



UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE



CARRERA:

ELECTRÓNICA Y AUTOMATIZACIÓN

INTEGRANTES:

CHICAIZA JEFFERSON

CAISAGUANO MAYCON

STEVEN LOPEZ

ASIGNATURA:

PROGRAMACION ORIENTADA A OBJETOS

DOCENTE:

ING. CESAR OSORIO

TEMA:

COMPOSICION EN JAVA

NRC:

3725

FECHA:

25 DE JULIO DEL 2021



OBJETIVOS

General

Comprender el uso de la composición en java como se estructura y de que manera agrupa los objetos para ser utilizado en la clase principal

Específicos

- Codificar un programa que implemente clases ya impartidas por el docente.
- Codificación e idealismo de cuatro personas para un programa que contemple todas las pautas, códigos desarrollados establecidas por el docente realizando una autoeducación.

II.- ANTECEDENTES

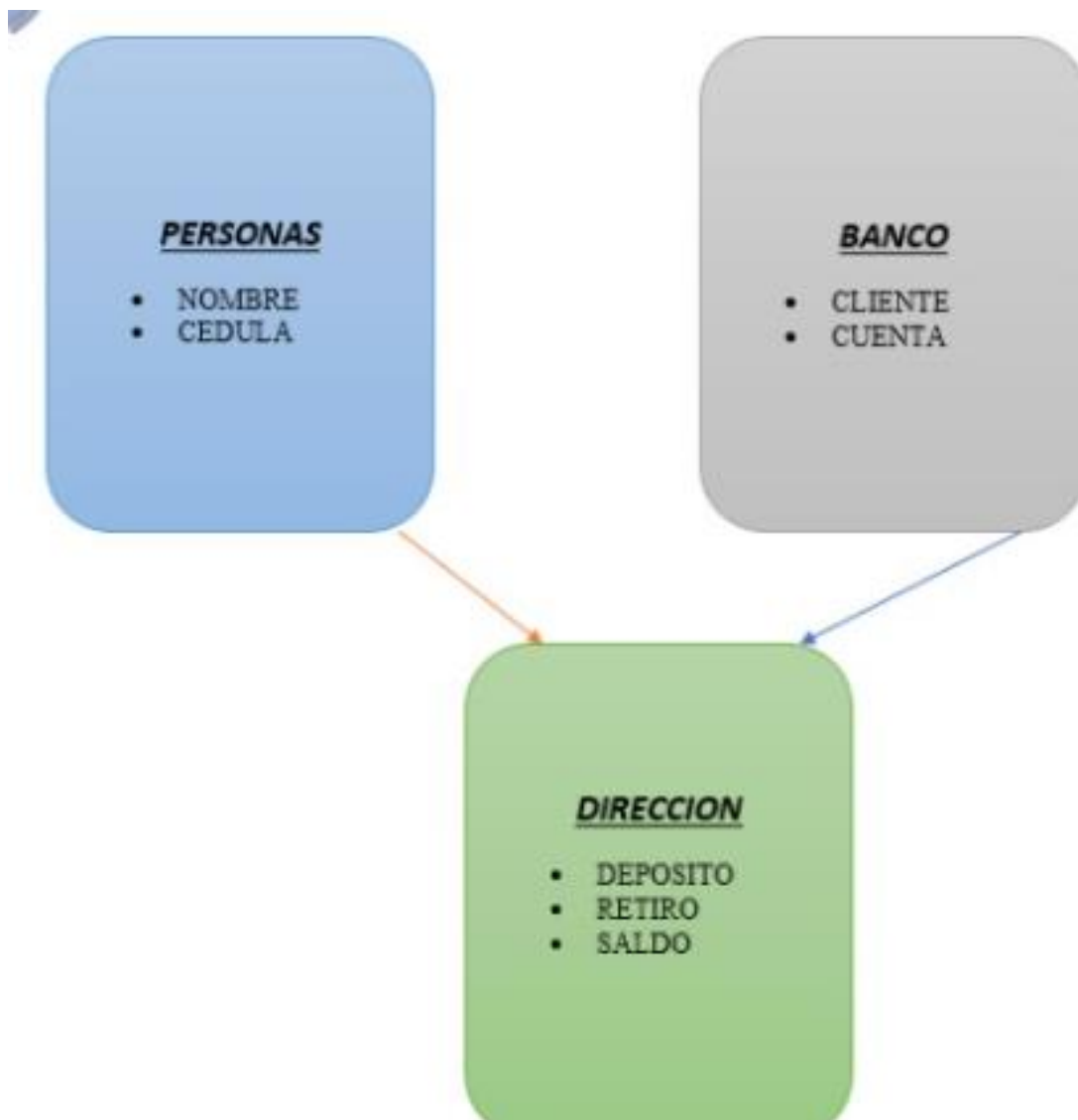
El uso de Java Composición en el trabajo diario es muy habitual. Cuando uno comienza a programar siempre parece que la herencia es la característica de un lenguaje de programación orientado a objeto que aportar mayor reutilización. Sin embargo, esto no es cierto ya que la herencia se usa normalmente para categorizar es decir por ejemplo Persona y Deportista un Deportista es una Persona. Fuera de este tipo de opciones el uso de herencia no suele encajar y se opta por un enfoque de composición una clase usa otras clases. Un ejemplo cotidiano podría ser el diseñar las Clases Persona, Empresa y Dirección.

III.- MARCO TEÓRICO

Composición en Java

La composición consiste en crear una clase nueva agrupando objetos de clases que ya existen. Una composición agrupa uno o más objetos para construir una clase, de manera que las instancias de esta nueva clase contienen uno o más objetos de otras clases. Normalmente los objetos contenidos se declaran con acceso privado y se inicializan en el constructor de la clase.

Vamos a ver este diagrama implementado en Java:



Cuando uno comienza a programar siempre parece que la herencia es la característica de un lenguaje de programación orientado a objeto que aportar mayor reutilización.

Fuera de este tipo de opciones el uso de herencia no suele encajar y se opta por un enfoque de composición una clase usa otras clases.

PROGRAMA DEL LAB (CÓDIGO)



```
Cliente.java x ComposicionJava.java x Cuenta.java x
Source History
1 package composicionjava;
2
3
4 import java.util.ArrayList;
5
6 public class Cliente {
7     private String cedula;
8     private String nombre;
9     private ArrayList<Cuenta> cuentas = new ArrayList<>();
10
11     public Cliente(String cedula, String nombre) {
12         this.cedula = cedula;
13         this.nombre = nombre;
14     }
15
16     public String getCedula() {
17         return cedula;
18     }
19
20     public String getNombre() {
21         return nombre;
22     }
23
24     public ArrayList<Cuenta> getCuentas() {
25         return cuentas;
26     }
27
28     public double consultarSaldo(int numeroCuenta) {
29         double saldo = 0;
30         return saldo;
31     }
32
33     public void ingresarDinero(int numeroCuenta, double dinero) {
34
35     }
36     public void retirarDinero(int numeroCuenta, double dinero) {
37
38     }
39
40 }
41
```



```
Cliente.java x ComposicionJava.java x Cuenta.java x
Source History
1 package composicionjava;
2
3 import java.util.Scanner;
4 public class Cuenta {
5     private int nroCuenta;
6     private double saldoCuenta;
7
8     public Cuenta(int nroCuenta, double saldoCuenta) {
9         this.nroCuenta = nroCuenta;
10        this.saldoCuenta = saldoCuenta;
11    }
12
13    /**
14     * @return the nroCuenta
15     */
16    public int getNroCuenta() {
17        return nroCuenta;
18    }
19
20    /**
21     * @return the saldoCuenta
22     */
23    public double getSaldoCuenta() {
24        return saldoCuenta;
25    }
26
27    public void depositar(double cantidad){
28        Scanner scan = new Scanner(System.in);
29        while(cantidad<0){
30            System.out.println("Cantidad ingresada no valida, reingrese: ");
31            cantidad = scan.nextDouble();
32        }
33        saldoCuenta+=cantidad;
34    }
35
36    public void retirar(double cantidad){
37        Scanner scan = new Scanner(System.in);
38        while(cantidad<0){
39            System.out.println("Cantidad ingresada no valida, reingrese: ");
40            cantidad = scan.nextDouble();
41        }
42        while(cantidad>saldoCuenta){
43            System.out.println("Cantidad no disponible para retirar, reingrese: ");
44            cantidad = scan.nextDouble();
45        }
46        saldoCuenta-=cantidad;
47    }
48 }
```



```

package composicionjava;

import java.util.ArrayList;
import java.util.Scanner;

public class ComposicionJava {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        ArrayList<Cliente> clientes = new ArrayList<>();
        clientes.add(new Cliente("1727010439", "Maycon Caisaguano"));
        int opcion;
        String cedula;
        String nombre;
        System.out.println("BIENVENIDO!!");
        for (;;) {
            System.out.println("1. Crear Cliente");
            System.out.println("2. Asignar cuenta o cuentas al cliente");
            System.out.println("3. Depositar");
            System.out.println("4. Retirar");
            System.out.println("5. Consultar Saldo");
            System.out.println("6. Salir");
            System.out.print("Ingrese la opcion deseada: ");
            opcion = scan.nextInt();
            switch (opcion) {
                case 1:
                    scan.nextLine();
                    System.out.println("Ingrese el nombre del Cliente: ");
                    nombre = scan.nextLine();
                    System.out.println("Ingrese el numero de Cedula del Cliente: ");
                    cedula = scan.nextLine();
                    clientes.add(new Cliente(cedula, nombre));
                    System.out.println("Cliente Ingresado exitosamente!!");
                    break;

                case 2:
                    scan.nextLine();
                    System.out.println("Ingrese la cedula del cliente que quiere asignar una cuenta o cuentas: ");
                    cedula = scan.nextLine();
                    for (Cliente cliente : clientes) {
                        if (cedula.equals(cliente.getCedula())) {
                            boolean bandera = true;
                            do {
                                System.out.println("Ingrese el numero de cuenta: ");
                                int nroCuenta = Integer.parseInt(scan.nextLine());
                                scan.nextLine();
                                System.out.println("Ingrese el saldo inicial de la cuenta: ");
                                double saldoCuenta = Double.parseDouble(scan.nextLine());
                                Cuenta cuenta = new Cuenta(nroCuenta, saldoCuenta);
                                cliente.getCuentas().add(cuenta);
                                System.out.println("Se ha asignado la cuenta al Cliente satisfactoriamente!!");
                                System.out.println("Desea asignar otra cuenta a este cliente?(S/N): ");
                                String opc = scan.nextLine();
                                opc = opc.toLowerCase();
                                if (opc.charAt(0) != 's') {
                                    bandera = false;
                                }
                            } while (bandera);
                            break;
                        }
                    }
                    break;

                case 3:
                    scan.nextLine();
                    System.out.println("Ingrese su cedula: ");
                    cedula = scan.nextLine();
                    for (Cliente cliente : clientes) {
                        if (cliente.getCedula().equals(cedula)) {
                            System.out.println("Ingrese el numero de la cuenta en la que depositara: ");
                            int nroCuenta = scan.nextInt();
                            for (Cuenta cuenta : cliente.getCuentas()) {
                                if (cuenta.getNroCuenta() == nroCuenta) {
                                    System.out.println("Ingrese el valor del deposito: ");
                                    double cantidad = scan.nextDouble();
                                    cuenta.depositar(cantidad);
                                }
                            }
                            break;
                        }
                    }
                    break;
            }
        }
    }
}

```



```
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128

case 4:
    scan.nextLine();
    System.out.println("Ingrese su cedula: ");
    cedula = scan.nextLine();
    for (Cliente cliente : clientes) {
        if (cliente.getCedula().equals(cedula)) {
            System.out.println("Ingrese el numero de la cuenta de la desea retirar: ");
            int nroCuenta = scan.nextInt();
            for (Cuenta cuenta : cliente.getCuentas()) {
                if (cuenta.getNroCuenta() == nroCuenta) {
                    System.out.println("Ingrese el valor del retiro: ");
                    double cantidad = scan.nextDouble();
                    cuenta.retirar(cantidad);
                }
            }
            break;
        }
    }
    break;

case 5:
    scan.nextLine();
    System.out.println("Ingrese su cedula: ");
    cedula = scan.nextLine();
    for (Cliente cliente : clientes) {
        if (cliente.getCedula().equals(cedula)) {
            System.out.println("Ingrese el numero de la cuenta en la que depositara: ");
            int nroCuenta = scan.nextInt();
            for (Cuenta cuenta : cliente.getCuentas()) {
                if (cuenta.getNroCuenta() == nroCuenta) {
                    System.out.println("El saldo de su cuenta es: " + cuenta.getSaldoCuenta());
                }
            }
            break;
        }
    }
    break;

case 6:
    System.out.println("Gracias por usar el sistema!!");
    System.exit(0);
    break;
}
```

RESULTADOS

Compilado

En la imagen Fig 1 podemos observar la parte del creado de cliente posterior a la asignación de cuenta donde se ingresa por su numero de cedula y el numero de cuenta en la creación de la misma, con un valor inicial de la cuenta en donde podemos cambiar el monto acorde al retiro o al deposito realizado.

```
BIENVENIDO!!
1. Crear Cliente
2. Asignar cuenta o cuentas al cliente
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir
Ingrese la opcion deseada: 1
Ingrese el nombre del Cliente:
maycon
Ingrese el numero de Cedula del Cliente:
1727010439
Cliente Ingresado exitosamente!!
1. Crear Cliente
2. Asignar cuenta o cuentas al cliente
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir
Ingrese la opcion deseada: 2
Ingrese la cedula del cliente que quiere asignar una cuenta o cuentas:
1727010439
Ingrese el numero de cuenta:
122597

Ingrese el saldo inicial de la cuenta:
250
Se ha asignado la cuenta al Cliente satisfactoriamente!!
```



Figura.1

En la siguiente podemos observar los resultados que nos imprime al momento de optar por la opción 3 para depositar y 5 para revisar el estado de cuenta.

```
Desea asignar otra cuenta a este cliente?(S/N):
N
1. Crear Cliente
2. Asignar cuenta o cuentas al cliente
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir
Ingrese la opcion deseada: 3
Ingrese su cedula:
1727010439
Ingrese el numero de la cuenta en la que depositara:
122597
Ingrese el valor del deposito:
200
1. Crear Cliente
2. Asignar cuenta o cuentas al cliente
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir
Ingrese la opcion deseada: 5
Ingrese su cedula:
1727010439
Ingrese el numero de la cuenta en la que depositara:
122597
El saldo de su cuenta es: 450.0
```

Esta es la vendría hacer la parte final del programa donde se pude ver la impresión del programa requerido.

```
Ingrese la opcion deseada: 4
Ingrese su cedula:
1727010439
Ingrese el numero de la cuenta de la desea retirar:
122597
Ingrese el valor del retiro:
380
1. Crear Cliente
2. Asignar cuenta o cuentas al cliente
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir
Ingrese la opcion deseada: 5
Ingrese su cedula:
1727010439
Ingrese el numero de la cuenta en la que depositara:
122597
El saldo de su cuenta es: 70.0
```




Por consiguiente tenemos la parte del retiro con el saldo actual dando como resultado la diferencia que vendría siendo la impresión dada por el saldo de cuenta.

IV.- RECOMENDACIONES

Se debe implementar los objetos desde el principio para poder componer la clase contenedora.

V.- CONCLUSIONES

En conclusión, el uso de la composición en java nos permitió añadir como se nos indica en el trabajo planteado, ya sea para ingresar los datos de la Cliente o la Cuenta en concepto con respecto al Banco de tal forma que este pueda estar ubicado en una clase diferente y tanto Cliente como Cuenta lo reutilicen. También podemos decir que la composición en java nos muestra la manera en como se van agrupando objetos y valores como atributos. Esta forma de diseñar nuestro software nos permite que el resultado sea un sistema mucho más flexible en tiempo de programación e incluso en tiempo de ejecución.