

异常处理允许我们以受控的方式处理异常情况，而不是直接退出程序，例如程序定义的错误。

发生异常时可以抛出异常。术语“异常”表示当前的程序执行停止，并且被重定向到最近的适用的catch子句。如果没有适用的catch子句存在，则程序的执行结束。

JDBC异常处理与Java异常处理非常相似，但对于JDBC，要处理的最常见异常是java.sql.SQLException。

SQLException方法

驱动程序和数据库中都会发生SQLException。发生这种异常时，SQLException类型的对象将被传递给catch子句。

传递的SQLException对象具有以下可用于检索有关异常信息的方法 -

方法	描述
getErrorCode()	获取与异常关联的错误代码。
getMessage()	获取驱动程序处理的错误的JDBC驱动程序的错误消息，或获取数据库错误的Oracle错误代码和消息。
getSQLState()	获取XOPEN SQLstate字符串。对于JDBC驱动程序错误，不会从此方法返回有用的信息。对于数据库错误，返回五位数的XOPEN SQLstate代码。此方法可以返回null。
getNextException()	获取异常链中的下一个Exception对象。
printStackTrace()	打印当前异常或可抛出的异常，并将其追溯到标准错误流。
printStackTrace(PrintStream s)	将此throwable及其回溯打印到指定的打印流。
printStackTrace(PrintWriter w)	打印这个throwable，它是回溯到指定的打印机(PrintWriter)。

通过利用Exception对象提供的信息，可以捕获异常并适当地继续执行程序。下面是一个try块的一般形式 -

```
try {
    // Your risky code goes between these curly braces!!!
}
catch(Exception ex) {
    // Your exception handling code goes between these
    // curly braces, similar to the exception clause
    // in a PL/SQL block.
}
finally {
    // Your must-always-be-executed code goes between these
    // curly braces. Like closing database connection.
}
```