异常(Exception)

课程目标

* 熟悉使用try-catch-finally处理异常
* 会使用throw、throws抛出异常
* 掌握异常及其分类
* 了解自定义异常
* 使用log4j记录日志

1. 异常
2. 现实生活中的异常:

现实生活中发生的意外情况

1. 什么是程序中的异常(Exception)

当程序运行的过程中发生的例外情况，则称为异常.它会导致程序中断运行.

1. 异常处理
2. 什么是异常处理

Java编程语言使用异常处理机制为程序提供了错误处理的能力.

注意:异常处理的关键字:try、catch、finally、throw、throws

1. 异常处理作用:增强程序的稳定性、健壮性.
2. 异常处理语句

try{

可能出错的代码块

}catch(异常类型 对象名){

出错时执行的代码块…

}finally{

有错没错都执行的代码. 一般释放资源操作

}

注意:

1. finally块可有可无
2. 多重catch

|  |
| --- |
| **catch** (ArithmeticException e){ //除数为0  e.printStackTrace(); } **catch** (InputMismatchException e){ //输入不匹配  e.printStackTrace(); }**catch** (Exception e){ //接收所有异常,加在最后  e.printStackTrace(); } |

1. 获取异常的信息

*4.1.获取异常的信息  
异常对象.getMessage()  
System.out.println(e.getMessage());*

*4.2.显示异常的堆栈信息  
异常对象*.printStackTrace();

1. (理解)异常的类型

|  |  |
| --- | --- |
| **异 常 类 型** | **说 明** |
| **Exception** | **异常层次结构的父类** |
| **ArithmeticException** | **算术错误情形，如以零作除数** |
| **ArrayIndexOutOfBoundsException** | **数组下标越界** |
| **NullPointerException** | **尝试访问 null 对象成员** |
| **ClassNotFoundException** | **不能加载所需的类** |
| **IllegalArgumentException** | **方法接收到非法参数** |
| **ClassCastException** | **对象强制类型转换出错** |
| **NumberFormatException** | **数字格式转换异常，如把"abc"转换成数字** |

1. 使用throws声明方法抛出的异常,交给调用者处理

|  |
| --- |
| **public class** MathTools {  *//求商的方法* **public void** calShang() **throws** Exception{  Scanner input=**new** Scanner(System.***in***);  **int** res=0;  System.***out***.println(**"请输入被除数:"**);  **int** one=input.nextInt();  System.***out***.println(**"请输入除数:"**);  **int** two=input.nextInt();  *//求商* res=one/two; System.***out***.println(**"结果是:"**+res);  }  **public static void** main(String[] args){  *//调用方法* MathTools math=**new** MathTools();  **try** {  math.calShang(); *//调用者处理异常* } **catch** (Exception e) {  e.printStackTrace();  }  System.***out***.println(**"继续"**);  } } |

1. 使用throw关键字手动抛出异常

|  |
| --- |
| **public static void** main(String[] args) {  *//求商* Scanner input=**new** Scanner(System.***in***);  **int** res=0;  **try** {  System.***out***.println(**"请输入被除数:"**);  **int** one=input.nextInt();  System.***out***.println(**"请输入除数:"**);  **int** two=input.nextInt();  **if**(two==5){  *//throw new Exception("除数不能为5 /by five"); //手动抛* **throw new** FiveException();  }  *//求商* res=one/two;  }**catch** (FiveException e){  System.***out***.println(e.getMessage());  }**finally** {  System.***out***.println(**"有错没错都执行....."**);  }  *//输出结果* System.***out***.println(**"结果是:"**+res); } |

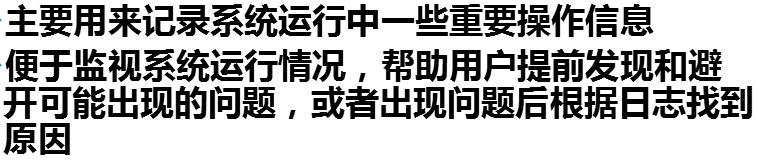
1. 自定义异常类型(了解)

定义一个类去继承Exception即可

|  |
| --- |
| **public class** FiveException **extends** Exception {  **public** FiveException(){  **super**(**"除数不能为5"**);  }  **public** FiveException(String message){  **super**(message);  } } |

扩展:使用Log4j工具记录日志

1. 什么是日志(Log)



1. 使用Log4j记录日志信息的步骤

2.1 在项目中添加log4j的jar包(log4j-1.2.17.jar)

在项目中创建lib文件夹，并将包复制到lib文件夹中. 在lib文件夹中找到对应的jar包右键--->add as libaray(添加到库)-->确定即可

2.2 在项目中添加log4j.properties属性文件，并存放在src目录中

log4j.properties文件作用就是配置log4j如何工作

2.3 在类中创建Logger对象进行日志信息的记录

|  |
| --- |
| **import** org.apache.log4j.Logger;  **public class** Demo4 {  *//1.创建日志记录对象* **static** Logger *logger*=Logger.*getLogger*(Demo4.**class**);  **public static void** main(String[] args) {  *//求商* Scanner input=**new** Scanner(System.***in***);  **int** res=0;  **try**{  *//可能出错的代码* System.***out***.println(**"请输入的被除数:"**);  **int** one=input.nextInt();  System.***out***.println(**"请输入的除数:"**);  **int** two=input.nextInt(); *// new Exception();  //求商* res=one/two; *//出错以后程序中断运行* }**catch** (Exception ex){  *// System.out.println(ex.getMessage());  //2.使用log4j记录异常信息  //logger.info(ex.getMessage()); //记录常规信息，如关键步骤  //logger.debug("错误信息"); //理解为异常  logger*.error(**"错误信息"**); *//表示严重错误  //logger.warn("错误信息"); //表示警告* }  *//输出结果* System.***out***.println(**"商是:"**+res);  } } |