Interfaces

By Rahul Barve

Objectives

- Introduction to Interfaces
- Why Interfaces
- Interface Example
- Understand Interface Rules.
- Abstract Class Vs. Interface
- New Features

Interface

By Rahul Barve

Interface

- Interface is a collection of `abstract` methods and possibly `final` variables.
- Used to declare methods where implementation is not available.

Interface

- An interface is declared using a keyword `interface`.
- E.g.

```
public interface MyInterface{
    int myVar = 100;
    void myMethod();
}
```

Interface

- Once an interface is created, it can be further used by creating an implementation class for that interface.

Interface

- E.g.

```
public class MyClass
                implements MyInterface {
    public void myMethod(){
        System.out.println(myVar);
    }
}
```

Why Interface

By Rahul Barve

Why Interface

- An interface is used to expand the scope of polymorphism.
- Achieving multiple inheritance in the context of methods.
- Used for loose coupling.

Interface Rules

- Possible associations:
  - `Class extends Class`
  - `Class implements Interface(s)`
  - `Interface extends Interface(s)`

Interface Rules

- Methods of interface are by default `public` and `abstract`
- Variables of interface are by default `public`, `static` and `final` .
- A class that implements interface, must implement all the methods of that interface; otherwise must be declared `abstract`.

Interface Rules

- An object of a class is always compatible with the interface type.
- An interface type is always compatible with `Object`.

Abstract Class Vs. Interface

- A class can extend only one abstract class.
- Useful to achieve polymorphism when classes are co-related.
- Can contain concrete methods also.

- A class can implement any no of interfaces.
- Useful to achieve polymorphism when classes are not co-related.
- Generally contains only abstract methods.

New Features

By Rahul Barve

New Features

- Since JDK 1.8, it is possible to define methods within an interface provided they are declared as either default or static.
- This feature enables to add a new functionality in the interfaces without breaking the existing contract of the implementing classes.

Default and Static Methods

- E.g.

```
public interface MyInterface {
 default void m1(){
  //Some Code
 }
  static void m2(){

 }
}
```

Lets Summarize

- What are Interfaces
- Why Interfaces
- Interface Examples
- Interface Rules.
- Abstract Class Vs. Interface
- New Features