

## 本章小结

1. 异常是导致程序中断运行的一种指令流，当异常发生时，如果没有进行良好的处理，则程序将会中断执行。
2. 异常处理可以使用 try...catch 进行处理，也可以使用 try...catch...finally 进行处理，在 try 语句中捕捉异常，之后在 catch 中处理异常，finally 作为异常的统一出口，不管是否发生异常都要执行此段代码。
3. 异常的最大父类是 Throwable，其分为两个子类：Exception、Error。Exception 表示程序处理的异常；而 Error 表示 JVM 错误，一般不是由程序开发人员处理的。
4. 发生异常之后，JVM 会自动产生一个异常类的实例化对象，并匹配相应的 catch 语句中的异常类型，也可以利用对象的向上转型关系，直接捕获 Exception。
5. throws 用在方法声明处，表示本方法不处理异常。
6. throw 表示在方法中手工抛出一个异常。
7. 自定义异常类时，只需要继承 Exception 类或 RuntimeException 类。
8. 断言是 JDK 1.4 之后提供的新功能，可以用来检测程序的执行结果，但开发中并不提倡使用断言进行检测。

## 课后习题

### 一、填空题

1. Throwable 下的两个子类是\_\_\_\_\_、\_\_\_\_\_。
2. ArithmeticException 类表示\_\_\_\_\_异常，ArrayIndexOutOfBoundsException 表示\_\_\_\_\_异常。
3. 一个 try 代码后面必须跟着若干个\_\_\_\_\_代码段或者一个\_\_\_\_\_代码段。
4. 如果一个方法使用了\_\_\_\_\_，则编译器会强制在使用此方法时进行异常的处理。
5. 异常处理中使用\_\_\_\_\_作为异常的统一出口。

### 二、选择题

1. 使用 ( ) 关键字可以在程序中手工抛出异常。  
A. throws      B. throw      C. assert      D. class
2. 下面 ( ) 关键字可以用在方法的声明处。  
A. throws      B. assert      C. class      D. interface
3. 为了捕获一个异常，代码必须放在下面 ( ) 中。  
A. try 块      B. catch 块      C. throws 块      D. finally 块
4. 下面关于 try 块的说法正确的是 ( )。  
A. try 块后至少应有一个 catch 块      B. try 块后必须有 finally 块  
C. 可能抛出异常的方法应放在 try 块中      D. 对抛出的异常的处理应放在 try 块中
5. finally 块中的代码将 ( )。  
A. 总是被执行

- B. 如果 try 块后面没有 catch 块时, finally 块中的代码才会执行
  - C. 异常发生时才被执行
  - D. 异常没有发生时才执行
6. 一个异常将终止 ( )。
- A. 整个程序
  - B. 只终止抛出异常的方法
  - C. 产生异常的 try 块
  - D. 上面的说法都不对
7. 所有异常的共同父类是 ( )。
- A. Error
  - B. Exception
  - C. Throwable
  - D. RuntimeException

### 三、判断题

- 1. 一个 try 语句后有多多个 catch 时, 捕获范围大的异常要放在捕获范围小的异常之后。 ( )
- 2. finally 语句可以根据需要地有选择地添加。 ( )

### 四、简答题

- 1. 简述 RuntimeException 和 Exception 的区别。
- 2. try、catch、finally 三种语句的功能是什么?
- 3. 简述 Java 中的异常处理机制。
- 4. 简述 Error 和 Exception 的区别。
- 5. 列举三个常见的 RuntimeException 子类。

### 五、编程题

编写应用程序, 从命令行输入两个小数参数, 求它们的商。要求程序中捕获 NumberFormatException 异常和 ArithmeticException 异常。