Aggregation in Java

If a class have an entity reference, it is known as Aggregation. Aggregation represents HAS-A relationship.

Consider a situation, Employee object contains many informations such as id, name, emailId etc. It contains one more object named address, which contains its own informations such as city, state, country, zipcode etc. as given below.

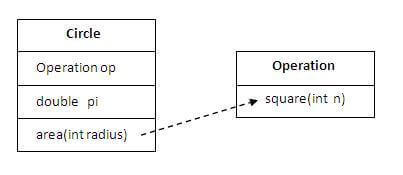
1. **class** Employee{
2. **int** id;
3. String name;
4. Address address;//Address is a class
5. ...
6. }

In such case, Employee has an entity reference address, so relationship is Employee HAS-A address.

Why use Aggregation?

* For Code Reusability.

Simple Example of Aggregation



In this example, we have created the reference of Operation class in the Circle class.

1. **class** Operation{
2. **int** square(**int** n){
3. **return** n\*n;
4. }
5. }
7. **class** Circle{
8. Operation op;//aggregation
9. **double** pi=3.14;
11. **double** area(**int** radius){
12. op=**new** Operation();
13. **int** rsquare=op.square(radius);//code reusability (i.e. delegates the method call).
14. **return** pi\*rsquare;
15. }


19. **public** **static** **void** main(String args[]){
20. Circle c=**new** Circle();
21. **double** result=c.area(5);
22. System.out.println(result);
23. }
24. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=Circle)

Output:78.5

When use Aggregation?

* Code reuse is also best achieved by aggregation when there is no is-a relationship.
* Inheritance should be used only if the relationship is-a is maintained throughout the lifetime of the objects involved; otherwise, aggregation is the best choice.

Understanding meaningful example of Aggregation

In this example, Employee has an object of Address, address object contains its own informations such as city, state, country etc. In such case relationship is Employee HAS-A address.

Address.java

1. **public** **class** Address {
2. String city,state,country;
4. **public** Address(String city, String state, String country) {
5. **this**.city = city;
6. **this**.state = state;
7. **this**.country = country;
8. }
10. }

Emp.java

1. **public** **class** Emp {
2. **int** id;
3. String name;
4. Address address;
6. **public** Emp(**int** id, String name,Address address) {
7. **this**.id = id;
8. **this**.name = name;
9. **this**.address=address;
10. }
12. **void** display(){
13. System.out.println(id+" "+name);
14. System.out.println(address.city+" "+address.state+" "+address.country);
15. }
17. **public** **static** **void** main(String[] args) {
18. Address address1=**new** Address("gzb","UP","india");
19. Address address2=**new** Address("gno","UP","india");
21. Emp e=**new** Emp(111,"varun",address1);
22. Emp e2=**new** Emp(112,"arun",address2);
24. e.display();
25. e2.display();
27. }
28. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=Emp)

Output:111 varun

gzb UP india

112 arun

gno UP india