面向对象程序设计Java实验指导书

实验四 异常处理

# 实验目的

* 掌握Java异常处理机制
* 理解Java异常类并掌握异常类的使用
* 了解自定义异常类及使用

# 实验要求

* 按照下述要求编写Java代码
* 按要求回答相关问题
* 按时完成并上交实验报告

# 实验内容

|  |  |
| --- | --- |
|  | 在Eclipse中创建一个Java项目，项目命名方式为：学号+实验编号，例如：学号为2014121001的同学的实验三项目命名为：2014121001exp4 |

## 运行时异常类

在你的项目中定义一个package，命名为：exceptions；在exceptions的包中定义一个类UncheckedExceptionExp，代码如下：

|  |  |
| --- | --- |
|  | **import** java.util.Scanner;  **public** **class** UncheckedExceptionExp {  **private** **static** **int** *op1* = 325;//被除数  **public** **static** **void** main(String[] args) {  System.*out*.println("请输入除数：");  Scanner scanner = **new** Scanner(System.*in*);  **int** op2 = scanner.nextInt();  System.*out*.println("两数相除的结果是："+*op1*/op2);  }  } |

* 运行上述代码，并分别输入一个非零数和零作为除数，验证程序会分别输出什么样的结果，当除数为零时会产生什么类型的异常对象。
* 修改UncheckedExceptionExp类，当除数为零时，不显示异常调用栈的信息，而输出“输入有误，除数不能为零”的信息。
* 为什么在UncheckedExceptionExp类中没有进行异常处理程序仍然能够通过编译。

## 非运行时异常的处理

* 下面给出了一个程序主要流程代码，但未处理非运行时异常，要求加入异常处理代码，让程序能正常运行，具体的要求可看代码的注释；

|  |  |
| --- | --- |
|  | **import** java.io.File;  **import** java.io.FileReader;  /\*\*  \* FileReaderDemo类实现了从文件（d:\\data.txt）中读取字符，并输出到标准的  \* Console;下面的代码中按照正常的流程完成了代码，但没有进行异常处理，所以程序无\* 法编译，要求采用 try...catch 的方式进行异常处理，保证程序能正常实现功能。异\* 常处理还必须满足下面的要求。  \*  \*  \*/  **public** **class** FileReaderDemo {  **public** **static** **void** main(String[] args) {  //创建要读取的数据文件的文件对象 file  File file = **new** File("d:\\data.txt");  //创建要读取数据的输入流  FileReader fileReader = **new** FileReader(file);  //从输入流中读取一个字符；  **int** c = fileReader.read();  //判断是否读取到文件结束，如果读取到文件的最后会返回-1  **while** (c!=-1) {  //输出读取的这个字符  System.***out***.println((**char**)c);  //再读取下一个字符  c = fileReader.read();  }  //所有读取完毕后，关闭fileReader对象，要求不管前面是否抛出异常，  //都必须要执行关闭操作。  fileReader.close();  }  } |

## 上述例子中用到的类与相关方法的API 文档说明。

### File

**public File(**[**String**](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\lang\String.html)**pathname)**

通过将给定路径名字符串转换为抽象路径名来创建一个新 File 实例。如果给定字符串是空字符串，那么结果是空抽象路径名。

**参数：**

pathname - 路径名字符串

**抛出：**

[NullPointerException](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\lang\NullPointerException.html) - 如果 pathname 参数为 null

### FileReader

**public FileReader(**[**File**](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\File.html)**file)**

**throws** [**FileNotFoundException**](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\FileNotFoundException.html)

在给定从中读取数据的 File 的情况下创建一个新 FileReader。

**参数：**

file - 要从中读取数据的 File

**抛出：**

[FileNotFoundException](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\FileNotFoundException.html) - 如果文件不存在，或者它是一个目录，而不是一个常规文件，抑或因为其他某些原因而无法打开进行读取。

**public int read()**

**throws** [**IOException**](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\IOException.html)

读取单个字符。在字符可用、发生 I/O 错误或者已到达流的末尾前，此方法一直阻塞。

用于支持高效的单字符输入的子类应重写此方法。

**返回：**

作为整数读取的字符，范围在 0 到 65535 之间 (0x00-0xffff)，如果已到达流的末尾，则返回 -1

**抛出：**

[IOException](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\IOException.html) - 如果发生 I/O 错误

**public void close()**

**throws** [**IOException**](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\IOException.html)

关闭该流并释放与之关联的所有资源。在关闭该流后，再调用 read()、ready()、mark()、reset() 或 skip() 将抛出 IOException。关闭以前关闭的流无效。

**指定者：**

接口 [Closeable](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\Closeable.html) 中的 [close](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\Closeable.html#close())

**指定者：**

类 [Reader](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\Reader.html) 中的 [close](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\Reader.html#close())

**抛出：**

[IOException](file:///D:\2015-2016-1%20%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%20Java\API%20Documentation\html_zh_CN\html\zh_CN\api\java\io\IOException.html) - 如果发生 I/O 错误