BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on Interface in Java.

Interface in Java:-

An interface in Java is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

Java Interface also **represents the IS-A relationship**.

It cannot be instantiated just like the abstract class.

Since Java 8, we can have **default** and **static methods** in an interface.

Since Java 9, we can have **private methods** in an interface.

Why use Java interface?

There are mainly three reasons to use interface. They are given below.

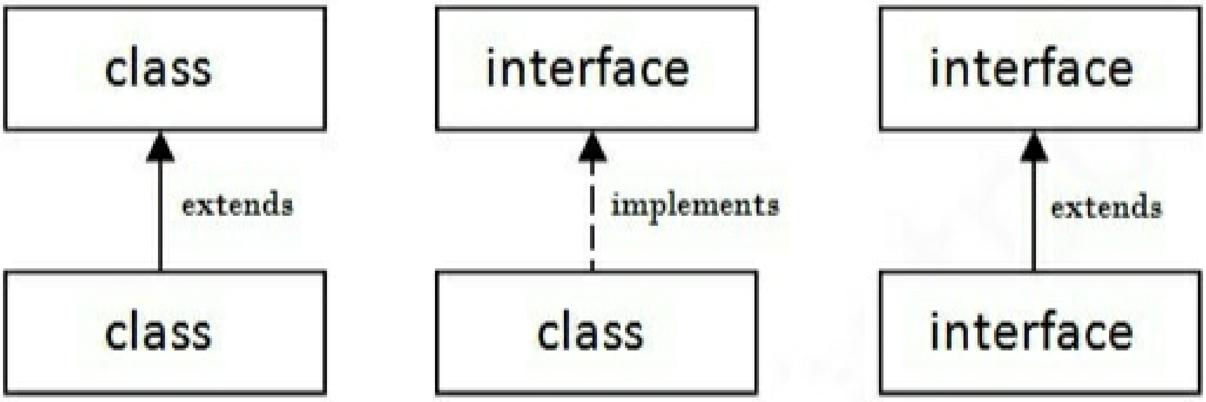
- It is used to achieve abstraction.
- By interface, we can support the functionality of

multiple inheritance.

- To achieve security - hide certain details and only show the important details of an object (interface).
- Java does not support "multiple inheritance" (a class can only inherit from one superclass). However, it can be achieved with interfaces, because the class can implement multiple interfaces.
- It can be used to achieve loose coupling. (Coupling is nothing but the dependency of one class on the other. If one object in a code uses the other object in the program, it is called loose coupling in Java. When two classes, modules, or components have low dependencies on each other, it is called loose coupling in Java. Loose coupling in Java means that the classes are independent of each other.)

The relationship between classes and interfaces:-

As shown in the figure given below, a class extends another class, an interface extends another interface, but a class implements an interface.



An interface is similar to a class in the following ways−

- An interface can contain any number of methods.
- An interface is written in a file with a .java extension, with the name of the interface matching the name of the file.
- The byte code of an interface appears in a .class file.
- Interfaces appear in packages, and their corresponding bytecode file must be in a directory structure that matches the package name.

However, an interface is different from a class in several ways, including −

- You cannot instantiate an interface.
- An interface does not contain any constructors.
- All of the methods in an interface are abstract.
- An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final.
- An interface is not extended by a class; it is implemented by a class.
- An interface can extend multiple interfaces.