



**UNIVERSIDAD
DE LA FRONTERA**

Introducción a la Programación

Taller 9. Listas y estructuras de control en JAVA

Octubre 2016



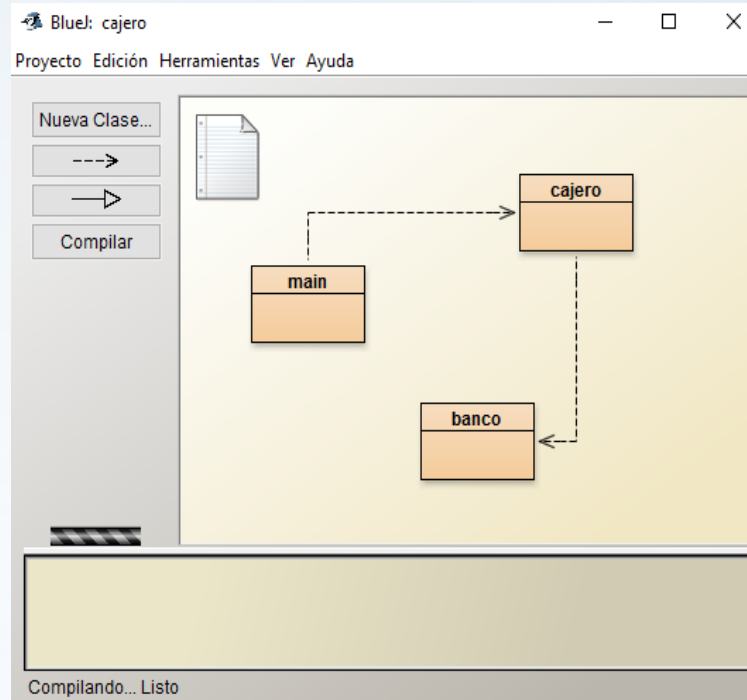
Departamento de
Computación e Informática

Actividad 1, “Cajero Automático”

A continuación construiremos una aplicación para el funcionamiento de un cajero automático. Lo primero es determinar los elementos que interactúan en el sistema. Para este caso consideraremos 2: el cajero y el banco. Si bien es cierto, el usuario también interactúa con el sistema, no lo reflejaremos en la aplicación ya que es externo a esta.

Luego, debemos definir cuales serán las operaciones (métodos) que realizará cada una:

- 1.- Cajero: Ingreso de datos usuario, menú de operaciones
- 2.- Banco: Verificación de datos del usuario, giro, depósito y mostrar saldos



Paso 1.1

Primero construiremos la clase cajero, que contendrá los métodos Ingreso de datos de usuario y menú de operaciones.

El método ingreso, se ocupará de solicitar el usuario y contraseña del cliente. Para leer estos datos, se utilizará la clase Scanner. Se debe crear un objeto («leer») de la clase de Scanner, quien se ocupará de realizar las lecturas.

Una vez leído los datos, se invocará el método «verifica» de la clase banco, la cual verificará que el usuario es cliente del banco. Si los datos del usuario coinciden con los datos almacenados por el banco, entonces se puede acceder a las transacciones ofrecidas por el cajero.

```

import java.util.Scanner;

public class cajero
{
    // instance variables - replace the example below with your own
    private int usuario;
    private int pass;
    private Scanner leer = new Scanner(System.in);
    banco objBanco = new banco();

    public void ingreso(){
        System.out.println("Bienvenido al sistema de cajero automático");
        System.out.println("Por favor ingrese su usuario");
        usuario = leer.nextInt();
        System.out.println("Por favor, Ingrese su contraseña");
        pass = leer.nextInt();
        if (objBanco.verifica(usuario,pass)){
            System.out.println("Datos correctos");
            menu(usuario);
        }
        else{
            System.out.println("Usuario o contraseña incorrectos");
        }
    }
}
  
```

The IDE window shows the implementation of the **cajero** class. It includes imports, instance variables for **usuario**, **pass**, a **Scanner** object **leer**, and a **banco** object **objBanco**. The **ingreso()** method handles user input and calls the **verifica** method of the **banco** class.

```

Compilar  Deshacer  Cortar  Copiar  Pegar  Encontrar...  Cerrar  Implementación

public void menu(int usr){
    int opc=0;
    int monto;

    do{
        System.out.println("Bienvenido - Elija una opción");
        System.out.println("1.- Giro");
        System.out.println("2.- Depósito");
        System.out.println("3.- Saldo");
        System.out.println("4.- Salir");
        opc = leer.nextInt();
        switch(opc){
            case 1: System.out.println("Ingrese monto a girar");
                    monto = leer.nextInt(); objBanco.giro(monto,usr); break;
            case 2: System.out.println("Ingrese monto a depositar");
                    monto = leer.nextInt(); /*objBanco.deposito(monto,usr);*/ break;
            case 3: objBanco.muestraSaldo(usr); break;
            case 4: System.out.println("Gracias por su operación"); break;
        }
    }while(opc!=4);
}

```

Paso 1.2

A continuación construimos el método menú, el cual desplegará en pantalla, las opciones para el usuario: Giro, depósito y mostrar saldo. Cada opción invoca a un método respectivo en la clase banco.

```

Compilar  Deshacer  Cortar  Copiar  Pegar  Encontrar...  Cerrar

/**
 * Write a description of class banco here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class banco
{
    private int posCliente;

    private int [] usuarios = {111,222};
    private int [] claves = {111,222};
    private int [] saldos = {100,0};
}

```

Paso 1.3

Ahora, trabajaremos sobre la clase banco. Lo primero será definir los elementos que contendrán los datos del usuario: nombre, clave y saldo. Para esto, utilizaremos una lista (arreglo), para cada elemento. Para efectos del ejemplo, consideraremos solo dos clientes.

```

Compilar  Deshacer  Cortar  Copiar  Pegar  Encontrar...  Cerrar

public boolean verifica(int cliente,int clave){
    posCliente = busqueda(cliente);
    if (posCliente >= 0){
        if (clave == claves[posCliente]){
            return true;
        }
        else{
            return false;
        }
    }
    else{
        return false;
    }
}

```

Paso 1.4

A continuación, construiremos el método que verifica que el usuario que se ha identificado en el cajero es cliente del banco. Éste método lo hemos definido boolean, ya que responderá con un verdadero falso. Éste método invoca al método búsqueda, quien indicará si existe una coincidencia entre el nombre de usuario ingresado y los clientes registrados.

Paso 1.5

Ahora construiremos el método búsqueda, que nos dirá la posición dentro de la lista, en la que se encuentra el usuario identificado en el cajero. Si el usuario no existe en el registro del banco, entonces devolverá un -1.



```

Compilar  Deshacer  Cortar  Copiar  Pegar  Encontrar...  Cerrar

private int busqueda(int cliente){
    int pos=-1;
    for(int i=0;i<usuarios.length;i++){
        if (cliente == usuarios[i]){
            pos = i;
        }
    }
    return pos;
}

```

Paso 1.6

Por último, construiremos el métodos giro y mostrar saldo. Giro requiere del nombre de usuario y del monto a girar, mientras que mostrar saldo, requiere solo del nombre de usuario.



```

Compilar  Deshacer  Cortar  Copiar  Pegar  Encontrar...  Cerrar

public void giro(int valor,int cliente){
    posCliente = busqueda(cliente);
    saldos[posCliente] = saldos[posCliente] - valor;
    System.out.println("Operación realizada exitosamente");
}

public void muestraSaldo(int cliente){
    posCliente = busqueda(cliente);
    System.out.println("Su saldo es: "+saldos[posCliente]);
}
}

```

Actividad 2, “Producto”

Para terminar la actividad:

1.- Construya el método para realizar depósitos.

FINISHED?
FINISHED?



**UNIVERSIDAD
DE LA FRONTERA**

Curso de Introducción a la Programación

Taller 9. Listas y Estructuras de Control en JAVA

UNIVERSIDAD DE LA FRONTERA

FACULTAD DE INGENIERÍA Y CIENCIAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

Avda Francisco Salazar 01145
Temuco – Chile / casilla 54-D
Fono (56) 45 2325000 /2744219

dci.ufro.cl



Departamento de
Computación e Informática