Exception Handling with Method Overriding in Java

Prerequisite: Exception and Exception Handling in Java, Overriding in Java, Checked and Unchecked Exception.

An Exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run-time, that disrupts the normal flow of the program's instructions. **Exception handling** are used to handle runtime error. It helps to maintain the normal flow of the program.

In any object-oriented programming language, Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to override the method in the super-class.

Exception Handling with Method Overriding

When Exception handling is involved with Method overriding, ambiguity occurs. The compiler gets confused as which definition is to be followed. Such problems were of two types:

- **Problem 1: If The SuperClass doesn't declare an exception:**
  In this problem, two cases arise:

- **Case 1: If SuperClass doesn't declare any exception and subclass declare checked exception**

  Example:

```
import java.io.*;

class SuperClass {

    // SuperClass doesn't declare any exception
    void method()
    {
        System.out.println("SuperClass");
    }
}

// SuperClass inherited by the SubClass
class SubClass extends SuperClass {

    // method() declaring Checked Exception IOException
    void method() throws IOException
    {

        // IOException is of type Checked Exception
        // so the compiler will give Error

        System.out.println("SubClass");
    }

    // Driver code
    public static void main(String args[])
    {
        SuperClass s = new SubClass();
        s.method();
    }
}
```

  Compile Errors:

```
prog.java:16: error:
method() in SubClass cannot override method() in SuperClass
    void method() throws IOException
         ^
  overridden method does not throw IOException
1 error
```

- **Case 2: If SuperClass doesn't declare any exception and SubClass declare Unchecked exception**

  Example:

```
import java.io.*;
```

```
class SuperClass {

    // SuperClass doesn't declare any exception
    void method()
    {
        System.out.println("SuperClass");
    }
}

// SuperClass inherited by the SubClass
class SubClass extends SuperClass {

    // method() declaring Unchecked Exception ArithmeticException
    void method() throws ArithmeticException
    {

        // ArithmeticException is of type Unchecked Exception
        // so the compiler won't give any error

        System.out.println("SubClass");
    }

    // Driver code
    public static void main(String args[])
    {
        SuperClass s = new SubClass();
        s.method();
    }
}
```

Output:

```
SubClass
```

- **Problem 2: If The SuperClass declares an exception:**
  In this problem also, two cases arise:
  - **Case 1: If SuperClass declares an exception and SubClass declares exceptions other than the child exception of the SuperClass declared Exception**
    Example:

```
import java.io.*;

class SuperClass {

    // SuperClass declares an exception
    void method() throws RuntimeException
    {
        System.out.println("SuperClass");
    }
}
```

```
// SuperClass inherited by the SubClass
class SubClass extends SuperClass {

    // SubClass declaring an exception
    // which are not a child exception of RuntimeException
    void method() throws Exception
    {

        // Exception is not a child exception
        // of the RuntimeException
        // So the compiler will give an error

        System.out.println("SubClass");
    }

    // Driver code
    public static void main(String args[])
    {
        SuperClass s = new SubClass();
        s.method();
    }
}
```

Compile Errors:

```
prog.java:16: error:
method() in SubClass cannot override method() in SuperClass
    void method() throws Exception
         ^
  overridden method does not throw Exception
1 error
```

- **Case 2: If SuperClass declares an exception and SubClass declares an child exception of the SuperClass declared Exception.**
  Example:

```
import java.io.*;

class SuperClass {

    // SuperClass declares an exception
    void method() throws RuntimeException
    {
        System.out.println("SuperClass");
    }
}

// SuperClass inherited by the SubClass
class SubClass extends SuperClass {

    // SubClass declaring a child exception
    // of RuntimeException
```

```
    void method() throws ArithmeticException
    {

        // ArithmeticException is a child exception
        // of the RuntimeException
        // So the compiler won't give an error
        System.out.println("SubClass");
    }

    // Driver code
    public static void main(String args[])
    {
        SuperClass s = new SubClass();
        s.method();
    }
}
```

Output:

```
SubClass
```

Conclusion for Handling such Exceptions: Hence, following conclusions can be derived from the above examples:

1. If SuperClass does not declare an exception, then the SubClass can only declare unchecked exceptions, but not the checked exceptions.
2. If SuperClass declares an exception, then the SubClass can only declare the child exceptions of the exception declared by the SuperClass, but not any other exception.

Recommended Posts:

Java | Exception Handling | Question 8

Java | Exception Handling | Question 6

Java | Exception Handling | Question 7

Java | Exception Handling | Question 8

Java | Exception Handling | Question 1

Comparison of Exception Handling in C++ and Java

Java | Exception Handling | Question 4

Java | Exception Handling | Question 3

Java | Exception Handling | Question 2