

JavaSE6调用Java编译器的两种新方法 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_JavaSE6_E8_B0_83_c104_145031.htm 在很多Java应用中需要在程序中调用Java编译器来编译和运行。但在早期的版本中（Java SE5及以前版本）中只能通过tools.jar中的com.sun.tools.javac包来调用Java编译器，但由于tools.jar不是标准的Java库，在使用时必须设置这个jar的路径。而在Java SE6中为我们提供了标准的包来操作Java编译器，这就是javax.tools包。使用这个包，我们可以不用将jar文件路径添加到classpath中了。

一、使用JavaCompiler接口来编译Java源程序 使用Java API来编译Java源程序有很多方法，现在让我们来看一种最简单的方法，通过JavaCompiler进行编译。我们可以通过ToolProvider类的静态方法getSystemJavaCompiler来得到一个JavaCompiler接口的实例。

```
JavaCompiler compiler =  
ToolProvider.getSystemJavaCompiler().
```

JavaCompiler中最核心的方法是run。通过这个方法可以编译java源程序。这个方法有3个固定参数和1个可变参数（可变参数是从Java SE5开始提供的一个新的参数类型，用type... argu表示）。前3个参数分别用来为java编译器提供参数、得到Java编译器的输出信息以及接收编译器的错误信息，后面的可变参数可以传入一个或多个Java源程序文件。如果run编译成功，返回0。

```
int  
run(InputStream in, OutputStream out, OutputStream err, String...  
arguments)
```

如果前3个参数传入的是null，那么run方法将以标准的输入、输出代替，即System.in、System.out和System.err。如果我们要编译一个test.java文件，并将使用标准输入输出

, run的使用方法如下 : int results = tool.run(null, null, null, "test.java"). 下面是使用JavaCompiler的完整代码 : import java.io.*.import javax.tools.*.public class test_compilerapi{ public static void main(String args[]) throws IOException { JavaCompiler compiler = ToolProvider.getSystemJavaCompiler(). int results = compiler.run(null, null, null, "test.java"). System.out.println((results == 0)?"编译成功":"编译失败"). // 在程序中运行test Runtime run = Runtime.getRuntime(). Process p = run.exec("java test"). BufferedInputStream in = new BufferedInputStream(p.getInputStream()). BufferedReader br = new BufferedReader(new InputStreamReader(in)). String s. while ((s = br.readLine()) != null) System.out.println(s). }} public class test{ public static void main(String[] args) throws Exception { System.out.println("JavaCompiler测试成功 ! "). }} 编译成功的输出结果 : 编译成功 JavaCompiler测试成功 编译失败的输出结果 : test.java:9: 找不到符号符号 : 方法 println(java.lang.String)位置 : 类 java.io.PrintStreamSystem.out.println("JavaCompiler测试成功 ! ").^1 错误编译失败

二、使用StandardJavaFileManager编译Java源程序

在第一部分我们讨论调用java编译器的最容易的方法。这种方法可以很好地工作 , 但它确不能更有效地得到我们所需要的信息 , 如标准的输入、输出信息。而在Java SE6中最好的方法是使用StandardJavaFileManager类。这个类可以很好地控制输入、输出 , 并且可以通过DiagnosticListener得到诊断信息 , 而DiagnosticCollector类就是listener的实现。使用StandardJavaFileManager需要两步。首先建立一

个DiagnosticCollector实例以及通过JavaCompiler的getStandardFileManager()方法得到一个StandardFileManager对象。最后通过CompilationTask中的call方法编译源程序。在使用这种方法调用Java编译时最复杂的方法就是getTask，下面让我们讨论一下getTask方法。这个方法有如下所示的6个参数

。getTask(Writer out,JavaFileManager fileManager,DiagnosticListener diagnosticListener,Iterable options,Iterable classes,Iterable compilationUnits) 这些参数大多数都可为null。它们的含义如下。 out:：用于输出错误的流，默认是System.err。 fileManager:：标准的文件管理。

diagnosticListener: 编译器的默认行为。 options: 编译器的选项 classes : 参与编译的class。 最后一个参数compilationUnits不能为null，因为这个对象保存了你想编译的Java文件。在使用完getTask后，需要通过StandardJavaFileManager

的getJavaFileObjectsFromFiles或getJavaFileObjectsFromStrings方法得到compilationUnits对象。调用这两个方法的方式如下

```
：.Iterable getJavaFileObjectsFromFiles(Iterable files)Iterable  
getJavaFileObjectsFromStrings(Iterable names)String[] filenames =  
....Iterable compilationUnits  
=fileManager.getJavaFileObjectsFromFiles(Arrays.asList(filenames))  
.JavaCompiler.CompilationTask task = compiler.getTask(null,  
fileManager,diagnostics, options, null, compilationUnits). 最后需要  
关闭fileManager.close(). 下面是一个完整的演示程序。import  
java.io.*.import java.util.*.import javax.tools.*.public class  
test_compilerapi{ private static void compilejava() throws Exception  
{ JavaCompiler compiler = ToolProvider.getSystemJavaCompiler().
```

```
// 建立DiagnosticCollector对象 DiagnosticCollector diagnostics =  
new DiagnosticCollector(). StandardJavaFileManager fileManager =  
compiler.getStandardFileManager(diagnostics, null, null). // 建立用  
于保存被编译文件名的对象 // 每个文件被保存在一个  
从JavaFileObject继承的类中 Iterable compilationUnits =  
fileManager.getJavaFileObjectsFromStrings(Arrays.asList("test3.java"  
)). JavaCompiler.CompilationTask task = compiler.getTask(null,  
fileManager, diagnostics, null, null, compilationUnits). // 编译源程  
序 boolean success = task.call(). fileManager.close().  
System.out.println((success)?"编译成功":"编译失败"). } public  
static void main(String args[]) throws Exception { compilejava(). }}  
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
```