JavaEE项目中异常处理计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/645/2021_2022_JavaEE_E9_A1_B9_E7_c97_645508.htm 我们在定义一个新的异常类时，必须提供这样一个可以包含嵌套异常的构造函数。并有一个私有成员来保存这个“ 起因异常 ”。java 代码 public Class ExceptionB extends Exception{ private Throwable cause. public ExceptionB(String msg, Throwable ex){ super(msg). this.cause = ex. } public ExceptionB(String msg){ super(msg). } public ExceptionB(Throwable ex){ this.cause = ex. } } 当然，我们在调用printStackTrace方法时，需要把所有的“ 起因异常 ”的信息也同时打印出来。所以我们需要覆写printStackTrace方法来显示全部的异常栈跟踪。包括嵌套异常的栈跟踪。java 代码 public void printStackTrace(PrintStrean ps){ if(cause == null){ super.printStackTrace(ps). }else{ ps.println(this). cause.printStackTrace(ps). } } 一个完整的支持嵌套的checked异常类源码如下。我们在这里暂且把它叫做NestedExceptionjava 代码 public NestedException extends Exception{ private Throwable cause. public NestedException (String msg){ super(msg). } public NestedException(String msg, Throwable ex){ super(msg). This.cause = ex. } public Throwable getCause(){ return (this.cause == null ? this :this.cause). } public getMessage(){ String message = super.getMessage(). Throwable cause = getCause(). if(cause != null){ message = message “ .nested Exception is ” cause. } return message. } 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com