包的定义及使用

# 引言

1. 包的主要作用以及定义；
2. 包的导入操作；
3. 系统常见的开发包；
4. Jar程序命令。

# 具体内容

## 包的定义

在任何的操作系统志宏都有一个统一的共识：同一个目录下不能存在有相同的文件名称，但是在实际的开发之中又不可能不出现不同名的情况，所以为了解决这样的冲突矛盾问题，在项目里面往往会引用包的概念，而所谓的包存储时就是一个文件夹，也就是说在不同的目录中可以存放有相同的文件。

如果要想在Java中定义，直接使用package关键字即可。

范例：打包编译

|  |
| --- |
| javac –d . Hello.java |

在以上的编译命令中有两个参数：

* “-d”：生成目录，默认情况下是根据pacakge定义的内容生成；
* “.”：生成程序的保存路径，如果设置为“.”表示当前目录下保存。

此时如果程序有包了，那么在解释程序的时候必须加上包名称。

|  |
| --- |
| java org.dougin.demo.Hello |

所有的Java项目开发过程之中，程序类必须要有包的定义才可以使用，没有包的类不存在。而类的完整名称就是“包.类”。

## 包的导入操作

利用程序包可以将一个很多的\*.class类归类保存起来，但是这样一来就有可能出现包与包之间的互相调用问题，所以此时就可以利用import语句来解决。

范例：自动编译模式

|  |
| --- |
| javac –d . \*.java |

## 系统常见包

Java的开发类库非常的庞大，而开发包也非常的多，比较常见的几个包如下：

* + java.lang：像Object、String、Integer、Number都是这个包中定义的，从JDK 1.1开始自动导入；
  + java.lang.reflect：反射开发包，直接与框架开发有关系；
  + java.util：Java的工具包；
  + java.util.regex：正则表达式的开发包；
  + java.text：国际化程序支持包；
  + java.io：数据流的开发包（Input、Output），前提：抽象类的概念弄明白；
  + java.net：网络编程包；
  + java.sql：数据库的编程包；
  + java.awt、javax.swing：图形界面开发；
    - Java在最初JDK 1.0的时候只提供有java.awt的开发包，这个开发包可以实现图形界面的开发，但是最初的设计较为仓促，要大量的使用windows的系统函数；
    - 在JDK 1.2（Java2）推出了轻量级的组件，有了swing开发组件；
  + java.applet.\*：是在网页上嵌入的java程序，是Java发展得标记；

## jar命令

当一个项目开发完成之后，虽然不同功能的类要保存在不同的包里面，但是这里面也会出现一个问题，你不可能将所有的\*.class文件散着给用户。往往有可能是若干个\*.class文件完成某一个特定的功能组合，那么如果要更新也只会更新这几个文件，所以在实际的项目之中为了方便管理，往往会利用jar文件负责处理。



1. javac –d . Message.java；
2. jar –cvf my.jar org

生成的jar文件可以通过rar压缩文件打开，而后在这个文件里面会存在有一个文件夹：META-INF/MANIFEST.MF。

实际上每一个jar文件都是一个独立的路径，所以要想让jar文件起作用，必须配置CLASSPATH，但是CLASSPATH中有一个“.”，所以如果要配置多个CLASSPATH使用“；”分割。

|  |
| --- |
| SET CLASSPATH:.;C:\testjava\my.jar |

# 总结

1. 以后的程序都要写上package打包；
2. import语句往往不需要编写，前提：使用开发工具；
3. jar命令是最好用的，比开发工具里提供的好用。