

# Java Polymorphism

Topics : [JAVA](#)

Written on [April 10, 2023](#)

Java polymorphism is the ability of an object to take on many forms, i.e., an object can have multiple types. In Java, polymorphism is achieved through inheritance, interfaces, and method overloading/overriding.

There are two types of polymorphism in Java: compile-time polymorphism (also known as static or method overloading) and runtime polymorphism (also known as dynamic or method overriding).

Compile-time polymorphism is achieved through method overloading, which allows a class to have multiple methods with the same name but different parameters. When an overloaded method is called, the Java compiler determines which method to call based on the number, type, and order of the arguments passed in.

Here's an example of method overloading:

```
public class MyClass {
    public void print(String message) {
        System.out.println(message);
    }

    public void print(int number) {
        System.out.println(number);
    }

    public void print(double number) {
        System.out.println(number);
    }
}

public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public int add(int a, int b, int c) {
        return a + b + c;
    }
}
```

In this example, the `Calculator` class has two `add()` methods with different numbers of parameters. Depending on how many arguments are passed in, the Java compiler will call the appropriate method.

Runtime polymorphism (also known as method overriding) occurs when a subclass provides its own implementation of a method that is already defined in its superclass. The Java runtime system determines which method to call based on the type of the object at runtime.

Here's an example of method overriding in Java:

```
public class Animal {
    public void makeSound() {
        System.out.println("The animal makes a sound");
    }
}

public class Dog extends Animal {
    public void makeSound() {
        System.out.println("The dog barks");
    }
}
```

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)