Algoritmos

Contenido:

- Programación Orientada a Objetos
  - Relaciones de Composición

Composición

Las relaciones de composición son relaciones entre **parte** y **todo** en las que la parte debe constituir parte del objeto completo.

La carpeta puede contener muchos archivos, mientras que cada archivo tiene exactamente una carpeta principal. **Si se elimina una carpeta, también se eliminan todos los archivos contenidos**

Composición

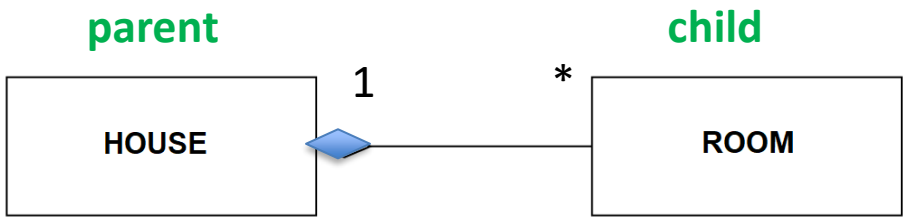
Por ejemplo, **un corazón es parte del cuerpo de una persona**. La parte de una composición solo puede ser parte de un objeto a la vez. **Un corazón que es parte del cuerpo de una persona** no puede ser parte del cuerpo de otra al mismo tiempo.

Otro ejemplo de la vida real es Nuestro **cuerpo y sangre**. La sangre reside dentro de un cuerpo. Esto significa que **la sangre es parte del cuerpo**. La sangre es inútil si no está dentro de un cuerpo. La sangre presente fuera del cuerpo no tiene importancia.

Composición

La relación de composición es una forma especializada de agregación. La composición es en realidad un tipo fuerte de agregación

En composición, **si el objeto padre se destruye**, **los objetos hijo también dejan de existir**.. Por ejemplo, una casa puede estar compuesta por una o más habitaciones.
Si la casa se destruye, todas las habitaciones que forman parte de la casa también se destruyen




```
public class House
{
    private Room room;
    public House()
    {
        room = new Room();
    }
}
```

Composición

Es una fuerte forma de agregación

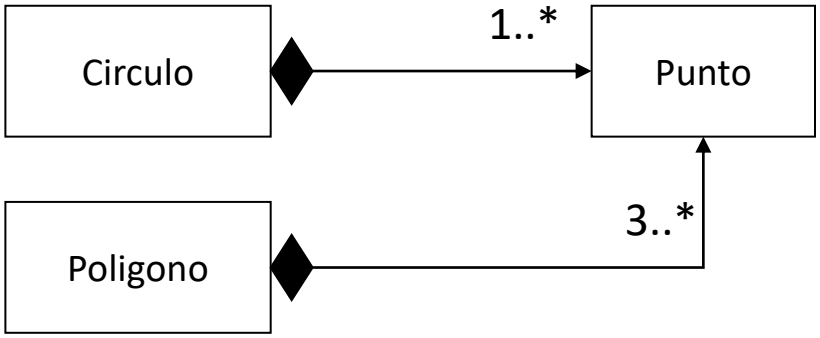
El tiempo de vida de la “parte” depende del “todo”

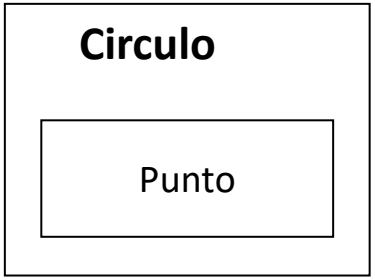
El conjunto es el único propietario de sus partes y la “parte” solamente pertenece a un solo conjunto

El tiempo de vida de la “parte” depende del “todo”

La multiplicidad en “todo” el lado debe ser cero o uno

El conjunto debe gestionar la creación y destrucción de sus partes





Composición