

零基础学 Java









初识异常和 try catch

初识异常和 try catch



● 看例程:初识异常

● 看例程:初识 try catch 语句

● 异常:在程序出错误的时候, Java 支持使用异常,将错误信息封装起来, 并让程序跳出正常的处理流程,交给异常处理部分去处理





Java 中异常的分类

- 按照异常的继承关系分类
- 按照处理方式不同分类

按照异常的继承关系分类



- 异常也是 Java 中的类
- 所有异常的父类: Throwable
- 两类异常: Error 和 Exception
- 看一下 Throwable 的继承关系

按照处理方式不同分类



- checked exception: 语法要求必须要用 try catch 或者 throws 语句处理的异常
- unchecked exception: 语法不要求一定要用 try catch 或者 throws 语句处理的异常
- Error 和 RuntimeException 是 unchecked exception 的父类。我们一般 使用 RuntimeException
- 看例程:什么叫做语法要求必须处理的异常





抛出异常的语法

抛出异常的语法



● 看例程:抛出别的代码抛过来的异常

● 看例程:在代码里自己创建一个异常,并抛出去

● 看例程:抽象方法中声明抛出的异常是接口方法签名的一部分





Java 异常的传递

Java 异常的传递



- 看例程:方法一个调一个的层层调用,组成了方法调用栈
- 看例程:学习 Java 异常在方法调用之间传递。
- Java 异常的的归宿:要么沿着方法调栈顺序一路抛,最终造成当前线程出错退出,要么被 catch 住
- 线程?线程的英文叫做 Thread 。 我们可以简单的认为是帮我们执行 代码的那个东西。迄今为止我们学习的都是单线程的程序。后面我们会 学习多线程的内容





自定义异常

自定义异常



● 异常最重要的信息:类型,错误信息和出错时的调用栈

● 看例程:创建自己的异常类





异常传递不是凌波微步

异常传递不是凌波微步



- 异常代表程序出错,不要使用异常来做正常的程序跳转
- 错误的示例:使用"凌波微步"跳转到 catch 语句
- 异常不是凌波微步,异常的创建和处理很耗费资源





try catch finally 语句

try catch finally 语句

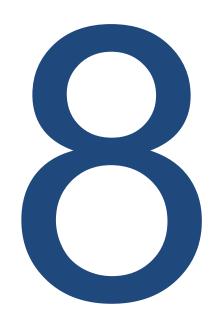


● 看例程: 学习 try catch finally 语句

● 看例程: 学习 try finally 语句

● 看例程:学习 catch 多种异常





自动回收资源的 try 语句

自动回收资源的 try 语句



● 和资源相关的异常处理比较繁复,尤其是有多个资源的时候

● 看例程:如何让资源回收的方法自动被调用





Java 中的常见异常

Java 中的常见异常



● 常见异常

- NullPointerException
- IndexOutOfBoundsException
- ClassCastException
- ClassNotFoundException
- IOException



