

如何在Java中执行其它程序Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/594/2021_2022__E5_A6_82_E4_BD_95_E5_9C_A8J_c104_594684.htm 在编写Java程序时，有时候需要在Java程序中执行另外一个程序。1、启动程序Java提供了两种方法用来启动其它程序：（1）使用Runtime的exec（）方法（2）使用ProcessBuilder的start（）方法不管在何种操作系统下，程序具有基本类似的一些属性。一个程序启动后就程序操作系统的一个进程，进程在执行的时候有自己的环境变量、有自己的工作目录。Runtime和ProcessBuilder提供了不同的方式来启动程序，设置启动参数、环境变量和工作目录。能够在Java中执行的外部程序，必须是一个实际存在的可执行文件，对于shell下的内嵌命令是不能直接执行的。采用Runtime的exec执行程序时，首先要使用Runtime的静态方法得到一个Runtime，然后调用Runtime的exec方法。可以将要执行的外部程序和启动参数、环境变量、工作目录作为参数传递给exec方法，该方法执行后返回一个Process代表所执行的程序。Runtime有六个exec方法，其中两个的定义为：
`public Process exec（String[] cmdarray，String[] envp，File dir）`
`public Process exec（String command，String[] envp，File dir）`
cmdarray和command为要执行的命令，可以将命令和参数作为一个字符串command传递给exec（）方法，也可以将命令和参数一个一个的放在数组cmdarray里传递给exec（）方法。envp为环境变量，以name=value的形式放在数组中。dir为工作目录。可以不要dir参数，或者不要envp和dir参数，这样就多出了其它4个exec（）方法。如果

没有dir参数或者为null，那么新启动的进程就继承当前java进程的工作目录。如果没有envp参数或者为null，那么新启动的进程就继承当前java进程的环境变量。也可以使用ProcessBuilder类启动一个新的程序，该类是后来添加到JDK中的，而且被推荐使用。通过构造函数设置要执行的命令以及参数，或者也可以通过command（）方法获取命令信息后再进行设置。通过directory（File directory）方法设置工作目录，通过environment（）获取环境变量信息来修改环境变量。在使用ProcessBuilder构造函数创建一个新实例，设置环境变量、工作目录后，可以通过start（）方法来启动新程序，与Runtime的exec（）方法一样，该方法返回一个Process对象代表启动的程序。ProcessBuilder与Runtime.exec（）方法的不同在于ProcessBuilder提供了redirectErrorStream（boolean redirectErrorStream）方法，该方法用来将进程的错误输出重定向到标准输出里。即可以将错误输出都将与标准输出合并。

2、Process 不管通过那种方法启动进程后，都会返回一个Process类的实例代表启动的进程，该实例可用来控制进程并获得相关信息。Process 类提供了执行从进程输入、执行输出到进程、等待进程完成、检查进程的退出状态以及销毁（杀掉）进程的方法：

- （1）void destroy（） 杀掉子进程。一般情况下，该方法并不能杀掉已经启动的进程，不用为好。
- （2）int exitValue（） 返回子进程的出口值。只有启动的进程执行完成、或者由于异常退出后，exitValue（）方法才会有正常的返回值，否则抛出异常。
- （3）InputStream getErrorStream（） 获取子进程的错误流。如果错误输出被重定向，则不能从该流中读取错误输出。
- （4）InputStream

getInputStream () 获取子进程的输入流。 可以从该流中读取进程的标准输出。 (5) OutputStream getOutputStream () 获取子进程的输出流。 写入到该流中的数据作为进程的标准输入。 (6) int waitFor () 导致当前线程等待，如有必要，一直要等到由该 Process 对象表示的进程已经终止。 通过该类提供的方法，可以实现与启动的进程之间通信，达到交互的目的。

3、从标准输出和错误输出流读取信息

从启动其他程序的Java进程看，已启动的其他程序输出就是一个普通的输入流，可以通过getInputStream () 和getErrorStream来获取。 对于一般输出文本的进程来说，可以将InputStream封装成BufferedReader，然后就可以一行一行的对进程的标准输出进行处理。

4、举例 (1)

```
Runtime.exec ( ) import
java.io.BufferedReader. import java.io.File. import
java.io.InputStreamReader. public class Test1 { public static void
main(String[] args) { try { Process p = null. String line = null.
BufferedReader stdout = null. //list the files and directorys under C:\
p = Runtime.getRuntime().exec("CMD.exe /C dir", null, new
File("C:\\")). stdout = new BufferedReader(new
InputStreamReader(p.getInputStream())). while ((line =
stdout.readLine()) != null) { System.out.println(line). }
stdout.close(). //echo the value of NAME p =
Runtime.getRuntime().exec("CMD.exe /C echo %NAME%", new
String[] {"NAME=TEST"}). stdout = new BufferedReader(new
InputStreamReader(p.getInputStream())). while ((line =
stdout.readLine()) != null) { System.out.println(line). }
stdout.close(). } catch (Exception e) { e.printStackTrace(). } } } ( 2
```

```
) ProcessBuilder import java.io.BufferedReader.import
java.io.File.import java.io.InputStreamReader.import
java.util.ArrayList.import java.util.List.public class Test2 {public
static void main(String[] args) {try {List list = new
ArrayList().ProcessBuilder pb = null.Process p = null.String line =
null.BufferedReader stdout = null.//list the files and directorys under
C:\list.add("CMD.EXE").list.add("/C").list.add("dir").pb = new
ProcessBuilder(list).pb.directory(new File("C:\\")).p =
pb.start().stdout = new BufferedReader(new
InputStreamReader(p.getInputStream())).while ((line =
stdout.readLine()) != null)
{System.out.println(line).}stdout.close().//echo the value of
NAMEpb = new ProcessBuilder().pb.command(new String[]
{"CMD.exe", "/C", "echo
%NAME%"}).pb.environment().put("NAME", "TEST").p =
pb.start().stdout = new BufferedReader(new
InputStreamReader(p.getInputStream())).while ((line =
stdout.readLine()) != null)
{System.out.println(line).}stdout.close().} catch (Exception e)
{e.printStackTrace().}}}
```

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