

PROGRAMACIÓN ORIENTADA A OBJETOS

Docente: Ing. Freddy Infantes Quiroz
Correo: finfantesq@upao.edu.pe



LABORATORIO 11

Colecciones II



Al término del Laboratorio enviar el Proyecto completo con el ejercicio guiado y los propuestos al

correo: finfantesq@upao.edu.pe

Asunto: [POO_Lab11_ColeccionArrayList](#)

Contenido

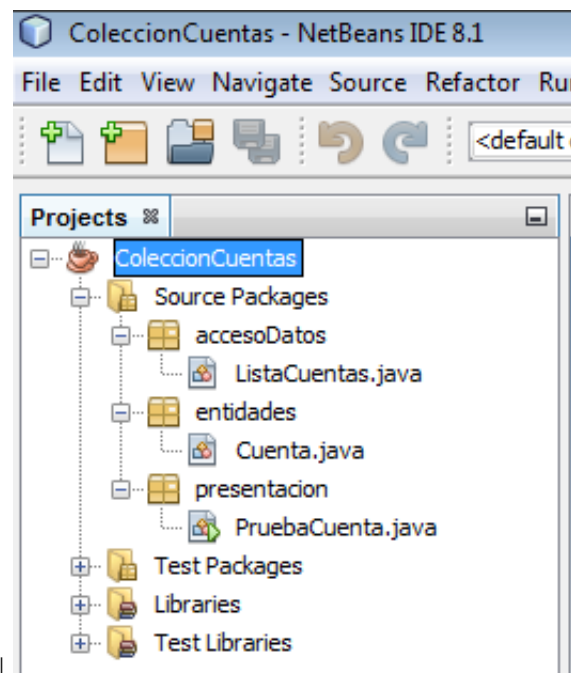
Enfatizar los conceptos en:

- Colecciones
- Uso de la Interfaces List -> **ArrayList**
 - Uso de sus métodos

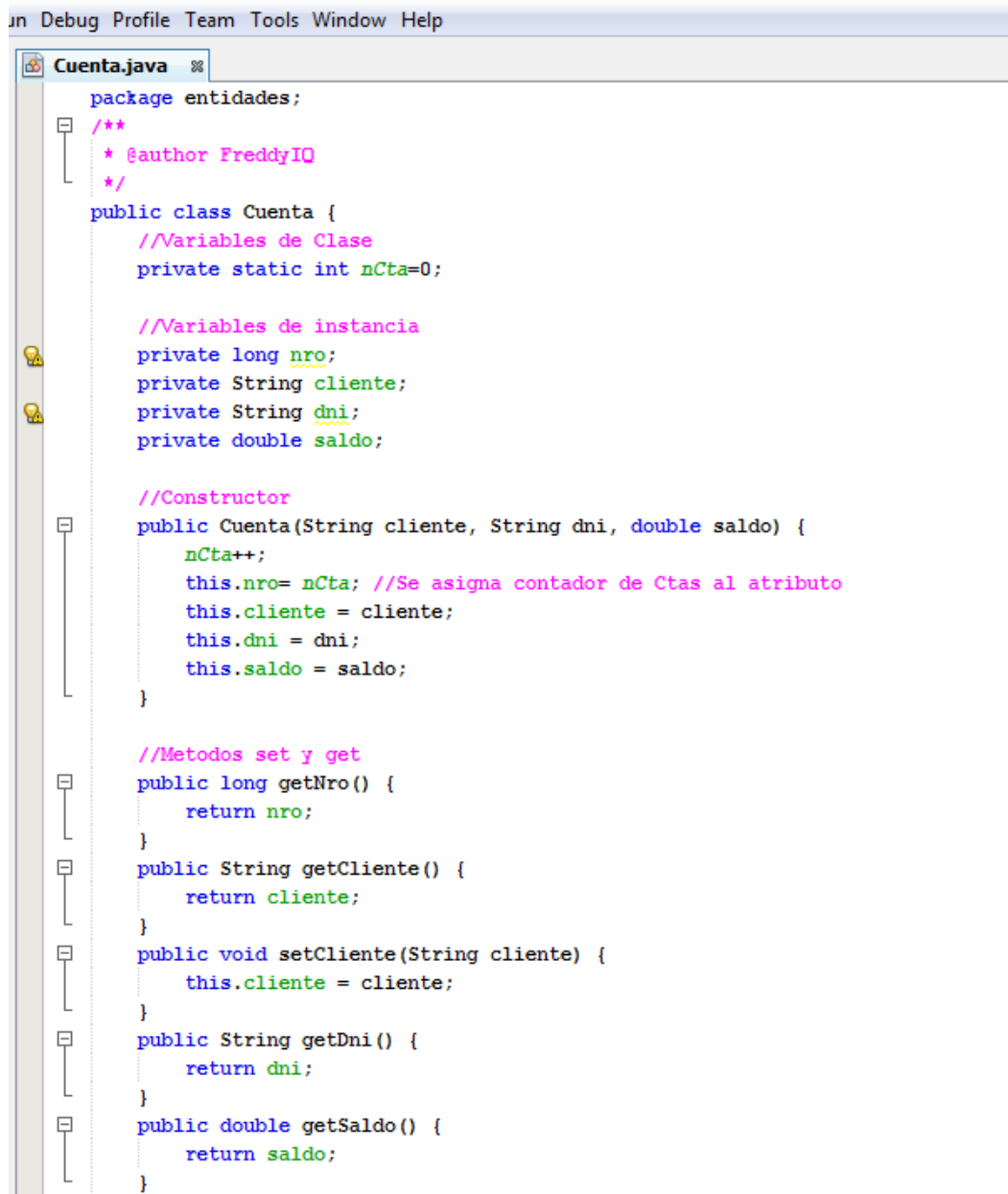
EJERCICIO – GUIADO

Crear un **Proyecto** en Netbeans con el nombre: **ColeccionCuentas_InfantesQ**, y luego crear tres **paquetes** según la siguiente estructura.

Cuenta
- nro: long
- cliente: String
- dni: String
- saldo: double
+ Cuenta()
//set y get
+ depositar(monto: double): boolean
+ retirar(monto: double): boolean
+ transferir(monto: double): boolean
+ toString(): String



Paquete entidades -> Cuenta



```
File Edit View Run Debug Profile Team Tools Window Help
Cuenta.java
package entidades;

/**
 * @author FreddyIQ
 */
public class Cuenta {
    //Variables de Clase
    private static int nCta=0;

    //Variables de instancia
    private long nro;
    private String cliente;
    private String dni;
    private double saldo;

    //Constructor
    public Cuenta(String cliente, String dni, double saldo) {
        nCta++;
        this.nro= nCta; //Se asigna contador de Ctas al atributo
        this.cliente = cliente;
        this.dni = dni;
        this.saldo = saldo;
    }

    //Metodos set y get
    public long getNro() {
        return nro;
    }
    public String getCliente() {
        return cliente;
    }
    public void setCliente(String cliente) {
        this.cliente = cliente;
    }
    public String getDni() {
        return dni;
    }
    public double getSaldo() {
        return saldo;
    }
}
```

```
public void setSaldo(double saldo) {
    this.saldo = saldo;
}

//Metodos adicionales
public boolean retirar(double monto){
    if (monto<=saldo){
        saldo -= monto; //saldo= saldo - monto
        return true;
    }
    else
        return false;
}

public boolean depositar(double monto){
    if(monto>0) {
        saldo += monto; //saldo= saldo + monto
        return true;
    }
    else
        return false;
}

public boolean transferir(Cuenta oCta, double monto){
    if (this.retirar(monto)==true){
        oCta.depositar(monto);
        return true;
    }
    else
        return false;
}

//Reporte
@Override
public String toString() {
    return "Cuenta{" + "nro=" + nro + ", cliente=" + cliente + ", dni=" + dni + ", saldo=" + saldo + '}';
}
```

Paquete accesoDatos -> Clase ListaCuentas

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ListaCuentas.java
package accesoDatos;

/**
 * @author FreddyIQ
 */
import entidades.Cuenta; //importa a clase Cuenta de otro paquete
import java.util.*; //importa colecciones

public class ListaCuentas {
    //Se declara la coleccion
    private ArrayList<Cuenta> cuentas;

    //Constructor
    public ListaCuentas()
    {
        //Se crea la coleccion
        cuentas = new ArrayList<Cuenta>();
    }

    //Metodos de mantenimiento
    public long agregarCta(String c, String d, double s )
    {
        Cuenta oC = new Cuenta(c, d, s); //se crea la cta
        cuentas.add(oC); //se agrega a la coleccion
        return oC.getNro(); //devuelve el nro cta
    }

    public Cuenta buscarCtaxNro(long nro){
        Cuenta oCta=null;
        for (Cuenta c: cuentas )
            if (nro==c.getNro())
                oCta = c;
        return oCta;
    }

    public Cuenta buscarCtaxCliente(String cli)
    {
        Cuenta oCta = null;
        //recorre la coleccion
        for(Cuenta c: cuentas )
            if(c.getCliente().equalsIgnoreCase(cli))
                oCta = c;
        return oCta;
    }

    public Cuenta buscarCtaxIndice(int i)
    {
        return cuentas.get(i);
    }

    public void eliminarCta(Cuenta oC)
    {
        cuentas.remove(oC);
    }

    public int obtenerNCtas() {
        return cuentas.size();
    }
}
```

Paquete presentacion -> PruebaCuenta

```

PruebaCuenta.java
package presentacion;

import accesoDatos.ListaCuentas;
import entidades.Cuenta;
import java.io.*;

/**
 * @author FreddyIQ
 */
public class PruebaCuenta {

    public static void main (String [] args ) throws IOException {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        int n;
        String clie, dni;
        double sal;

        //Creación de Objeto para acceder a los métodos del ArrayList
        ListaCuentas lista= new ListaCuentas();

        System.out.println("Ingrese N Cuentas a crear: ");
        n = Integer.parseInt(br.readLine());

        //Ingreso de Datos para Cuentas
        for(int i=0; i<n; i++) {
            System.out.println("\nCliente: ");
            clie = br.readLine();
            System.out.println("Dni: ");
            dni = br.readLine();
            System.out.println("Saldo: ");
            sal = Double.parseDouble(br.readLine());

            //Se agrega la cuenta a la coleccion y devuelve el nro
            long num = lista.agregarCta(clie, dni, sal);
            System.out.println("Nro Cuenta creada: " + num);
        }

        //Reporte de cuentas
        for(int i=0; i<n; i++) {
            Cuenta oC;
            oC= lista.buscarCtaxIndice(i);
            System.out.println("\nReporte");
            System.out.println(oC.toString());
        }

        //Depositar en una Cuenta
        System.out.println("\nNro Cuenta a depositar: ");
        long nro = Long.parseLong(br.readLine());

        Cuenta oC = lista.buscarCtaxNro(nro); //Busca Cuenta x nro
        if (oC==null)
            System.out.println("Cuenta no existe");
        else{
            System.out.println("Monto a depositar: ");
            double mon = Double.parseDouble(br.readLine());
            boolean resp = oC.depositar(mon);
            if (resp==true)
                System.out.println("Nuevo saldo ->" + oC.getSaldo());
            else
                System.out.println("Operacion no exitosa");
        }
    }
}

```

EJERCICIOS PROPUESTOS

Sobre el ejercicio anterior completar lo siguiente:

- Realizar algún retiro
- Realizar alguna transferencia
- Eliminar una Cuenta
- Modificar datos de una Cuenta. (Preguntando qué desea modificar)