
    System.out.println("Error. El texto que ingreso no es un Entero.");
}
```

Caso Correcto:

```
Conversion de cadena a numero:

Porfavor ingrese un texto para convertirlo a Entero(int):
10
Texto convertido a Entero: 10
```

Caso Erroneo:

```
Conversion de cadena a numero:

Porfavor ingrese un texto para convertirlo a Entero(int):
Hola
Error. El texto que ingreso no es un Entero.
```

3. Lectura de archivo:

```
    System.out.println("\nLectura de archivo:");
    System.out.println("\nPorfavor 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 fnfe) {
        System.out.println("Error. Archivo no encontrado.");
    } catch (IOException ioe) {
        System.out.println("Error de E/S");
        System.out.println(ioe.getMessage());
    }
```

Caso Correcto:

```
Lectura de archivo:

Porfavor ingrese el nombre del archivo .txt:
C:\\Users\\Gonza\\Desktop\\Lectura.txt
Hola, este es un archivo de Lectura txt.
```

Caso Erróneo:

```
Lectura de archivo:

Porfavor ingrese el nombre del archivo .txt:
HOla
Error. Archivo no encontrado.
```

4. Excepción personalizada:

```java
public class EdadInvalidaException extends RuntimeException {

    public EdadInvalidaException() {
    }

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

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

    public EdadInvalidaException(Throwable cause) {
        super(cause);
    }

    public EdadInvalidaException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
    }
}
```

```java
System.out.println("\nExcepcion personalizada:");
System.out.println("\nIngrese su edad: ");
int edad = Integer.parseInt(scan.nextLine());

if (edad < 0 || edad > 100) {
    throw new EdadInvalidaException("Error. Edad invalida.");
} else {
    System.out.println("Edad valida y guardada.");
}
```

Caso Correcto:

```
Excepcion personalizada:

Ingrese su edad:
25
Edad valida y guardada.
```

Caso Erróneo:

```
Excepcion personalizada:

Ingrese su edad:
150
] Exception in thread "main" interfacesyexcepciones.EdadInvalidaException: Error. Edad invalida.
-       at interfacesyexcepciones.InterfacesYExcepciones2.main(InterfacesYExcepciones2.java:73)
C:\Users\Gonza\AppData\Local\NetBeans\Cache\26\executor-snippets\run.xml:111: The following error occurred while executing this line:
C:\Users\Gonza\AppData\Local\NetBeans\Cache\26\executor-snippets\run.xml:68: Java returned: 1
BUILD FAILED (total time: 2 seconds)
```

5. Uso de try-with.resources:

```
    System.out.println("\nUso de try-with-resources:");

    File archivo2 = new File("C:\\Users\\Gonza\\Desktop\\Lectura.txt");

    try (BufferedReader br2 = new BufferedReader(new FileReader(archivo2))) {
        System.out.println("\n" +br2.readLine());
    } catch (IOException ioe) {
        System.out.println("Error de E/S: " + ioe.getMessage());
    }
}
```

Caso Correcto:

```
Uso de try-with-resources:

Hola, este es un archivo de Lectura txt.
```

Caso Erróneo:

```
Uso de try-with-resources:
Error de E/S: Lectura.txt (El sistema no puede encontrar el archivo especificado)
BUILD SUCCESSFUL (total time: 0 seconds)
```