**Fajar Zuhri Hadiyanto**
Posted on Oct 7, 2021



# Types of Relation Between Classes in Object Oriented Programming

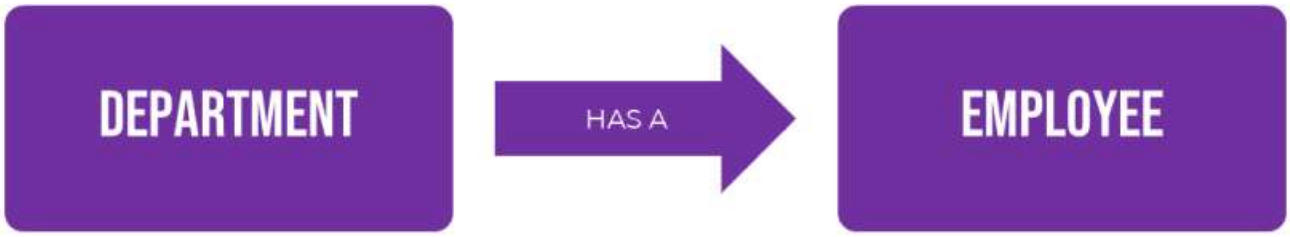
#oop #java #programming

Object-oriented programming is a programming paradigm that sees everything as an object. Object-oriented programming is adapted from a real-world problem. For example, you have a cat, so do i. Your cat and mine have the same properties, such as weight and color. Your cat and mine also have similar behavior, such as eating, sleeping, and walking. Your cat is an object and mine is another object. We can classify these objects into one class, just say it Cat class.

In object-oriented programming, class is a blueprint for objects that we want to create. One class can be related to another class or not. Generally, there are some relations between classes in oop, which are :

- aggregation
- composition
- inheritance

## Aggregation

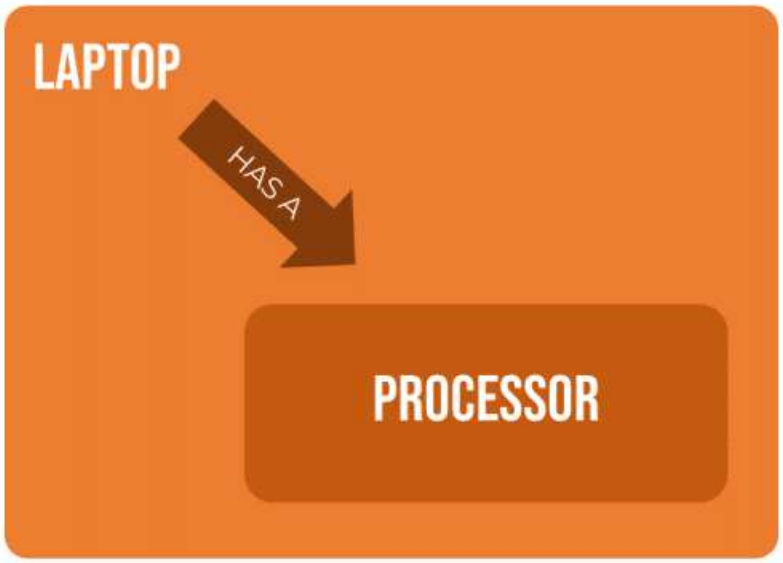



Aggregation is a Has-A relationship between two objects where each object can exist without another object. In other words, these objects are independent. For example, the relationship between employee and department. An employee can stand alone without a department, so does the department.

```
public class Employee {
    private String name;
    private double salary;

    public Employee() {}
}

public class Department {
    public String name;
    public List<Employee> employees;

    public Department() {}
}
```

The source code above is an example of the implementation of aggregation in java. The Department class has a relationship with the Employee class. Department object can have a list of employees or not at all. Employee object might belong to Department object or not. There is no restriction in the aggregation relationship.

# Composition

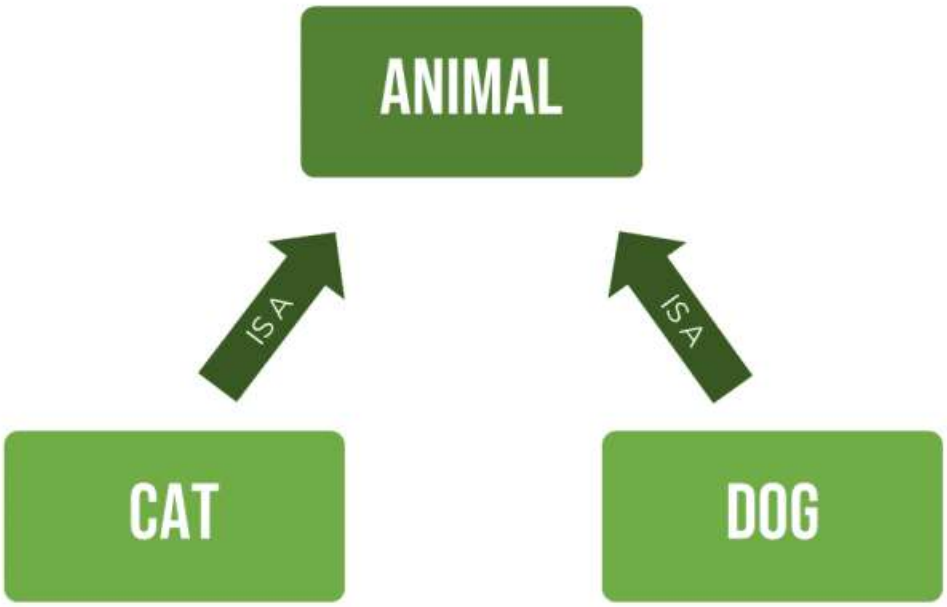



Composition is a Has-A relationship between classes where both classes are dependent on each other. One object cannot exist without the existence of another object. For example, the relation between Laptop and its Processor. A laptop cannot exist without its processor.

```
public class Processor {
    private String modelName;
    private int frequency;

    public Processor () {}
}

public class Laptop {
   private Processor processor = new Processor();

   public Laptop () {}
}
```

The source code above is an example of the implementation of composition in java. The Laptop class has a relationship with the Processor class. When a laptop object is created, a processor object that belongs to that laptop is automatically created.

# Inheritance




Inheritance is an Is-A relationship between classes where parent class is a general class and child class is a specific class. For example, the relation between cat, dog, and its general class, which is animal. An animal can be specified as a cat or dog. Cat and dog can be generalized as an animal.

```
public class Animal {
    protected weight;
    protected color;

    public Animal () {}

    public void talk () {
        System.out.println("It is animal");
    }
}

public class Cat extend Animal {
    public Cat () {}

    public void talk () {
        System.out.println("meow");
    }
```

```
}

public class Dog extend Animal {
    public Dog () {}

    public void talk () {
        System.out.println("bark");
    }
}
```

The source code above is an example of the implementation of inheritance in java. The Cat and The Dog both have properties of weight and color. To generalize those properties of Cat and Dog, we can create one parent class that can represent both cat and dog with those properties, which is Animal class.

## Top comments (0)

Code of Conduct • Report abuse

## Fajar Zuhri Hadiyanto

Student of informatics Engineering

**LOCATION**
Jakarta

**EDUCATION**
Institut Teknologi Sepuluh Nopember

**JOINED**
Oct 7, 2021

## Trending on DEV Community 🔥

How to get the Top Software Engineering Practices Voice Badge on LinkedIn
#softwareengineering #discuss #tutorial #coding

JS Design Patterns: A Comprehensive Guide
#javascript #programming #frontend #webdev

Refactorización de código JavaScript para Startups: Eficiencia con Set y Map
#webdev #javascript #programming #spanish

DEV Community


**Check out this survey and help us moderate our community by becoming a tag moderator here at DEV.**