



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

GUÍA DE PRÁCTICA DE LABORATORIO / TALLER

| | | |
|---|--|--|
| CARRERA: Electrónica y Automatización | GUÍA: NO. 04 | TIEMPO ESTIMADO: 1h y 20min. |
| ASIGNATURA: Programación Orientada a Objetos NRC: 3725 | FECHA DE ELABORACIÓN: 27/07/2021 SEMESTRE: mayo 2021 – septiembre 2021 | |
| TÍTULO: Relación de Composición | DOCENTE: Ing. Cesar Osorio | |
| GRUPO N°3 INTEGRANTES: | Andrango José, Chalán Dilan, Flores Steven, Guanoliquin Brayam, Quishpi Jordan, | |

I. OBJETIVOS

- Investigar y crear un algoritmo que simule el sistema financiero de un banco, para tener un conocimiento más amplio sobre relaciones en java, mediante el conocimiento adquirido en clase con el instructor a cargo.

OBJETIVOS ESPECÍFICOS:

- Uso e implementación de Composición en Java.
- Analizar y Determinar el uso de Relaciones: Agregación, composición, uno a muchos.
- Codificar de manera didáctica y con bases en los aprendizajes adquiridos.

II. INSTRUCCIONES

- Utilice como material principal, aquel indicado en clase por el docente.
- Utilice información consultada en Internet y conocimiento adquirido en clase.

III. ANTECEDENTES

Para que un lenguaje de programación pueda ser considerado orientado a objetos, debe implementar el mecanismo de herencia. La relación de herencia entre dos clases es una relación binaria entre dos clases que nos dice: La clase subclase o derivada hereda de la clase base o superclase. La agregación en Java es una relación entre dos clases que se describe mejor como una relación "tiene-a" y "todo / parte". La clase agregada contiene una referencia a otra clase y se dice que tiene la propiedad de esa clase.

IV. MARCO TEÓRICO

LA ASOCIACIÓN

En una asociación, dos instancias A y B relacionadas entre sí existen de forma independiente. No hay una relación fuerte. La creación o desaparición de uno de ellos implica únicamente la creación o destrucción de la relación entre ellos y nunca la creación o destrucción del otro. Por ejemplo, un cliente puede tener varios pedidos de compra o ninguno.

La relación de asociación expresa una relación (unidireccional o bidireccional) entre las instancias a partir de las clases conectadas. El sentido en que se recorre la asociación se denomina navegabilidad de la asociación. Cada extremo de la asociación se caracteriza por el rol o papel que juega en dicha relación el objeto situado en cada extremo. La cardinalidad o multiplicidad es el número mínimo y máximo de instancias que pueden relacionarse con la otra instancia del extremo opuesto de la relación. Por defecto es 1. El formato en el que se especifica es (mínima, máxima).

Es una relación estructural que describe una conexión entre objetos. Suelen ser bidireccionales pero es importante a veces hacerlas unidireccionales para restringir su navegación en un solo sentido.

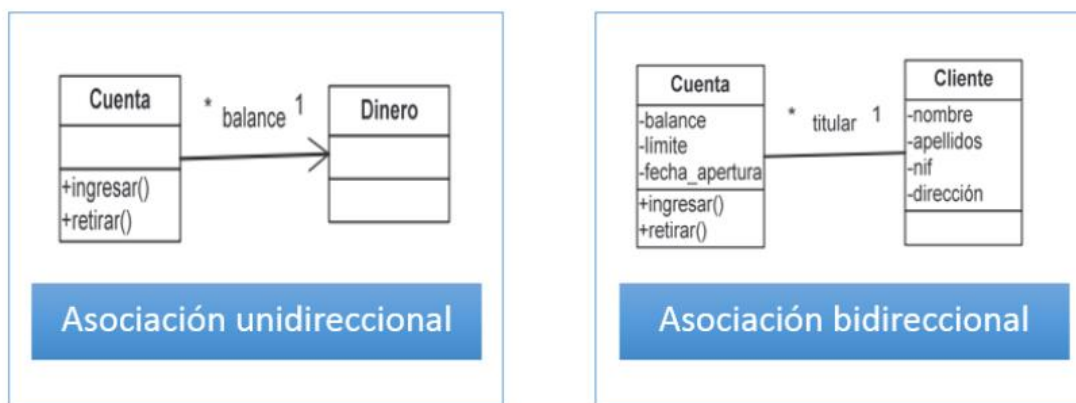


Figura Nº1

■ Relaciones de Asociación

- ☐ Al menos, un atributo de la ClaseB es una referencia a un objeto de ClaseA.
 - Nota: A veces puede ser uno y otras el otro
 - ☐ No tiene por que ser siempre el mismo.
- ☐ La creación de B no implica la creación de A.
- ☐ Durabilidad: **Duradera**

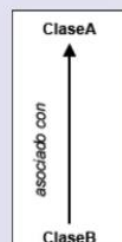


Figura Nº2

Multiplicidad de asociaciones

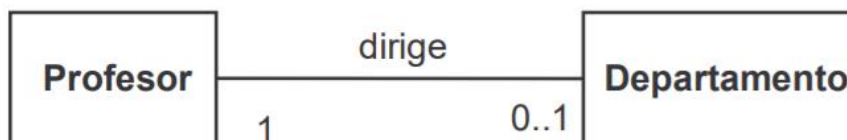
Determina cuantos objetos de cada tipo intervienen en la relación, en otras palabras es el número de instancias de una clase que se relacionan con una instancia de otra clase. Cada asociación tiene dos multiplicidades, una para cada extremo de la relación.

Tipos de multiplicidad

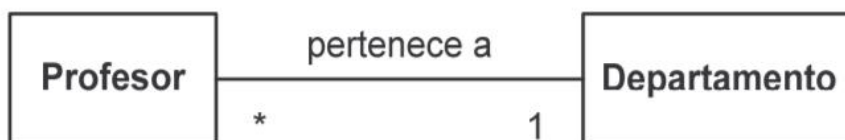
| Multiplicidad | Significado |
|---------------|-----------------------------|
| 1 | Uno y sólo uno |
| 0..1 | Cero o uno |
| N..M | Desde N hasta M |
| * | Cero o varios |
| 0..* | Cero o varios |
| 1..* | Uno o varios (al menos uno) |

Figura N°3

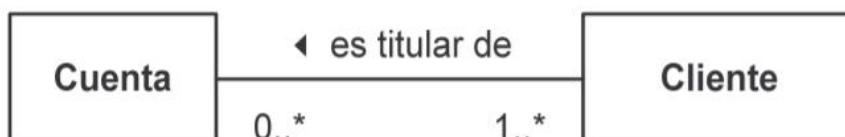
Ejemplos:



Todo departamento tiene un director.
Un profesor puede dirigir un departamento.



Todo profesor pertenece a un departamento.
A un departamento pueden pertenecer varios profesores.



Relación opcional
Un cliente puede o no
ser titular de una cuenta

Relación obligatoria
Una cuenta ha de tener
un titular como mínimo

Figura N°4

AGREGACIÓN Y LA COMPOSICIÓN

En una relación todo-parte una instancia forma parte de otra. En la vida real se dice que A está compuesto de B o que A tiene B. La diferencia entre asociación y relación todo-parte radica en la asimetría presente en toda relación todo-parte. En teoría se distingue entre dos tipos de relación todo-parte:

La agregación es una asociación binaria que representa una relación todo-parte (pertenece a tiene un, es parte de). Por ejemplo, un centro comercial tiene clientes.

La composición es una agregación fuerte en la que una instancia 'parte' está relacionada, como máximo, con una instancia 'todo' en un momento dado, de forma que cuando un objeto 'todo' es eliminado, también son eliminados sus objetos 'parte'. Por ejemplo: un rectángulo tiene cuatro vértices, un centro comercial está organizado mediante un conjunto de secciones de venta...

A nivel práctico se suele llamar agregación cuando la relación se plasma mediante referencias (lo que permite que un componente esté referenciado en más de un compuesto). Así, a nivel de implementación una agregación no se diferencia de una asociación binaria. Por ejemplo: un equipo y sus miembros. Por otro lado, se suele llamar composición cuando la relación se conforma en una inclusión por valor (lo que implica que un componente está como mucho en un compuesto, pero no impide que haya objetos componentes no relacionados con ningún compuesto). En este caso si se destruye el compuesto se destruyen sus componentes. Por ejemplo: un ser humano y sus miembros. Algunas relaciones pueden ser consideradas agregaciones o composiciones, en función del contexto en que se utilicen.

Agregación

Las partes pueden formar parte de distintos agregados.

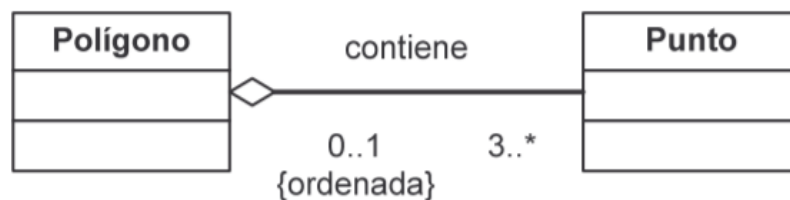


Figura Nº5

Composición

Agregación disjunta y estricta:

Las partes sólo existen asociadas al compuesto
(sólo se accede a ellas a través del compuesto)

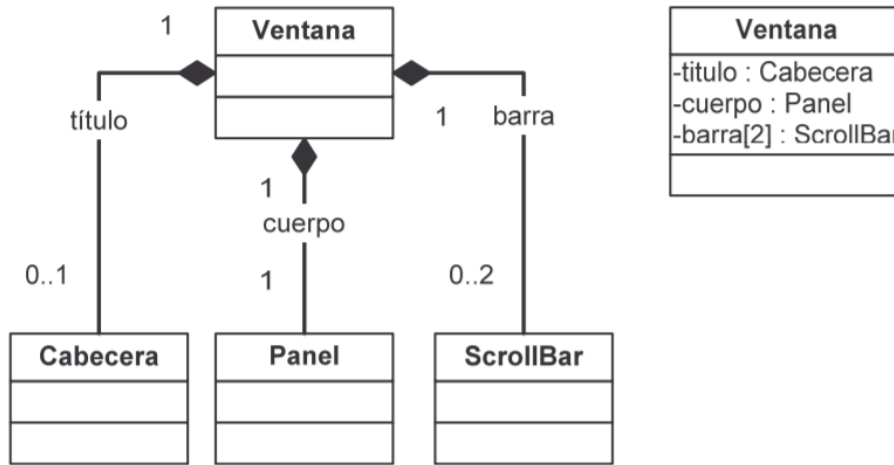


Figura N°6

EJEMPLOS DE AGREGACIÓN O COMPOSICIÓN

Como se ha comentado anteriormente, la agregación o composición son mecanismos diferentes de la herencia que consiste en que uno o más atributos de una clase pertenecen a una o más clases previamente declaradas. Es decir, un objeto puede componerse de otros pertenecientes a otras clases.

Por ejemplo, la clase Persona se compone de dos variables de instancia, una de la clase String y otra de la clase Fecha:

```
/**
 * Declaracion de la clase Persona
 */
public class Persona {
    String nombre;
    Fecha fechaNacimiento;
    public void asignaDatos(String nombre, Fecha f) {
        this.nombre = nombre;
        this.fechaNacimiento = f;
    }

    public String toString() {
        return nombre + " nacido el dia " + fechaNacimiento.toString();
    }
}
```

Se dice que la clase Persona es una agregación de las clases String y Fecha. La clase PruebaPersona muestra un ejemplo de uso de la clase Persona:

```
/**
 * Ejemplo de uso de la clase Persona
 */
```

```
public class PruebaPersona {  
    public static void main (String [] args ) {  
        Persona p = new Persona();  
        Fecha n = new Fecha(11,2,2002);  
        p.asignaDatos("Arkaitz Garro", n);  
        System.out.println(p.toString());  
    } }  
}
```

La salida por pantalla al ejecutar el código anterior es:

```
$>java PruebaPersona
```

```
Arkaitz Garro nacido el dia 11/2/2002
```

V. RESULTADOS OBTENIDOS

```
**Menú de Banco**
```

1. Crear Cliente
2. Asignar Cuenta/s
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir

```
Digite una Opción: 1
```

```
Usted a seleccionado Crear Cliente
```

```
Espere un momento...
```

```
Cédula del Cliente: 123456
```

```
Nombre del Cliente: johao flores
```

```
Nro de cuentas que desea abrir el Cliente JOHAO FLORES: 2
```

```
Datos de la Cuenta N° 1
```

```
Nro de Cuenta: 20021
```

```
Saldo de cuenta: $50
```

```
Datos de la Cuenta N° 2
```

```
Nro de Cuenta: 20022
```

```
Saldo de cuenta: $30
```

****Menú de Banco****

1. Crear Cliente
2. Asignar Cuenta/s
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir

Digite una Opción: 2

Usted a seleccionado Asignar Cuenta/s
Espere un momento...

****Asignación de Cuentas****

Nombre: JOHAO FLORES

Cedula: 123456

Nro de Cuenta: 20021

Saldo: \$50.0

Nro de Cuenta: 20022

Saldo: \$30.0

Saldo Total de las Cuentas: \$80.0

****Menú de Banco****

1. Crear Cliente
2. Asignar Cuenta/s
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir

Digite una Opción: 3

Usted a seleccionado Depositar

Espere un momento...

Ingrese Nro de Cuenta al que desea depositar: 20021

Ingrese la cantidad a depositar 10

Su saldo actual es: \$60.0

****Menú de Banco****

1. Crear Cliente
2. Asignar Cuenta/s
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir

Digite una Opción: 4

Usted a seleccionado Retirar

Espere un momento...

Ingrese el Nro de Cuenta para realizar el retiro: 20021

Ingrese la cantidad a retirar: 45

Dinero retirado

Su Saldo Actual es: 15.0

****Menú de Banco****

1. Crear Cliente
2. Asignar Cuenta/s
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir

Digite una Opción: 5

Usted a seleccionado Consultar Saldo

Espere un momento...

Ingrese el Nro de Cuenta a Consultar: 20021

El Saldo de la cuenta ingresada es: \$15.0

****Menú de Banco****

1. Crear Cliente
2. Asignar Cuenta/s
3. Depositar
4. Retirar
5. Consultar Saldo
6. Salir

Digite una Opción: 6

Usted a seleccionado Salir

Gracias por preferirnos

BUILD SUCCESSFUL (total time: 2 minutes 18 seconds)

VI. CONCLUSIONES

- Aprendimos que la agregación es un tipo de asociación que indica que una clase es parte de otra clase con un funcionamiento específico.
- Es claro que en multiplicidades cada asociación contiene dos de estas siendo así una para cada extremo de la relación.
- Concluimos que las relaciones pueden contener asociaciones unidireccionales y también bidireccionales

VII. RECOMENDACIONES

- Se requiere profundizar el tema para poder tener una comprensión más eficaz al momento del desarrollo del ejercicio planteado.
- Es muy importante recordar como es el planteamiento requerido en nuestro código para obtener resultados eficaces
- Considerar las clases impartidas como un material de apoyo para aclarar dudas al momento de estar construyendo nuestro programa de relaciones.