Entrega 10 Programación Juan Carlos Saldaña Herrero

Este programa simula un cajero en cuanto a que introducimos una cantidad a extraer (tenemos 100 euros) y se deduce o indica fallo en caso de no tener suficiente.

Para esto tenemos la clase cuenta bancaria, queda comentada cada parte básica de la clase, que cuenta con sus atributos privados, contructor…

CuentaBancaria.java

```java

import java.util.ArrayList;

public class CuentaBancaria {

    //Atributos
    private int numeroCuenta;
    private String cliente;
    private double saldo;
    private ArrayList<Movimiento> movimientos;

    //Constructor
    public CuentaBancaria(int numeroCuenta, String cliente) {
        this.numeroCuenta = numeroCuenta;
        this.cliente = cliente;
        this.saldo = 0;
        this.movimientos = new ArrayList();
    }

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

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

Movimiento.java

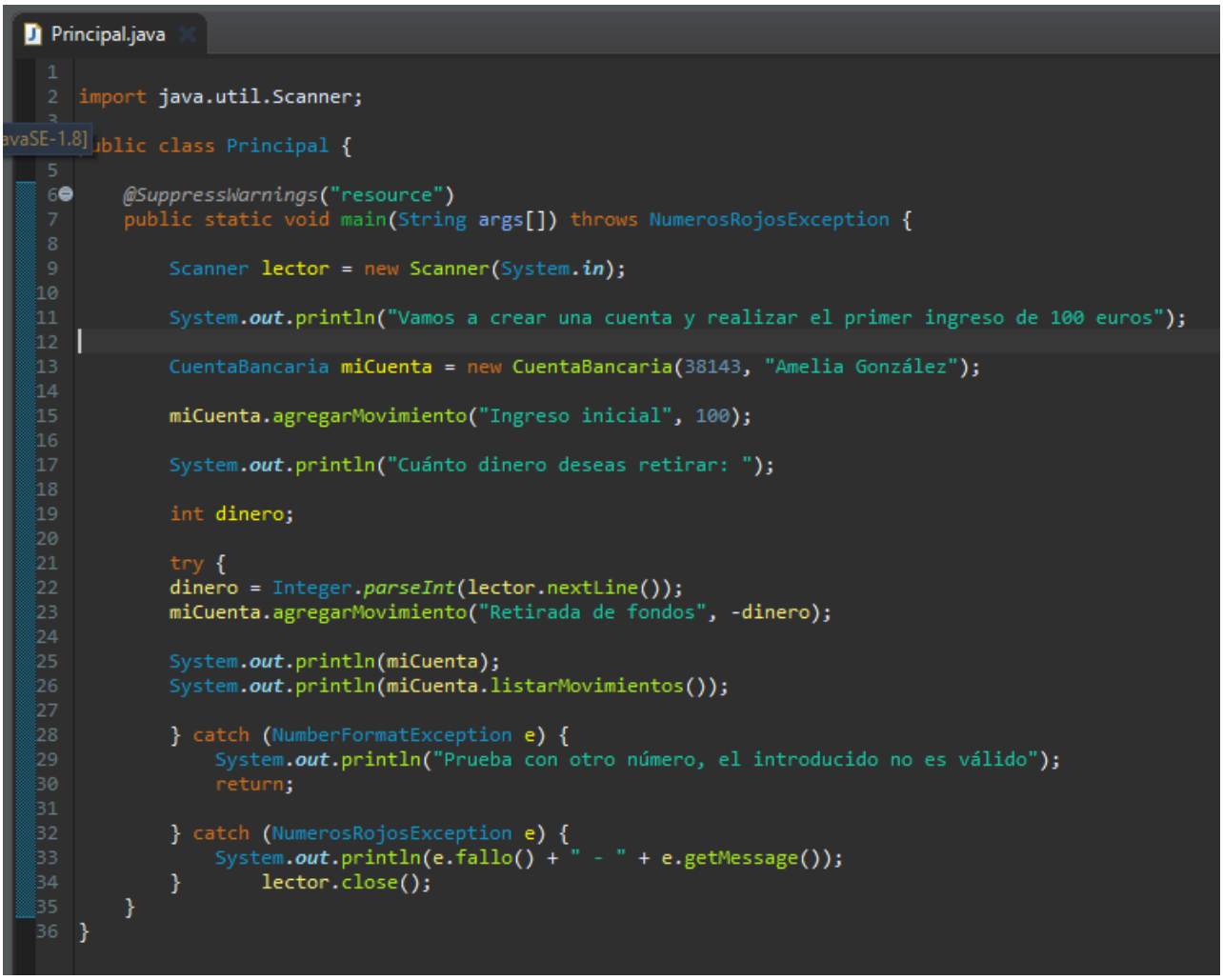
```java
import java.time.LocalDate;

public class Movimiento {
    //Atributos
    private LocalDate fecha;
    private String concepto;
    private double cantidad;
    private double saldo;

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

        this.concepto = concepto;
        this.cantidad = cantidad;
        this.saldo = saldo;
        this.fecha = LocalDate.now();
    }

    //Funciones
    @Override
    public String toString() {
        return fecha + " Concepto=" + concepto + ", Cantidad=" + cantidad + ", Saldo=" + saldo;
    }
}
```

Esta clase, movimiento, simplemente tiene sus atributos, constructor y toString.

NumerosRojosException.java

```java

public class NumerosRojosException extends Exception {

    /**
     * Generado automáticamente por la clase
     */
    private static final long serialVersionUID = 1L;

    public NumerosRojosException(double newCantidad) {
        super ("No tienes tanto dinero.");
    }

    @Override
    public String toString() {
        return "NumerosRojosException.";
    }

}

```

La clase NumerosRojosException tiene dos funciones, que indican el fallo en caso de no tener dinero suficiente o introducir número no entero. “seralVersionUID” queda generado automáticamente, su función es guardar las excepciones en un archivo externo.



```java
import java.util.Scanner;

ublic class Principal {

    @SuppressWarnings("resource")
    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: ");

        int dinero;

        try {
        dinero = Integer.parseInt(lector.nextLine());
        miCuenta.agregarMovimiento("Retirada de fondos", -dinero);

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

        } catch (NumberFormatException e) {
            System.out.println("Prueba con otro número, el introducido no es válido");
            return;

        } catch (NumerosRojosException e) {
            System.out.println(e.fallo() + " - " + e.getMessage());
        }       lector.close();
    }
}
```

Finalmente, donde se comprende el programa, tenemos la clase main. No queda comentada tal cual viene en el enunciado, pero la parte del ejercicio, try, incluye un lector para introducir la cantidad a extraer, y desencadena la función correspondiente. Tras esto, se imprime en pantalla el estado de la cuenta, y expulsa el error en caso necesario.