<http://www.360doc.com/content/07/0927/10/36589_773953.shtml>

* 1．关于动态加载机制——  
  学习Java比C++更容易理解OOP的思想，毕竟C++还混合了不少面向过程的成分。很多人都能背出来Java语言的特点，所谓的动态加载机制等等。当然概念往往是先记住而后消化的，可有多少人真正去体会过动态加载的机制，试图去寻找过其中的细节呢? 提供大家一个方法：  
  在命令行窗口运行Java程序的时候，加上这个很有用的参数：

java –verbose \*.class

这样会清晰的打印出被加载的类文件，大部分是jdk自身运行需要的，最后几行会明显的看到自己用到的那几个类文件被加载进来的顺序。即使你声明了一个类对象，不实例化也不会加载，说明只有真正用到那个类的实例即对象的时候，才会执行加载。这样是不是大家稍微能明白一点动态加载了呢？^\_^

2．关于寻找class文件原理——  
建议大家在入门的时候在命令行窗口编译和运行，不要借助JCreator或者Eclipse等IDE去帮助做那些事情。尝试自己这样做：  
javac -classpath yourpath \*.java  
java -classpath yourpath \*.class  
也许很多人都能看懂，设置classpath的目的就是告诉编译器去哪里寻找你的class文件. 不过至少笔者今日才弄懂JVM去查询类的原理，编译器加载类要依靠classloader， 而classloader有3个级别，从高到低分别是BootClassLoader(名字可能不准确) , ExtClassLoader, AppClassLoader.

这3个加载器分别对应着编译器去寻找类文件的优先级别和不同的路径：BootClassLoader对应jre/classes路径，是编译器最优先寻找class的地方  
ExtClassLoader对应jre/lib/ext路径，是编译器次优先寻找class的地方  
AppClassLoader对应当前路径，所以也是编译器默认找class的地方

其实大家可以自己写个程序简单的测试，对任何class，例如A,   
调用new A().getClass().getClassLoader().toString() 打印出来就可以看到，把class文件放在不同的路径下再次执行，就会看到区别。特别注意的是如果打印出来是null就表示到了最高级BootClassLoader, 因为它是C++编写的，不存在Java对应的类加载器的名字。

寻找的顺序是一种向上迂回的思想，即如果本级别找不到，就只能去本级别之上的找，不会向下寻找。不过似乎从Jdk1.4到Jdk1.6这一特点又有改变，没有找到详细资料。所以就不举例子了。告诉大家设计这种体系的是Sun公司曾经的技术核心宫力先生，一个纯种华人哦！^\_^

这样希望大家不至于迷惑为什么总报错找不到类文件，不管是自己写的还是导入的第三方的jar文件（J2ee中经常需要导入的）。

3．关于jdk和jre——  
大家肯定在安装JDK的时候会有选择是否安装单独的jre，一般都会一起安装，我也建议大家这样做。因为这样更能帮助大家弄清楚它们的区别：

Jre 是java runtime environment, 是java程序的运行环境。既然是运行，当然要包含jvm，也就是大家熟悉的虚拟机啦， 还有所有java类库的class文件，都在lib目录下打包成了jar。大家可以自己验证。至于在windows上的虚拟机是哪个文件呢？ 学过MFC的都知道什么是dll文件吧，那么大家看看jre/bin/client里面是不是有一个jvm.dll呢？那就是虚拟机。

Jdk 是java development kit，是java的开发工具包，里面包含了各种类库和工具。当然也包括了另外一个Jre. 那么为什么要包括另外一个Jre呢？而且jdk/jre/bin同时有client和server两个文件夹下都包含一个jvm.dll。 说明是有两个虚拟机的。这一点不知道大家是否注意到了呢？

相信大家都知道jdk的bin下有各种java程序需要用到的命令，与jre的bin目录最明显的区别就是jdk下才有javac，这一点很好理解，因为jre只是一个运行环境而已。与开发无关，正因为如此，具备开发功能的jdk自己的jre下才会同时有client性质的jvm和server性质的jvm， 而仅仅作为运行环境的jre下只需要client性质的jvm.dll就够了。

记得在环境变量path中设置jdk/bin路径麽？这应该是大家学习Java的第一步吧， 老师会告诉大家不设置的话javac和java是用不了的。确实jdk/bin目录下包含了所有的命令。可是有没有人想过我们用的java命令并不是jdk/bin目录下的而是jre/bin目录下的呢？不信可以做一个实验，大家可以把jdk/bin目录下的java.exe剪切到别的地方再运行java程序，发现了什么？一切OK！

那么有人会问了？我明明没有设置jre/bin目录到环境变量中啊？

试想一下如果java为了提供给大多数人使用，他们是不需要jdk做开发的，只需要jre能让java程序跑起来就可以了，那么每个客户还需要手动去设置环境变量多麻烦啊？所以安装jre的时候安装程序自动帮你把jre的java.exe添加到了系统变量中，验证的方法很简单，大家看到了系统环境变量的path最前面有“%SystemRoot%\system32;%SystemRoot%;”这样的配置，那么再去Windows/system32下面去看看吧，发现了什么？有一个java.exe。

如果强行能够把jdk/bin挪到system32变量前面，当然也可以迫使使用jdk/jre里面的java，不过除非有必要，我不建议大家这么做。使用单独的jre跑java程序也算是客户环境下的一种测试。

这下大家应该更清楚jdk和jre内部的一些联系和区别了吧？

PS: 其实还有满多感想可以总结的，一次写多了怕大家扔砖头砸死我，怪我太罗唆。大家应该更加踏实更加务实的去做一些研究并互相分享心得，大方向和太前沿的技术讨论是必要的但最好不要太多，毕竟自己基础都还没打好，什么都讲最新版本其实是进步的一大障碍！

Jre 是java runtime environment, 是java程序的运行环境。既然是运行，当然要包含jvm，也就是大家熟悉的虚拟机啦， 还有所有java类库的class文件，都在lib目录下打包成了jar。大家可以自己验证。至于在windows上的虚拟机是哪个文件呢？ jre/bin/client里面是不是有一个jvm.dll呢？那就是虚拟机。

Jdk 是java development kit，是java的开发工具包，里面包含了各种类库和工具。当然也包括了另外一个Jre. 那么为什么要包括另外一个Jre呢？而且jdk/jre/bin同时有client和server两个文件夹下都包含一个jvm.dll。 说明是有两个虚拟机的。这一点不知道大家是否注意到了呢？

相信大家都知道jdk的bin下有各种java程序需要用到的命令，与jre的bin目录最明显的区别就是jdk下才有javac，这一点很好理解，因为 jre只是一个运行环境而已。与开发无关，正因为如此，具备开发功能的jdk自己的jre下才会同时有client性质的jvm和server性质的 jvm， 而仅仅作为运行环境的jre下只需要client性质的jvm.dll就够了。

记得在环境变量path中设置jdk/bin路径麽？这应该是大家学习Java的第一步吧， 老师会告诉大家不设置的话javac和java是用不了的。确实jdk/bin目录下包含了所有的命令。可是有没有人想过我们用的java命令并不是 jdk/bin目录下的而是jre/bin目录下的呢？不信可以做一个实验，大家可以把jdk/bin目录下的java.exe剪切到别的地方再运行 java程序，发现了什么？一切OK！  
  
那么有人会问了？我明明没有设置jre/bin目录到环境变量中啊？  
  
试想一下如果java为了提供给大多数人使用，他们是不需要jdk做开发的，只需要jre能让java程序跑起来就可以了，那么每个客户还需要手动去设置环境变量多麻烦啊？所以安装jre的时候安装程序自动帮你把jre的java.exe添加到了系统变量中，验证的方法很简单，去Windows/system32下面去看看吧，发现了什么？有一个java.exe。

以上是我最近研究JRE和JDK区别时通过网上寻找的材料然后自己整理的，希望对大家有点帮助！

<http://www.cnblogs.com/decennium/archive/2012/03/17/2403985.html>

Jre 是java runtime environment, 是java程序的运行环境。既然是运行，当然要包含jvm，也就是大家熟悉的虚拟机，   
还有所有java类库的class文件，都在lib目录下打包成了jar。大家可以自己验证。至于在windows上的虚拟机是哪个文件呢？   
jre/bin/client里面是不是有一个jvm.dll呢？那就是虚拟机。  
  
Jdk 是java development kit，是java的开发工具包，里面包含了各种类库和工具。当然也包括了另外一个Jre.   
那么为什么要包括另外一个Jre呢？而且jdk/jre/bin同时有client和server两个文件夹下都包含一个jvm.dll。   
说明是有两个虚拟机的。这一点不知道大家是否注意到了呢？  
相信大家都知道jdk的bin下有各种java程序需要用到的命令，与jre的bin目录最明显的区别就是jdk下才有javac，  
这一点很好理解，因为 jre只是一个运行环境而已。与开发无关，正因为如此，具备开发功能的jdk自己的jre下才会同时有client性质的jvm和server性质的 jvm，   
而仅仅作为运行环境的jre下只需要client性质的jvm.dll就够了。  
  
记得在环境变量path中设置jdk/bin路径麽？这应该是大家学习Java的第一步吧，   
老师会告诉大家不设置的话javac和java是用不了的。确实jdk/bin目录下包含了所有的命令。  
可是有没有人想过我们用的java命令并不是 jdk/bin目录下的而是jre/bin目录下的呢？  
不信可以做一个实验，大家可以把jdk/bin目录下的java.exe剪切到别的地方再运行 java程序，发现了什么？一切OK！  
那么有人会问了？我明明没有设置jre/bin目录到环境变量中啊？

试想一下如果java为了提供给大多数人使用，他们是不需要jdk做开发的，只需要jre能让java程序跑起来就可以了，  
那么每个客户还需要手动去设置环境变量多麻烦啊？所以安装jre的时候安装程序自动帮你把jre的java.exe添加到了系统变量中，  
验证的方法很简单，去Windows/system32下面去看看吧，发现了什么？有一个java.exe。

以上是我最近研究JRE和JDK区别时通过网上寻找的材料然后自己整理的，希望对大家有点帮助！

环境变量配置：

Classpath: 告诉类装载器到哪里去寻找三方提供的类和用户定义的类  
  
.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;  
  
Java\_home: jdk变量   
  
C:\Program Files\Java\jdk1.6.0\_10;  
  
Path: Windows操作系统用来寻找可执行文件的路径  
  
C:\Program Files\Java\jdk1.6.0\_10\bin;

<http://xinklabi.iteye.com/blog/964076>

JDK(java develop kit ,java开发套件)，JRE(java runtime environment,java运行时环境),JVM(java virtual machine,java虚拟机)  
JDK  
++  
|-----jre  
++   
|-------jvm  
|-------  
jre只是jdk的一个部分，我们只要拥有了一份jdk就可以进行java程序开发了，而顾名思义，jre是java运行的时候所依赖的平台，jvm则更只是jre的一个部分，jre除了虚拟机外还提供了众多链接库，这些库会被jvm在运行时调用。  
我们会发现java\_home/下有一个bin文件夹和一个jre文件夹，而jre文件夹下也有一个bin文件夹，它们一样吗？显然不一样。  
java\_home/bin中的工具是sun公司提供给我们一些工具，如java.exe,javac.exe,而jre/bin下的工具是运行时所需的，于是我们可以发现java\_home/bin下有一个javac.exe,而jre/bin下肯定没有javac.exe，这是很容易解释的。  
奇怪的是，sun公司给我们提供了两套jre,稍留意一下我们就会发现在装完jdk，还可以选择装一个jre,这个jre是被默认装到 C:\Program Files\Java下的。sun提供两套jre是有道理的，如果我们只是想运行java程序，那么去官网下一个jre就行了，而无需拿来一个jdk,如果是要开发java程序那就需要一个jdk了."jdk里面的工具几乎都是用java写的，所以jdk本身就是java应用程序，因此要使用jdk附的工具来开发java程序也必须要自行附一套jre才行，这就是java\_home\jre存在的原因，而位于Program File\jre的那套jre就是拿来执行我们自己写的java应用程序的。不过，两套中任何一套jre都可以拿来执行我们所写的java应用程序，可是 jdk内附的开发工具在预设开发工具使用包装器（.exe）来启动的情况下，都拿自己支选用java\_home\jre".(《Java深度历险》)  
因此，现在可以明白设path和classpath的作用了：  
  
path是为jdk提供的那些工具设的，如java.exe,javac.exe 。  
classpath是为java程序所用到的库文件所设的 。  
  
于是path设成：java\_home\bin;而classpath设为：java\_home\jre\lib\rt.jar,如果我们开发的一个应用中用到了非j2se提供的库文件，则需将这些文件加到classpath 中，比如说利用javamail开发一个邮件应用，就应该把mail.jar  
等文件加到classpath中：java\_home\jre\lib\ext\mail.jar;.  
如果一台PC上装了几个jre，那么就要注意了，有时很容易产生问题，很可能我们用来编译java程序的javac.exe所相关的jre与运行程序时所用的jre不是一个版本， 这可能会产生版本冲突，这可以通过设定path变量来解决。  
最后应用中的第三方文件应该加到java\_home\jre\lib\ext下，就像刚才那样，与java相关的安全设定档都放置java\_home \jre\lib\security。最后扩展一点，与java相关的很多工具的path设定原则都与上述差不多，比如说想安装一个Ant,应该在 path 中这样设：  
F:\\Ant\apache-ant-1.6.5\bin;看看bin下有什么就明白这样做是为了指定命令工具ant所在的目录。

JDK(Java Development Kit)是**面向开发人员**使用的SDK，它提供了Java的开发环境和运行环境。SDK是Software Development Kit 一般指软件开发包，可以包括函数库、编译程序等。

JRE(Java Runtime Environment)是指Java的运行环境，是**面向Java程序的使用者**，而不是开发者。

JVM（Java Virtual Machine）是java虚拟机，它是整个java实现跨平台的最核心的部分，所有的java程序会首先被编译为.class的类文件，这种类文件可以在虚拟机上执行，也就是说class并不直接与机器的操作系统相对应，而是经过虚拟机间接与操作系统交互，由虚拟机将程序解释给本地系统执行。

如果安装了JDK，会发现你的电脑有两套JRE，一套位于Java\jdk\jre ,另外一套位于 Java\jre目录下，后面这套比前面那套少了Server端的Java虚拟机。jdk的bin下有各种java程序需要用到的命令，与jre下的bin目录最明显的区别就是jdk下才有javac，这一点很好理解，因为 jre只是一个运行环境而已，与开发无关，正因为如此，具备开发功能的jdk自己的jre下才会同时有client性质的jvm和server性质的 jvm， 而仅仅作为运行环境的jre下只需要client性质的jvm.dll就够了。

JRE的地位就像一台PC机一样，我们写好的 Win32应用程序需要操作系统帮我们运行，同样的，我们编写的Java程序也必须要JRE才能运行。所以当你装完JDK后，如果分别在硬盘上的两个不同地方安装了两套JRE，那么你可以想象你的电脑有两台虚拟的Java PC机，都具有运行Java程序的功能。所以我们可以说，只要你的电脑安装了JRE，就可以正确运行Java应用程序。

1． 为什么Sun要让JDK安装两套相同的JRE？

这是因为JDK里面有很多用Java所编写的开发工具（如javac.exe、jar.exe等），而 且都放置在 \lib\tools.jar 里。从下面例子可以看出，先将tools.jar改名为tools1.jar，然后运行javac.exe，显示如下结果： Exception in thread "main" java.lang.NoClassDefFoundError: com/sun/tools/javac /Main 这个意思是说，你输入javac.exe与输入 java -cp c:\jdk\lib\tools.jar com.sun.tools.javac.Main 是一样的，会得到相同的结果。从这里我们可以证明javac.exe只是一个包装器（Wrapper），而制作的目的是为了让开发者免于输入太长的指命。 而且可以发现\lib目录下的程序都很小，不大于29K，从这里我们可以得出一个结论，就是JDK里的工具几乎是用Java所编写，所以也是Java应用程序，因此要使用JDK所附的工具来开发Java程序，也必须要自行附一套JRE才行，所以位于Java目录下的那套JRE就是用来运行一般Java程序用的，Java\jdk\jre是用来开发程序的。

2.如果一台电脑安装两套以上的JRE，谁来决定呢？

这个重大任务就落在java.exe身上。java.exe的工作就是找到合适的JRE来运行 Java程序。 java.exe依照以下顺序来查找JRE：自己的目录下有没有JRE；父目录有没有JRE；查询注册表： [HKEY\_LOCAL\_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment] 所以java.exe的运行结果与你的电脑里面哪个JRE被执行有很大的关系。

3．介绍JVM JRE目录下的Bin目录有两个目录：server与client，这就是真正的jvm.dll所在。 当jvm.dll启动后，会使用explicit的方法（就是使用Win32 API之中的LoadLibrary（）与GetProcAddress（）来载入辅助用的动态链接库），而这些辅助用的动态链接库（dll）都必须位于jvm.dll所在目录的父目录之中。因此想使用哪个JVM，只需要设置PATH，指向JRE所在目录底下的jvm.dll.