# JAVA 7新特性——在单个catch代码块中捕获多个异常，以及用升级版的类型检查重新抛出异常

在Java 7中，catch代码块得到了升级，用以在单个catch块中处理多个异常。如果你要捕获多个异常并且它们包含相似的代码，使用这一特性将会减少代码重复度。下面用一个例子来理解。

Java 7之前的版本：

*catch (IOException ex) {*

*logger.error(ex);*

*throw new MyException(ex.getMessage());*

*catch (SQLException ex) {*

*logger.error(ex);*

*throw new MyException(ex.getMessage());*

*}catch (Exception ex) {*

*logger.error(ex);*

*throw new MyException(ex.getMessage());*

*}*

在Java 7中，我们可以用一个catch块捕获所有这些异常：

*catch(IOException | SQLException | Exception ex){*

*logger.error(ex);*

*throw new MyException(ex.getMessage());*

*}*

如果用一个catch块处理多个异常，可以用管道符（|）将它们分开，在这种情况下异常参数变量（ex）是定义为final的，所以不能被修改。这一特性将生成更少的字节码并减少代码冗余。

另一个升级~~完成于~~是编译器对重新抛出异常（rethrown exceptions）的处理。这一特性允许在一个方法声明的throws从句中指定更多特定的异常类型。

让我们来看看一个小例子：

*package com.journaldev.util;*

*public class Java7MultipleExceptions {*

*public static void main(String[] args) {*

*try{*

*rethrow("abc");*

*}catch(FirstException | SecondException | ThirdException e){*

*//below assignment will throw compile time exception since e is final*

*//****以下赋值将会在编译期抛出异常，因为e是final型的***

*//e = new Exception();*

*System.out.println(e.getMessage());*

*}*

*}*

*static void rethrow(String s) throws FirstException, SecondException,*

*ThirdException {*

*try {*

*if (s.equals("First"))*

*throw new FirstException("First");*

*else if (s.equals("Second"))*

*throw new SecondException("Second");*

*else*

*throw new ThirdException("Third");*

*} catch (Exception e) {*

*//below assignment disables the improved rethrow exception type checking feature of Java 7*

***~~//以下赋值将使会使Java 7升级的重新抛出异常的类型检查功能不被启用~~***

***//下面的赋值没有启用重新抛出异常的类型检查功能，这是Java 7的新特性***

*// e=new ThirdException();*

*throw e;*

*}*

*}*

*static class FirstException extends Exception {*

*public FirstException(String msg) {*

*super(msg);*

*}*

*}*

*static class SecondException extends Exception {*

*public SecondException(String msg) {*

*super(msg);*

*}*

*}*

*static class ThirdException extends Exception {*

*public ThirdException(String msg) {*

*super(msg);*

*}*

*}*

*}*

如你所见在rethrow方法中，catch块捕获的异常并没有出现在throws从句中。Java 7编译器会分析完整的try代码块以检查从catch块中什么类型的异常被抛出和重新抛出。

需要注意一旦改变了catch块的参数，编译器的分析将不会被启用。