Andrés Lettieri

Programación II
Trabajo Práctico Nº 8: Interfaces y Excepciones
Link a repositorio con código:
https://github.com/AndresLettieri/UTN-P2/tree/7fc35357c0a31f6116f9f2b6b3f821e7fe9c1941/08%20Interfaces%20y%20excepciones/TP8/TP8

Parte 1: Interfaces en un sistema de E-commerce

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

```java
package tp8.Ejercicio1;
public interface Pagable {
    double calcularTotal();

}
```

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

```java
package tp8.Ejercicio1;

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

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

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

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

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

```java
package tp8.Ejercicio1;

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

```java
package tp8.Ejercicio1;

public interface PagoConDescuento {
    void aplicarDescuento(double descuento);
}
```

```java
package tp8.Ejercicio1;

public class TarjetaCredito implements PagoConDescuento, Pago{

    @Override
    public void aplicarDescuento(double descuento) {
        System.out.println("Se aplica descuento con tarjeta del " + descuento + "%");
    }

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

}
```

```java
package tp8.Ejercicio1;

public class PayPal implements PagoConDescuento, Pago {

    @Override
    public void aplicarDescuento(double descuento) {
        System.out.println("Se aplica descuento con paypal del " + descuento + "%");
    }

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

}
```

5. 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.Ejercicio1;

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

```java
package tp8.Ejercicio1;
public class Cliente implements Notificable{
    private String nombre;

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

    public String getNombre() {
        return nombre;
    }

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

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

```java
private void notificarEstado() {
    if (notificable != null) {
        notificable.notificarCambio("cambio el estado a: " + estado);
    }
}
```

Ejemplo de código corriendo:

```java
private static void ejercicio1(){
    System.out.println("Ejercicio 1\n\n");
    List<Producto> productos = new ArrayList<Producto>();
    productos.add( new Producto("Remera", 10000));
    productos.add( new Producto("Pantalon", 30000));

    Cliente cliente = new Cliente("Andres");
    Pedido pedido = new Pedido(productos, cliente, "nuevo");

    pedido.setEstado("Pagando");

    Pago metodoPago = new TarjetaCredito();

    pedido.procesarPago(metodoPago);

    if (metodoPago instanceof PagoConDescuento) {
        ((PagoConDescuento) metodoPago).aplicarDescuento(5.0);
    }
}
```

```
Ejercicio 1


Notificación para Andres: cambio el estado a: Pagando
Pago con tarjeta de $40000.0
Se aplica descuento con tarjeta del 5.0%
```

Parte 2: Ejercicios sobre Excepciones

1. División segura
2. Conversión de cadena a número

```java
private static void divisionSeguraYCadenaANumero(){
    System.out.println("1. Division segura y 2.Conversion de cadena a numero\n");
    try {
        System.out.println("Ingrese el primer numero: ");
        int num1 = Integer.parseInt(sc.nextLine().trim());
        System.out.println("Ingrese el segundo numero: ");
        int num2 = Integer.parseInt(sc.nextLine().trim());

        System.out.println("El resultado de la division es " + num1 / num2);
    } catch (ArithmeticException ae) {
        System.out.println("No se puede dividir por cero.");
    } catch (NumberFormatException nfe){
        System.out.println("Ingrese solo números.");
    }
}
```

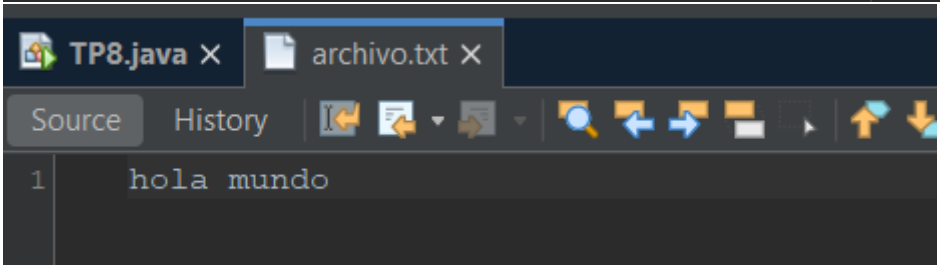
```
1. Division segura y 2.Conversion de cadena a numero

Ingrese el primer numero:
20
Ingrese el segundo numero:
5
El resultado de la division es 4
```

```
1. Division segura y 2.Conversion de cadena a numero

Ingrese el primer numero:
23
Ingrese el segundo numero:
0
No se puede dividir por cero.
```

```
1. Division segura y 2.Conversion de cadena a numero

Ingrese el primer numero:
andres
Ingrese solo números.
```

3. Lectura de archivo

```java
private static void leerArchivo(){
    System.out.println("3. Lectura de archivo.\n");
    String rutaArchivo = "src\\tp8\\archivo.txt";
    BufferedReader reader = null;
    try {
        reader = new BufferedReader(new FileReader(rutaArchivo));
        System.out.println(reader.readLine());
    }catch (FileNotFoundException e) {
        System.out.println("Error: El archivo no se encuentra en la ruta especificada.");
    }catch (IOException ex) {
        System.out.println("Error de IO. " + ex.getMessage() );
    }finally{
        try {
            reader.close();
        } catch (IOException ex) {
            System.out.println("No se pudo liberar el archivo");
        }
    }
}
```



```
3. Lectura de archivo.

hola mundo
```

4. Excepción personalizada

```java
private static void validarEdad() throws EdadInvalidaException {
    System.out.println("4. Excepcion personalizada\n");
    try {
        System.out.println("Ingrese la edad de la persona: ");
        int edad = Integer.parseInt(sc.nextLine().trim());
        if (edad < 0 || edad > 120)
        throw new EdadInvalidaException();
            System.out.println("La edad es valida");
    } catch (NumberFormatException nfe){
        System.out.println("Ingrese solo números.");
    } catch (EdadInvalidaException e) {
            //coloco catch para no interrumpir la ejecucion y completar los ejercicios en una sola corrida.
            System.out.println("Error: " + e.getMessage());
        }
}
```

```
4. Excepcion personalizada

Ingrese la edad de la persona:
123
La edad debe ser entre 0 y 120
Error: null
```

5. Uso de try-with-resources

```java
private static void tryWithResources(){
    System.out.println("5. Try with resources.");
    String rutaArchivo = "src\\tp8\\archivo.txt";
    try (  BufferedReader reader = new BufferedReader(new FileReader(rutaArchivo)); ){
        System.out.println(reader.readLine());
    }catch (FileNotFoundException e) {
        System.out.println("Error: El archivo no se encuentra en la ruta especificada.");
    }catch (IOException ex) {
        System.out.println("Error de IO. " + ex.getMessage() );
   }
}
```

```
5. Try with resources.
hola mundo
```