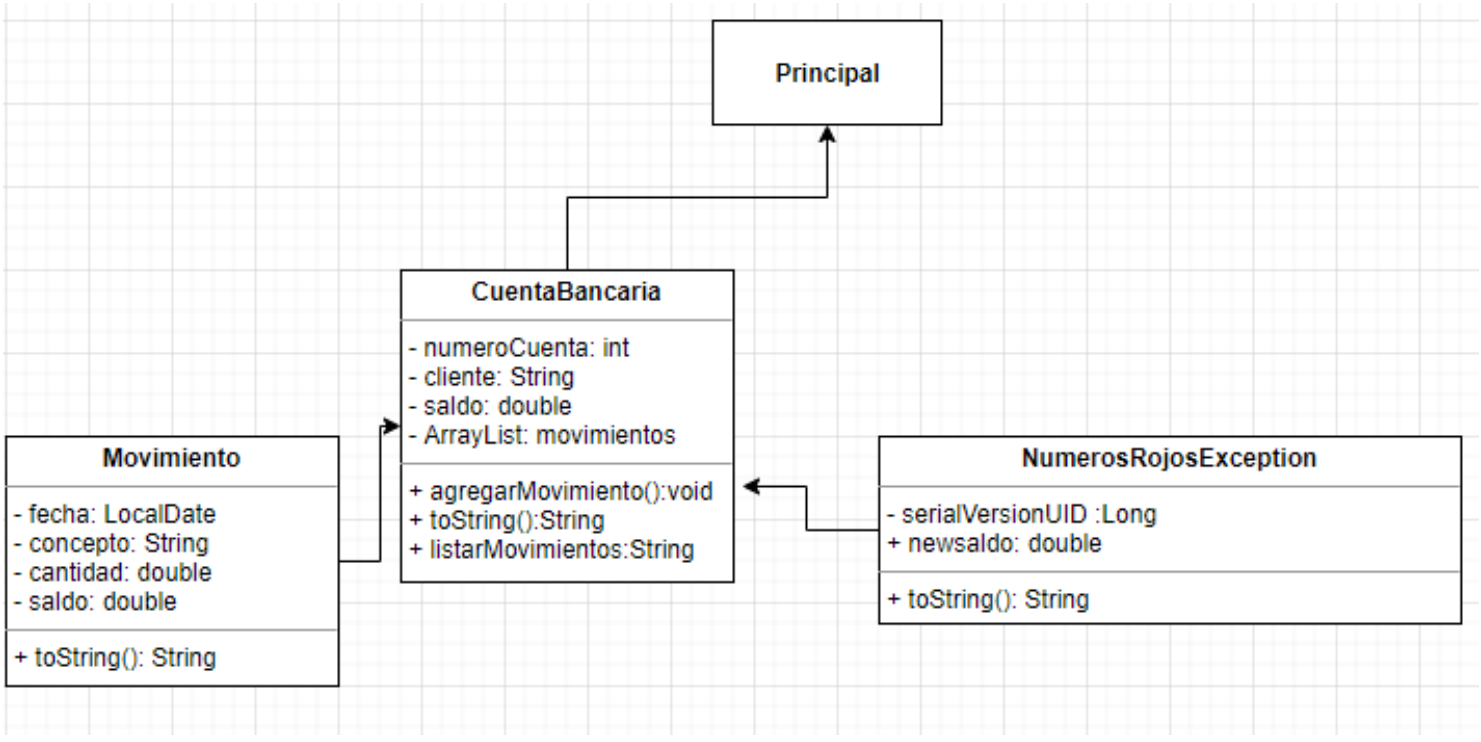
ACTIVIDAD 10. TAREA INDIVIDUAL. NUMEROSROJOSEXCEPTION

El programa consta de 4 clases, **Principal** , **Movimiento** , **CuentaBancaria** y **NumerosRojosException**

He añadido2 excepciones para el código **NumerosRojosException** y **NumberFormatException**.

Ambas tienen efecto sobre la clase Principal y están situadas en una estructura **try catch.**
Estas 2 excepciones sirven para arrojar un error avisando de que no hay fondo disponible, y por lo tanto para la operación, y un error de que la entrada de datos no admite letras solo números.



Metodo: Movimiento

```java
package actividad.pkg10.tarea.individual.numerosrojosexception;

import java.time.LocalDate;

public class Movimiento {

    private LocalDate fecha;

    private String concepto;

    private double cantidad;

    private double saldo;

    public Movimiento(String concepto, double cantidad, double saldo) {

        this.concepto = concepto;

        this.cantidad = cantidad;

        this.saldo = saldo;

        this.fecha = LocalDate.now();

    }

    @Override

    public String toString() {

        return fecha + " Concepto=" + concepto +
                ", Cantidad=" + cantidad + ", Saldo=" + saldo;

    }

}
```

Declaración de variables

Creación del constructor

Sobreescritura del método toString

Metodo: Cuenta Bancaria

```java
package actividad.pkg10.tarea.individual.numerosrojosexception;

/**
 *
 * @author Rodri
 */
import java.util.ArrayList;

public class CuentaBancaria {

    private int numeroCuenta;

    private String cliente;

    private double saldo;

    private ArrayList<Movimiento> movimientos;

    public CuentaBancaria(int numeroCuenta, String cliente) {

        this.numeroCuenta = numeroCuenta;

        this.cliente = cliente;

        this.saldo = 0;

        this.movimientos = new ArrayList<Movimiento>();
    }

    public void agregarMovimiento(String concepto, double cantidad)throws NumerosRojosException {

        this.saldo = this.saldo + cantidad;

        this.movimientos.add(new Movimiento(concepto, cantidad,saldo));
        if (this.saldo < 0){
            throw new NumerosRojosException((int) this.saldo);
        }
    }

    @Override

    public String toString() {

        return "Número= "+ numeroCuenta + ", Cliente= "+ cliente + " Saldo= "+saldo;

    }

    public String listarMovimientos() {

        String listado = "";

        for (Movimiento mov : this.movimientos) {

            listado = listado + mov.toString()+"\n";

        }

        return listado;
    }
}
```

Declaración de variables

Creación del constructor

Añado la excepción Numeros rojos

El error se produce cuando saldo sea menor que 0

Sobreescritura del método toString

Metodo listarMovimientos

Metodo: NumerosRojosException

```java
package actividad.pkg10.tarea.individual.numerosrojosexception;

/**
 *
 * @author Rodri
 */

public class NumerosRojosException extends Exception {

    private static final long serialVersionUID = 1L;
    public double newsaldo;
    public NumerosRojosException(int newsaldo) {
        super();
        this.newsaldo = newsaldo;

    }

    @Override
    public String toString() {
        return "No se puede retirar el dinero, los fondos no son suficientes";
    }


}

```

Declaración de variables

Creación del constructor

Esta metodo se utilizara cuando el dinero que saquemos sea mayor que el saldo de la cuenta

Metodo: Principal

```java
import java.util.Scanner;

public class Principal {

        public static void main(String args[]) throws NumerosRojosException {

                Scanner lector = new Scanner(System.in);

                System.out.println
("Vamos a crear una cuenta y realizar el primer ingreso de 100 euros");

                CuentaBancaria miCuenta = new CuentaBancaria(38143, "Amelia González");

                miCuenta.agregarMovimiento("Ingreso inicial ", 100);

                System.out.println("Cuánto dinero deseas retirar: ");

try {
    int dinero;

    dinero = Integer.parseInt(lector.nextLine());

    miCuenta.agregarMovimiento("Retirada de fondos ",-dinero);

    lector.close();

    System.out.println(miCuenta);

    System.out.println(miCuenta.listarMovimientos());
        }

catch (NumerosRojosException e) {
                System.out.println(e.toString());
        }
catch (NumberFormatException n) {
                System.out.println(n.getMessage()+ " Solo se aceptan numeros");
        }
        }
}
```

Utilizo la estructura try para probar el código y si hay algún error con los 2 catch recojo los errores de NumerosRojos Exception y de NumberFormat Exception

```
run:
Vamos a crear una cuenta y realizar el primer ingreso de 100 euros
Cuánto dinero deseas retirar:
20
Número= 38143, Cliente= Amelia González Saldo= 80.0
2021-02-28 Concepto=Ingreso inicial , Cantidad=100.0, Saldo=100.0
2021-02-28 Concepto=Retirada de fondos , Cantidad=-20.0, Saldo=80.0

BUILD SUCCESSFUL (total time: 6 seconds)
```

Salida cuando el dinero a retirar es menor que el saldo

```
run:
Vamos a crear una cuenta y realizar el primer ingreso de 100 euros
Cuánto dinero deseas retirar:
120
No se puede retirar el dinero, los fondos no son suficientes
BUILD SUCCESSFUL (total time: 5 seconds)
```

Salida cuando el dinero a retirar es mayor que el saldo

```
run:
Vamos a crear una cuenta y realizar el primer ingreso de 100 euros
Cuánto dinero deseas retirar:
a
For input string: "a" Solo se aceptan numeros
BUILD SUCCESSFUL (total time: 2 seconds)
```

Salida cuando no se introduce un numero