关于java源程序打包的问题，比如：

如何把\*.java（源程序文件）转换成\*.class（二进制文件）？

如何将其打包成别的电脑可直接执行的jar文件？（前提是对方电脑要配置好java环境变量，主要还是为了java虚拟机）

1. 如何把\*.java（源程序文件）转换成\*.class（二进制文件）？

很简单的问题，

a.首先，先在.java文件所在的目录的上一级新建一个classes文件夹，(用于存放生成的多个.class文件)

b.再切换到.java文件所在的目录下，然后打开终端或命令行

输入

javac -d ../classes MainClassName.java

(-d 是指定生成的class文件所在目录，\*.java是编译当前目录下所有的java文件, MainClassName.java是你的主类java文件，根据实际情况改成自己的)

结果就是classes文件夹下生成了java源文件中每个类的class文件

P.S.:现在生成的class文件能运行吗？

答：可以，但是只能在本机运行，进入到classes文件夹下，输入

java MainClassName

（注意MainClassName后面既没有.class也没有.java后缀）

2. 如何将其打包成别的电脑可直接执行的jar文件？

（来一个最简单的打包方式）

在第一步完成javac的前提下我们才可以打包jar文件

a.首先切换到classes目录下

新建manifest.txt文件，然后输入一行文字并保存

Main-Class: MainClassName

（Main-Class告诉编译器主类是哪一个，切记一点Main-Class: 和MainClassName之间有一个空格，一定要有,不然打包成jar文件一定会报错

错误如下：java.io.IOException: invalid header field

）

b.在classes目录下打开终端或命令行，输入

jar -cvmf manifest.txt your.jar \*.class

(jar打包命令，后跟cvmf四个参数，参考manifest.txt的信息

打包好的jar文件自己起名，我这起了your.jar

根据当前目录下所有.class文件进行打包)

然后classes目录下就会有.jar文件啦，双击可直接运行

P.S.:本例的manifest.txt其实是最简单的打包说明文件，标准的打包说明文件应该是manifest.mf，详细解释如下（切记每一项冒号后面都有空格，勿忘）

打开Java的JAR文件我们经常可以看到文件中包含着一个META-INF目录，这个目录下会有一些文件，其中必有一个MANIFEST.MF，这个文件描述了该Jar文件的很多信息，下面将详细介绍MANIFEST.MF文件的内容，先来看struts.jar中包含的MANIFEST.MF文件内容：   
Manifest-Version: 1.0   
Created-By: Apache Ant 1.5.1   
Extension-Name: Struts Framework   
Specification-Title: Struts Framework   
Specification-Vendor: Apache Software Foundation   
Specification-Version: 1.1   
Implementation-Title: Struts Framework   
Implementation-Vendor: Apache Software Foundation   
Implementation-Vendor-Id: org.apache   
Implementation-Version: 1.1   
Class-Path:   commons-beanutils.jar commons-collections.jar commons-dig   
ester.jar commons-logging.jar commons-validator.jar jakarta-oro.jar s   
truts-legacy.jar   
如果我们把MANIFEST中的配置信息进行分类，可以归纳出下面几个大类：   
**一. 一般属性**   
1. Manifest-Version   
用来定义manifest文件的版本，例如：Manifest-Version: 1.0   
2. Created-By   
声明该文件的生成者，一般该属性是由jar命令行工具生成的，例如：Created-By: Apache Ant 1.5.1   
3. Signature-Version   
定义jar文件的签名版本   
4. Class-Path   
应用程序或者类装载器使用该值来构建内部的类搜索路径   
**二. 应用程序相关属性**   
1. Main-Class   
定义jar文件的入口类，该类必须是一个可执行的类，一旦定义了该属性即可通过 java -jar x.jar来运行该jar文件。   
  
**三. 小程序(Applet)相关属性**   
1. Extendsion-List   
该属性指定了小程序需要的扩展信息列表，列表中的每个名字对应以下的属性   
2. <extension>-Extension-Name   
3. <extension>-Specification-Version   
4. <extension>-Implementation-Version   
5. <extension>-Implementation-Vendor-Id   
5. <extension>-Implementation-URL   
**四. 扩展标识属性**   
1. Extension-Name   
该属性定义了jar文件的标识，例如Extension-Name: Struts Framework   
  
**五. 包扩展属性**   
  
1. Implementation-Title    定义了扩展实现的标题   
2. Implementation-Version    定义扩展实现的版本   
3. Implementation-Vendor    定义扩展实现的组织     
4. Implementation-Vendor-Id    定义扩展实现的组织的标识   
5. Implementation-URL :    定义该扩展包的下载地址(URL)   
6. Specification-Title    定义扩展规范的标题   
7. Specification-Version    定义扩展规范的版本   
8. Specification-Vendor    声明了维护该规范的组织   
9. Sealed    定义jar文件是否封存，值可以是true或者false (这点我还不是很理解)   
**六. 签名相关属性**   
签名方面的属性我们可以来参照JavaMail所提供的mail.jar中的一段   
Name: javax/mail/Address.class   
Digest-Algorithms: SHA MD5   
SHA-Digest: AjR7RqnN//cdYGouxbd06mSVfI4=   
MD5-Digest: ZnTIQ2aQAtSNIOWXI1pQpw==   
这段内容定义类签名的类名、计算摘要的算法名以及对应的摘要内容(使用BASE64方法进行编码)   
**七.自定义属性**   
除了前面提到的一些属性外，你也可以在MANIFEST.MF中增加自己的属性以及响应的值，例如J2ME程序jar包中就可能包含着如下信息   
MicroEdition-Configuration: CLDC-1.0   
MIDlet-Name: J2ME\_MOBBER Midlet Suite   
MIDlet-Info-URL: [http://www.javayou.com](http://www.javayou.com/)   
MIDlet-Icon: /icon.png   
MIDlet-Vendor: Midlet Suite Vendor   
MIDlet-1: mobber,/icon.png,mobber   
MIDlet-Version: 1.0.0   
MicroEdition-Profile: MIDP-1.0   
MIDlet-Description: Communicator   
关键在于我们怎么来读取这些信息呢？其实很简单，JDK给我们提供了用于处理这些信息的API，详细的信息请见java.util.jar包中，我们可以通过给JarFile传递一个jar文件的路径，然后调用JarFile的getManifest方法来获取Manifest信息。