

通过实例学习Java对象的构造过程 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E9_80_9A_E8_BF_87_E5_AE_9E_E4_c104_145213.htm 本文提供一个项目中的错误实例，提供对其观察和分析，揭示出Java语言实例化一个对象具体过程，最后总结出设计Java类的一个重要规则。通过阅读本文，可以使Java程序员理解Java对象的构造过程，从而设计出更加健壮的代码。本文适合Java初学者和需要提高的Java程序员阅读。程序抛出了一个异常 作者曾经在一个项目里面向项目组成员提供了一个抽象的对话框基类，使用者只需在子类中实现基类的一个抽象方法来画出显示数据的界面，就可使项目内的对话框具有相同的风格。具体的代码实现片断如下（为了简洁起见，省略了其他无关的代码）

```
: public abstract class BaseDlg extends JDialog {public
BaseDlg(Frame frame, String title) {super(frame, title,
true).this.getContentPane().setLayout(new
BorderLayout()).this.getContentPane().add(createHeadPanel(),
BorderLayout.NORTH).this.getContentPane().add(createClientPa
nel(),
BorderLayout.CENTER).this.getContentPane().add(createButtonP
anel(), BorderLayout.SOUTH).}private JPanel createHeadPanel()
{... // 创建对话框头部} // 创建对话框客户区域，交给子类实
现protected abstract JPanel createClientPanel().private JPanel
createButtonPanel {... // 创建按钮区域} 这个类在有的代码中工
作得很好，但一个同事在使用时，程序却抛出了一个NullPointerException违例！经过比较，找出了工作正常和不
```

正常的程序的细微差别，代码片断分别如下：一、正常工作的代码：

```
public class ChildDlg1 extends BaseDlg { JTextField jTextFieldName. public ChildDlg1() { super(null, "Title"). } public JPanel createClientPanel() { jTextFieldName = new JTextField(). JPanel panel = new JPanel(new FlowLayout()). panel.add(jTextFieldName). ... // 其它代码 return panel. } ...}ChildDlg1 dlg = new ChildDlg1(frame, "Title"). // 外部的调用
```

二、工作不正常的代码：

```
public class ChildDlg2 extends BaseDlg { JTextField jTextFieldName = new JTextField(). public ChildDlg2() { super(null, "Title"). } public JPanel createClientPanel() { JPanel panel = new JPanel(new FlowLayout()). panel.add(jTextFieldName). ... // 其它代码 return panel. } ...}ChildDlg2 dlg = new ChildDlg2(). // 外部的调用
```

你看得出来两段代码之间的差别了吗？对了，两者的差别仅仅在于类变量jTextFieldName的初始化时间。经过跟踪，发现在执行panel.add(jTextFieldName)语句之时，jTextFieldName确实是空值。我们知道，Java允许在定义类变量的同时给变量赋初始值。系统运行过程中需要创建一个对象的时候，首先会为对象分配内存空间，然后在“先于调用任何方法之前”根据变量在类内的定义顺序来初始化变量，接着再调用类的构造方法。那么，在本例中，为什么在变量定义时便初始化的代码反而会出现空指针违例呢？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