**Composition**

Composite aggregation is depicted as a binary association decorated with a **filled black diamond** at the aggregate (whole) end.

Composite aggregation is depicted as binary associations with filled black diamond

Folder could contain many files, while each File has exactly one Folder parent.  
If Folder is deleted, all contained Files are deleted as well.

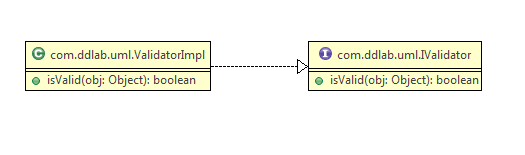
#### Composition (Not-Shared Association)

In cases where in addition to the part-of relationship between ClassA and ClassB - there’s a strong life cycle dependency between the two, meaning that when ClassA is deleted then ClassB is also deleted as a result, we should be more specific and use the composition link instead of the aggregation link or the association link.

[](http://lh5.ggpht.com/_aUOgqE3fGXc/Sh35YNKDw9I/AAAAAAAAAao/E4v4uDJcD5w/s1600-h/image%5B5%5D.png)

The composition link shows that a class (container, whole) has exclusive ownership over other class/s (parts), meaning that the container object and its parts constitute a parent-child/s relationship.

**Realization**



Java Code

**package** com.ddlab.uml;

**public** **interface** IValidator {

**public** **boolean** isValid(Object obj);

}

**package** com.ddlab.uml;

**public** **class** ValidatorImpl **implements** IValidator {

@Override

**public** **boolean** isValid(Object obj) {

**return** **false**;

}

}

**Aggregation**

package com.ddlab.uml;

import java.util.ArrayList;

import java.util.List;

public class Faculty {

private List<Student> studentsList ;

public Faculty() {

studentsList = new ArrayList<Student>();

}

}

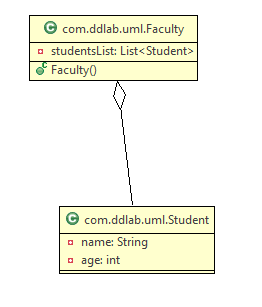
package com.ddlab.uml;

public class Student {

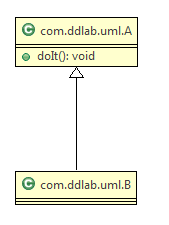
private String name;

private int age;

}



Generalization



package com.ddlab.uml;

public class A {

public void doIt() {

}

}

package com.ddlab.uml;

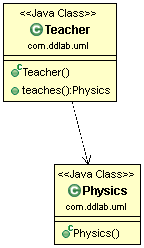
public class B extends A {

}

**Composition**

**In composition** (Person, Heart, Hand), "sub objects" (Heart, Hand) will be destroyed as soon as Person is destroyed.

**Dependencies**



package com.ddlab.uml;

public class Teacher {

public Physics teaches() {

return new Physics();

}

}

package com.ddlab.uml;

public class Physics {

}