

Java入门实例classpath及package详解 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/240/2021_2022_Java_E5_85_A5_E9_97_A8_c104_240585.htm Java很诱人，但对于刚跨入Java门槛的初学者来说，编译并运行一个无比简单的Java程序简直就是一个恶梦。明明程序没错，但各种各样让人摸不着头脑的错误信息真的让你百思不得其解，许多在Java门口徘徊了很久的初学者就这样放弃了学习Java的机会，很是可惜。笔者也经历过这个无比痛苦的阶段，感觉到编译难的问题就出在classpath的设置及对package的理解之上。本文以实例的方式，逐一解决在编译过程中所出现的各种classpath的设置问题。本文实例运行的环境是在Windows XP JDK 1.5.0。对其他的环境，读者应该很容易进行相应的转换。

1. 下载并安装JDK1.5.0，并按默认路径，安装到C:\Program Files\Java\jdk1.5.0中。
2. 用鼠标单击WindowsXP的“开始”->“运行”，在弹出的运行窗口中输入cmd，按确定或回车，打开一个命令行窗口。
3. 在命令行中输入：java有一列长长的洋文滚了出来，这是JDK告诉我们java这个命令的使用方法。其中隐含了一个重要信息，即JDK安装成功，可以在命令行中使用java此命令了。
4. 在命令行中输入javac屏幕显示：javac不是内部或外部命令，也不是可运行的程序或批处理文件。这是由于windows找不到javac这个命令的原因。这就不明白了，java与javac都是JDK在同一个子目录里面的两个文件，为什么可以直接运行java而不能直接运行javac呢？原来，Sun公司为了方便大家在安装完JDK后马上就可以运行Java类文件，在后台悄悄地将java命令加入了Path的搜索路径中，因此我们可以直接运行java命令(

但我们是看不到它到底是在哪设置的，无论是在用户的Path或系统的Path设置中均找不到这个java存放的路径)。但Sun所做的到此为止，其他JDK的命令，一概不管，需要由用户自己添加到搜索路径中。

5. 既然如此，那我们自己添加Path的搜索路径吧。对“我的电脑”按右键，选“属性”，在“系统属性”窗口中选“高级”标签，再按“环境变量”按钮，弹出一个“环境变量”的窗口，在用户变量中新建一个变量，变量名为“Path”，变量值为"C:\Program Files\Java\jdk1.5.0\bin.%PATH%”。最后的%PATH%的意思是说，保留原有的Path设置，且将目前的Path设置新加到其前面。一路按“确定”退出(共有3次)。关掉原来的命令行窗口，依照第2步，重新打开一个新的命令行窗口。在此窗口中输入javac长长的洋文又出现了，这回是介绍javac的用法。设置成功。

6. So far so good. 到目前为止，我们已经可以编程了。但是，这不是一个好办法。因为随着以后我们深入学习Java，我们就会用到JUnit、Ant或NetBeans等应用工具，这些工具在安装时，都需要一个名为指向JDK路径的“JAVA_HOME”的环境变量，否则就安装不了。因此，我们需要改进第5步，为以后作好准备。依照第5步，弹出“环境变量”的窗口，在用户变量中新建一个变量，变量名为“JAVA_HOME”，变量值为"C:\Program Files\Java\jdk1.5.0”。注意，这里的变量值只到jdk1.5.0，不能延伸到bin中。确定后，返回“环境变量”的窗口，双击我们原先设定的Path变量，将其值修改为“%JAVA_HOME%\bin.%PATH%”。这种效果与第5步是完全一样的，只不过多了一个JAVA_HOME的变量。这样，以后当我们需要指向JDK的路径时，只需要加入

“ %JAVA_HOME% ”就行了。至此，Path路径全部设置完毕。一路确定退出，打开新的命令行窗口，输入javac如果长长的洋文出现，Path已经设置正确，一切正常。如果不是，请仔细检查本步骤是否完全设置正确。

7. 开始编程。在C盘的根目录中新建一个子目录，名为“JavaTest”，以作为存放Java源代码的地方。打开XP中的记事本，先将其保存到JavaTest文件夹中，在“文件名”文本框中输入"Hello.java"。注意，在文件名的前后各加上一个双引号，否则，记事本就会将其存为"Hello.java.txt"的文本文件。然后输入以下代码：`public class Hello {public static void main(String[] args) {System.out.println("Hello, world").}}`再次保存文件。

8. 在命令行窗口中输入`cd C:\JavaTest`将当前路径转入JavaTest中。然后，输入`javac Hello.java`JDK就在JavaTest文件夹中编译生成一个Hello.class的类文件。如果出现“1 error”或“XX errors”的字样，说明是源代码的输入有误，请根据出错提示，仔细地按第7步的代码找出并修正错误。请读者注意甄别代码输入有误的问题与classpath设置有误的问题。因为本文是关于如何正确设置classpath及package的，因此，这里假设读者输入的代码准确无误。到目前为此，由于我们是在源代码的当前路径下编译，因此，不会出现classpath设置有误的问题。

9. 在命令行窗口中输入`java Hello`屏幕出现了Hello world成功了，我们已经顺利地编译及运行了第一个Java程序。但是，第8步及第9步是不完美的，因为我们是在JavaTest这个存放源码的文件夹中进行编译及运行的，因此，一些非常重要的问题并没有暴露出来。实际上，第8步的“`javac Hello.java`”及第9步的“`java Hello`”涉及到两个问题，一是操作系统如何寻找“javac”及

“ java ” 等命令，二是操作系统如何寻找 “ Hello.java ” 及 “ Hello.class ” 这些用户自己创建的文件。对于 “ javac ” 及 “ java ” 等命令，由于它们均是可执行文件，操作系统就会依据我们在第6步中设置好的Path路径中去寻找。而对于 “ Hello.java ” 及 “ Hello.class ” 这些文件，Path的设置不起作用。由于我们是在当前工作路径中工作，java及javac会在当前工作路径中寻找相应的java文件(class文件的寻找比较特殊，详见第11步)，因此一切正常。下面我们开始人为地将问题复杂化，在非当前工作路径中编译及运行，看看结果如何。

10. 在命令行窗口中输入cd C:转入到C盘根目录上，当前路径离开了存放源码的工作区。输入javac Hello.java屏幕出现：error: cannot read: Hello.java1 error找不到Hello.java了。我们要给它指定一个路径，告诉它到C:\JavaTest去找Hello.java文件。输入javac C:\JavaTest\Hello.javaOK，这回不报错了，编译成功。

11. 输入java C:\JavaTest\Hello这回屏幕出现：Exception in thread "main" java.lang.NoClassDefFoundError: C:\JavaTest\Hello意思为在 “ C:\JavaTest\Hello ” 找不到类的定义。明明C:\JavaTest\Hello是一个.class文件，为什么就找不到呢？原来，Java对待.java文件与.class文件是有区别的。对.java文件可以直接指定路径给它，而java命令所需的.class文件不能出现扩展名，也不能指定额外的路径给它。那么，如何指定路径呢？对于Java所需的.class文件，必须通过classpath来指定。

12. 依照第5步，弹出 “ 环境变量 ” 窗口，在用户变量中新建一个变量，变量名为 “ classpath ” ，变量值为"C:\JavaTest"。一路按 “ 确定 ” 退出。关闭原命令行窗口，打开新的命令行窗口，输入java Hello “ Hello world ” 出来了。由此可见，在 “ 环境变

量”窗口中设置classpath的目的就是告诉JDK，到哪里去寻找.class文件。这种方法一旦设置好，以后每次运行java或javac时，在需要调用.class文件时，JDK都会自动地来到这里寻找。因此，这是一个全局性的设置。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