

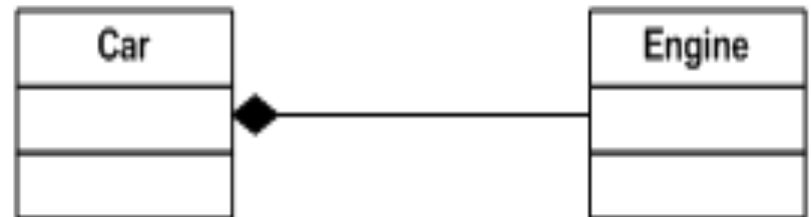
Relaciones entre Clases

Modelado con UML e
implementación con Java

Composición

- Es una relación entre clases del tipo <<... es parte de ...>>
- La clase que es <<parte de>> es definida como un atributo de la clase de la que es parte.

```
public class Car {  
    //final will make sure engine is initialized  
    private final Engine engine;  
  
    public Car(){  
        engine = new Engine();  
    }  
}  
  
class Engine {  
    private String type;  
}
```

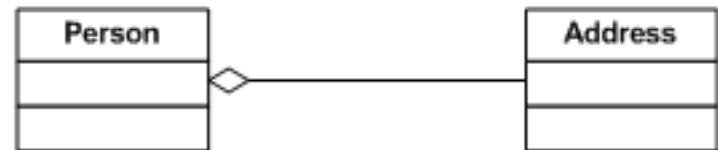


Agregación

- Es una relación entre clases del tipo <<... tiene un ...>>
- La clase que es “tenida” se define como un atributo dentro de la clase que “la tiene”.

```
public class Address {  
    ...  
}
```

```
public class Person {  
    private Address address;  
    public Person(Address address) {  
        this.address = address;  
    }  
    ...  
}
```



Comparando las relaciones...

Composición	Agregación
<< ...es parte de ...>>	<<... tiene un ...>>
La clase se define como un atributo de la clase que la contiene	La clase se define como un atributo de la clase que la contiene
Los objetos se instancian y asignan valores dentro de la clase	Los objetos se instancia fuera de la clase y se asignan atributos en el constructor enviándolo como argumento de la clase

Ejemplo detallado

- Para ver un ejemplo detallado se recomienda analizar, descargar y ejecutar el ejemplo del siguiente link

<http://stackoverflow.com/questions/6668615/aggregation-and-composition-representation-in-class-definition/6668981#6668981>