

调用Java编译器API编译Java PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/214/2021_2022__E8_B0_83_E7_94_A8Java_c104_214154.htm 从第一天开始,标准Java平台就缺少能够被调用,去产生Java字节码的编译器接口. 使用Sun实现的平台,一个用户可以通过非标准的 com.sun.tools.javac 包中的Main class 去编译你的代码 (你可以在lib子目录下的 tools.jar 文件里找到它). 然而这个包并没有提供一个标准的公开的编程接口. 使用其它实现的用户必然不能访问这个类. 使用Java SE 6和在JSR-199中定义的它的新的Java编译器接口,你可以从你自己的应用程序里访问javac编译工具了. 有两种方式使用这种工具. 一种是简单的,一种是稍微复杂点但拥有更多选项的. 你首先将会用较简单的一种去编译 "Hello, World"程序,就是下面的这个:

```
public class Hello { public static void main(String args[]) { System.out.println("Hello, World"). } }
```

要想从Java程序里调用Java编译器,你需要访问JavaCompiler 接口. 除此外,通过访问这个接口,你可以设置源代码的路径,classpath,和目标目录. 通过指定可编译的文件为 JavaFileObjectinstance ,你可以将它们全部编译. 然而,你并不需要对 JavaFileObject 了解多少. 可以使用 ToolProvider 类去请求 theJavaCompiler 接口的缺省实现. 这个 ToolProvider 类提供了一个 getSystemJavaCompiler() 方法, 它返回一个 JavaCompiler 接口的实例.

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

使用 JavaCompiler 运行编译最简单的方法是调用在这个接口工具里定义的 run() 方法, 它的实现是:

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

分别为前三个缺省参数

System.in, System.out, and System.err 传入 null 值. 参数集 String 对象表示着传入编译器的文件的名称. 这样, 你应该像下面这样去编译前面显示的 Hello 源程序: `int results = tool.run(null, null, null, "Hello.java")`. 假设没有编译错误, 这样会在目标目录里产生一个 Hello.class 文件. 如果这里有错误, `run()` 方法会把它输出到标准错误输出流里, 也就是 `run()` 方法的第三个参数. 当错误发生时这个方法返回一个非0的结果. 你可以使用下面的代码去编译 Hello.java 源文件:

```
import java.io.*; import
javax.tools.*; public class CompileIt { public static void main(String
args[]) throws IOException { JavaCompiler compiler =
ToolProvider.getSystemJavaCompiler(). int results = compiler.run(
null, null, null, "Hello.java"). System.out.println("Result code: "
results). }}
```

一旦你编译了 `CompileIt` 一次, 你就可以多次运行它, 当你修改了 Hello.java 源程序时或者要重新编译它, 你不需要重新编译 `CompileIt`. 如果没有错误, 运行 `CompileIt` 会产生下面的输出: `> java CompileIt Result code: 0` 运行 `CompileIt` 同样也会在相同的目录下产生一个 Hello.class 文件: `>`

```
ls CompileIt.class CompileIt.java Hello.class Hello.java 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