

Interfaces in Java

Interfaces in Java

Interfaces in Java – Interfaces have similar structure as that of a [class](#), but with abstract methods and variables that are static & final . Which means, the variables of interfaces are constants and methods do not have implementations. But an Interface in Java enforces definite implementations of methods/behaviors/routines in classes that implements the interface.

If a class is implementing an interface, this means that class is promising that the abstract methods in the interface shall be implemented. Compiler makes sure that this promise is not broken. If this promise is broken, in cases like any of the methods is not implemented, compiler throws an error during compilation and stops.

For an example from Java language itself, FileInputStream class implements Closeable interface. And FileInputStream implements the method close(), which is enforced by Closeable interface.

Example 1 – Java Interface

In the following program, InterfaceDemo.java, we shall look create an interface called Car, and a class called Jaguar. The Car interface has methods accelerate and break. Jaguar implements Car interface and so Jaguar should implement the methods accelerate and break enforced by the Car interface.

Car.java

```
public interface Car {  
    public void accelerate();  
    public void breaking();  
}
```

Jaguar.java

```
public class Jaguar implements Car {  
  
    String name;  
  
    public Jaguar(String name){  
        this.name = name;  
    }  
  
    @Override  
    public void accelerate() {  
        System.out.println(name + " is accelerating.");  
    }  
}
```

```

    }

    @Override
    public void breaking() {
        System.out.println(name + " is breaking.");
    }
}

```

InterfaceDemo.java

```

/**
 * Example program to show the working of an interface in Java
 * @author tutorialkart.com
 *
 */
public class InterfaceDemo {

    public static void main(String[] args) {
        System.out.println("---Demo on Interfaces in Java---");
        Jaguar myNewJaguar = new Jaguar("XJ");
        myNewJaguar.accelerate();
        myNewJaguar.breaking();
    }

}

```

Output

```

---Demo on Interfaces in Java---
XJ is accelerating.
XJ is breaking.

```

Conclusion

In this [Java Tutorial](#), we have learnt about Interfaces in Java that realize [Abstraction in Java](#). In our next tutorial, we shall learn about [Abstract Classes and Methods](#).

Java Tutorial

- ◆ [Java Tutorial](#)
- ◆ [Java Introduction](#)
- ◆ [Java Installation](#)
- ◆ [IDEs for Java Development](#)
- ◆ [Java HelloWorld Program](#)
- ◆ [Java Program Structure](#)

◆ Java Datatypes
◆ Java Variable Types
◆ Java Access Modifiers
◆ Java Operators
◆ Java Decision Making
◆ Java Loops
◆ Java Array
◆ Java OOPs
◆ Java String
◆ Java Exception Handling
◆ Java File Operations
◆ Java Date & Time
◆ Java MySQL
◆ Java Random
◆ Java Math