TutorialKart

Method Overriding in Java

Method Overriding in Java

Method Overriding is a way to realize Polymorphism in Java.

In this tutorial, we shall learn Overriding in Java with Example Programs, where methods of Super Class are overridden by methods of Sub Class.

In context of sub-class extending a super-class, the sub-class can access super-class's methods. If in case the sub-class wants functionality of an existing method in Super Class to be replaced by some specific implementation, then Java provides the ability to override the existing functionality through Method Overriding. It can also be thought of as **Runtime Polymorphism**.

Rules for Method Overriding

Following are the rules for implementing Method Overriding in Java.

1. The method declaration should be same as that of the method that is to be overridden.
2. The class (sub class) should extend another class (super class), prior to even try overriding.
3. Sub Class can never override final methods of Super Class.

Example 1 – Method Overriding

We shall take an example scenario where Car is a super-class and Audi is a sub-class.

Following is the super class.

Car.java

```
public class Car {

    public void brake(){
        System.out.println("break in Car");
    }

    public void accelerate(){
        System.out.println("accelerate in Car");
    }
}
```

Following is the sub class.

Audi.java

```
public class Audi extends Car {

    public static void main(String[] args) {
        Audi audi = new Audi();
        audi.accelerate();
        audi.brake();
    }
}
```

Run the above program. You shall get the following output in console.

```
accelerate in Car
break in Car
```

Sub Class Overriding Method of Super Class

Now we shall override accelerate method in Audi.java by overriding accelerate() in Car.java

Car.java

```
public class Car {

    public void brake(){
        System.out.println("break in Car");
    }

    public void accelerate(){
        System.out.println("accelerate in Car");
    }
}
```

Audi.java

```
public class Audi extends Car {

    public static void main(String[] args) {
        Audi audi = new Audi();
        audi.accelerate();
        audi.brake();
    }

    public void accelerate(){
        System.out.println("accelerate in Audi");
    }
}
```

Output

```
accelerate in Audi
break in Car
```

Override final method of Super Class

Any method that is declared final in the Super Class cannot be overridden by Sub Class. If you try doing so, you may get an Error as shown in the following.

Car.java

```
public class Car {

    public void brake(){
        System.out.println("break in Car");
    }

    public final void accelerate(){
        System.out.println("accelerate in Car");
    }
}
```

Audi.java

```
public class Audi extends Car {

    public static void main(String[] args) {
        Audi audi = new Audi();
        audi.accelerate();
        audi.brake();
    }

    public void accelerate(){
        System.out.println("accelerate in Audi");
    }
}
```

Output

```
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.VerifyError: class Audi overrides final method acce
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:763)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)
```

Conclusion

In this Java Tutorial, we have learnt Method Overriding in Java with example programs, the pre-requisites to override a method.

In our next tutorial, we shall learn about Final Keyword in Java.

Java Tutorial