

# Programación Orientada a Objetos

## Agregación de clases

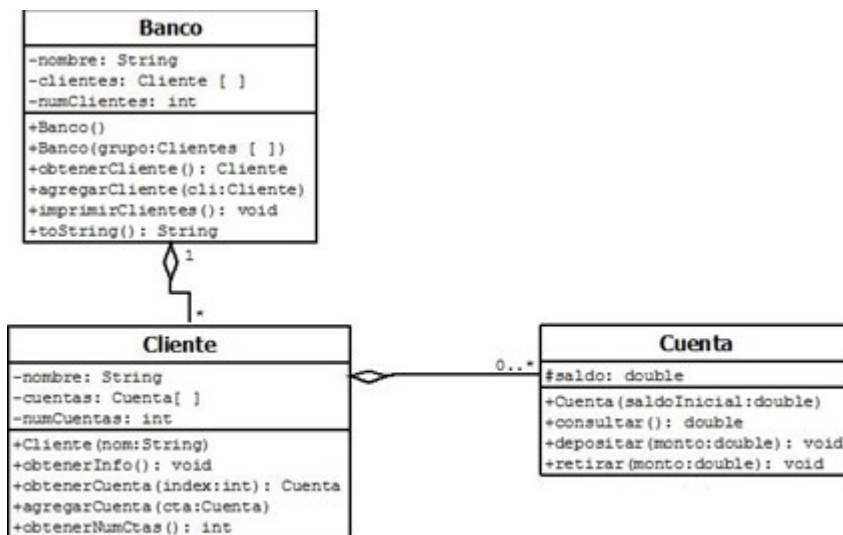
### Práctica Opcional: Banco-Cliente-Cuenta

#### Objetivo:

Aplicar el concepto de Agregación de Clases mediante la implementación de las clases: Banco, Cliente, y Cuenta.

#### Desarrollo:

Con base en el siguiente diagrama UML, realiza lo que se pide:



**En la clase usuario de la clase Banco (UsuarioBanco) realiza lo siguiente:**

- Crea al menos dos instancias de la clase Banco.
- Asigna al menos cuatro objetos de la clase Cliente a cada objeto creado de la clase Banco.
- Cada objeto Cliente debe tener al menos dos objetos de la clase Cuenta.
- Modificar la clase Cliente para que incluya el atributo NIP (número de Identificación Personal), el cual es la clave de acceso para poder realizar cualquier operación bancaria (depositar, retirar, consultar). El NIP está constituido por cuatro dígitos, y es generado de forma aleatoria al momento de crearse una instancia de la clase Cliente.

- Realiza las operaciones: depositar, retirar, consultar, para cada uno de los objetos de la clase Cliente. Cada operación bancaria debe incluir el NIP para que la operación se realice de forma satisfactoria.
- Lleva a cabo el registro de cada movimiento (depósito, retiro, consulta) de cada objeto de la clase Cliente.
- Reporta los movimientos (tipo -depósito o retiro- y número de movimientos correspondientes, para cada cuenta) de cada uno de los clientes de cada objeto de la clase Banco, usando el método **imprimirClientes( )**. Se debe proporcionar toda la información de un cliente (excepto el NIP).
- Hacer búsquedas de clientes ya sea por número de cuenta o por el nombre del cliente, y proporcionar la información completa de este cliente (operaciones bancarias y datos del cliente).
- Incorporar la hora y la fecha para cada movimiento bancario y reportarlas en el método **imprimirClientes( )**. Para resolver este apartado se requiere incluir una nueva clase en el diagrama UML: la clase Fecha.

**Nota:** Para resolver esta práctica puedes agregar una nueva clase al diagrama UML, o bien, métodos a las clases existentes; en cualquier caso se debe preservar la esencia del diagrama UML original.

## Evaluación:

Concepto	Valor (Participaciones extra)
Diseño general UML.	4
Realizar operaciones bancarias (retirar, depositar, y consultar) para cada una de las cuentas de los clientes.	3
Reportar las operaciones bancarias de cada uno de los clientes de un determinado Banco.	3
Incorporación de la hora y de la fecha en cada una de las operaciones bancarias (retirar depositar, consultar) del reporte.	2

## Entrega:

- Presentar el diagrama UML que se usó como base para esta práctica. El código debe corresponder con el diagrama.
- Presentar el programa en ejecución.
- Sustentar un breve examen oral acerca del código.
- No es necesario entregar reporte escrito,

**Fecha de entrega:**

Fecha de caducidad: viernes 19 de octubre de 2019.