**package 的目录结构**

类放在包中会有两种主要的结果：

包名成为类名的一部分，正如我们前面讨论的一样。

包名必须与相应的字节码所在的目录结构相吻合。

下面是管理你自己 java 中文件的一种简单方式：

将类、接口等类型的源码放在一个文本中，这个文件的名字就是这个类型的名字，并以.java作为扩展名。例如：

// 文件名 : Car.java

package vehicle;

public class Car {

// 类实现

}

接下来，把源文件放在一个目录中，这个目录要对应类所在包的名字。

....\vehicle\Car.java

现在，正确的类名和路径将会是如下样子：

类名 -> vehicle.Car

路径名 -> vehicle\Car.java (在 windows 系统中)

通常，一个公司使用它互联网域名的颠倒形式来作为它的包名.例如：互联网域名是 runoob.com，所有的包名都以 com.runoob 开头。包名中的每一个部分对应一个子目录。

例如：有一个 com.runoob.test 的包，这个包包含一个叫做 Runoob.java 的源文件，那么相应的，应该有如下面的一连串子目录：

....\com\runoob\test\Runoob.java

编译的时候，编译器为包中定义的每个类、接口等类型各创建一个不同的输出文件，输出文件的名字就是这个类型的名字，并加上 .class 作为扩展后缀。 例如：

// 文件名: Runoob.java

package com.runoob.test;

public class Runoob {

}

class Google {

}

现在，我们用-d选项来编译这个文件，如下：

$javac -d . Runoob.java

这样会像下面这样放置编译了的文件：

.\com\runoob\test\Runoob.class

.\com\runoob\test\Google.class

你可以像下面这样来导入所有 \com\runoob\test\ 中定义的类、接口等：

import com.runoob.test.\*;

编译之后的 .class 文件应该和 .java 源文件一样，它们放置的目录应该跟包的名字对应起来。但是，并不要求 .class 文件的路径跟相应的 .java 的路径一样。你可以分开来安排源码和类的目录。

<path-one>\sources\com\runoob\test\Runoob.java

<path-two>\classes\com\runoob\test\Google.class

这样，你可以将你的类目录分享给其他的编程人员，而不用透露自己的源码。用这种方法管理源码和类文件可以让编译器和java 虚拟机（JVM）可以找到你程序中使用的所有类型。

类目录的绝对路径叫做 class path。设置在系统变量 CLASSPATH 中。编译器和 java 虚拟机通过将 package 名字加到 class path 后来构造 .class 文件的路径。

<path- two>\classes 是 class path，package 名字是 com.runoob.test,而编译器和 JVM 会在 <path-two>\classes\com\runoob\test 中找 .class 文件。

一个 class path 可能会包含好几个路径，多路径应该用分隔符分开。默认情况下，编译器和 JVM 查找当前目录。JAR 文件按包含 Java 平台相关的类，所以他们的目录默认放在了 class path 中。