



Relaciones entre clases

ASOCIACIONES

Asociación

- La asociación es la relación entre dos clases separadas que se establece a través de sus Objetos.
- Puede ser de uno a uno, de uno a varios, de varios a uno, de varios a varios.

Ejemplo

```
// clase Escuela
class Escuela
{
    private String nombre;
    Escuela(String nombre)
    {
        this.nombre = nombre;
    }
    public String getNombreEsc()
    {
        return this.nombre;
    }
}
```

```
// Clase Alumno
class Alumno
{
    private String nombre;
    Alumno(String nombre)
    {
        this.nombre = nombre;
    }
    public String getNombreAlum()
    {
        return this.nombre;
    }
}
```

```
public class TieneAsocia
{
    public static void main (String[] args)
    {
        Escuela esc = new Escuela("FCC");
        Alumno grupo[]=new Alumno[3];
        grupo[0]=new Alumno("Tere");
        grupo[1]=new Alumno("Marcos");
        grupo[2]=new Alumno("Olivia");

        System.out.println(grupo[1].getNombreAlum() +
            " Es un alumno de " +
            esc.getNombreEsc());    }
}
```

Dos clases separadas Escuela y Alumno están asociadas a través de sus Objetos. La Escuela puede tener muchos Alumnos, por lo que es una relación de uno a muchos.

Agregación

- ¿Es una asociación unidireccional, pues una clase contiene a la otra.?
- ¿La agregación consiste en hacer referencia a una clase dentro de otra clase.?
- ¿Es una relación débil entre el objeto contenedor y el contenido. Esto significa que la destrucción de uno no implica la destrucción del otro.?
- ¿Se usa para la reutilización del código?

Ejemplo

- Programa para llevar el registro de cuáles Alumnos tienen pc (será la de escritorio) propio y cuáles no, así como de cuáles tienen portátil y cuáles no.
- Alumno “tiene” un pc y/o que el Alumno “tiene” una portátil, se trata de una relación de agregación entre la clase a la que pertenecerá el Alumno y dos clases adicionales, una para la pc y otra para la portátil.

Ejemplo

```
class portatil{
    String marca;
    String color;
    String serie;
    portatil(String marca,String color,
String serie){this.marca=marca;
                this.color=color;
                this.serie=serie;
            }
}
```

```
class Pc{
    String marca;
    String serie;
    Pc(String marca, String serie){
        this.marca=marca;
        this.serie=serie;}
}
```

```
public class Estudiante{
    String nomb Kemp;
    int num Kemp;
    portatil portatil;
    Pc Pc;

    Estudiante(String nomb Kemp, int num Kemp, portatil portatil, Pc Pc){
        this.nomb Kemp=nomb Kemp;
        this.num Kemp=num Kemp;
        this.portatil=portatil;
        this.Pc = Pc;
    }
}
```



```
public static void main(String[] args){
    portatil portatil = new portatil("XY", "Azul", "102SD");
    Pc Pc = new Pc("TR", "1569TYUB");
    Estudiante e = new Estudiante("Margarita López",202215489,portatil,Pc);
    System.out.println("Los Datos del Estudiante son");
    System.out.println("Nombre: " + e.nombremp);
    System.out.println("Matrícula: " + e.numemp);
    System.out.println("Y tiene una portátil con");
    System.out.println("marca, de " + e.portatil.marca);
    System.out.println("color y con "+ e.portatil.color);
    System.out.println("serie " + e.portatil.serie);
    System.out.println("También tiene una Pc");
    System.out.println("de marca "+ e.Pc.marca);
    System.out.println("y serie " + e.Pc.serie);

}

}
```

Composición

- ¿Se logra mediante el uso de una variable de instancia que hace referencia a otros objetos. ?
- ¿Si un objeto contiene el otro objeto y el objeto contenido no puede existir sin la existencia de ese objeto?
- ¿Es una forma de describir la referencia entre dos o más clases usando una variable de instancia y se debe crear una instancia antes de usarla?
- ¿La composición permite la reutilización de código.?



Ejemplo

- Los Alumnos y Escuelas, tienen dirección

Ejemplo

```
private String nombre;  
private int edad;  
  
private Direccion dirección;  
  
public Alumno(String nombre, int edad) {  
    super();  
    this.nombre = nombre;  
    this.edad = edad;  
}  
public Direccion getDirección() {  
    return dirección;  
}  
public void setDirección(Direccion dirección) {  
    this.dirección = dirección;  
}
```

```
public String getNombre() {  
    return nombre;  
}  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
public int getEdad() {  
    return edad;  
}  
public void setEdad(int edad) {  
    this.edad = edad;  
}
```

```
class Escuela {
    private String registro;
    private Direccion dirección;

    public Direccion getDirección() {
        return dirección;
    }
    public void setDirección(Direccion dirección) {
        this.dirección = dirección;
    }
    public String getregistro() {
        return registro;
    }
    public void setregistro(String registro) {
        this.registro = registro;
    }
}
```

```
class Direccion {
    private String calle;
    private int numero;

    public String getCalle() {
        return calle;
    }
    public void setCalle(String calle) {
        this.calle = calle;
    }
    public int getNumero() {
        return numero;
    }
    public void setNumero(int numero) {
        this.numero = numero;
    }
    public Direccion(String calle, int
numero) {
        super();
        this.calle = calle;
        this.numero = numero;
    }
}
```

```
public class Principal {
    public static void main(String[] args) {

        Direccion d1= new Direccion("calle 2 oriente",125);
        Direccion d2= new Direccion("calle privada 3 sur",25);
        Alumno a= new Alumno("Angeles",30);
        a.setDirección(d1);
        Escuela e= new Escuela();
        e.setregistro("XYCVF");
        e.setDirección(d2);

        System.out.println("Dirección del alumno " + a.getDirección().getCalle());
        System.out.println("Dirección de su escuela "+ e.getDirección().getCalle());
    }
}
```