Head First Java: Chapter 17 Notes

September 17, 2019

JAR Files and Pakages

Java contains the ability to organize and package compiled classes
into a single executable (providing the end-user has access to a
JVM) JAR file. There are some good practice in organizing source
code files and class files. First, it is accepted convention to put all
classes in a program into a package to avoid name-conflicts. In prac-
tice package naming often goes by reverse domain hierarchy. For
example, if the user's domain name is `headfirstjava.com`, then
the package would begin with `com.headfirstjava`. As many lev-
els can be added after that depending on what makes the most
sense. On the developments computer, the file structure of the
source code files is important. Using the `com.headfirstjava` struc-
ture as an example, and assuming the program name is `MyProject`,
the source code file(s) (that is, the .java files) should be located at
`/MyProject/source/com/headfirstjava`. Then, to compile the .java
source code files, the correct file structure can be created by using the
commend `javac -d ../classes com/headfirstjava/*.java` while
situated in `/MyProject/source/`. If compilation is successful, all class
files (including inner classes) would be located in `/MyProject/classes/com/headfirstjava/`.

The source code all related .java files shoudl begin (before any
`import` statements) with `package com.headfirstjava;` which will
ensure that all resulting class files (and later jar files) know they are
in the same package. A `manifext.txt` file should be created and
placed into `/MyProject/classes/`, with one single line: "`Main-Class:`
`com.headfirstjava.MyProjectMain`" (where `MyProjectMain` is the
program name; the class where the `main()` method is located at.)
Then, the jar file can be created by the commend line command
`jar -cvmf manifest.txt MyProject.jar com` (the name of the .jar
file can be anything) while located in the directory containing the
`manifest.txt` file.

The jar file is handy to project source code from end-user, and also
deploy without having to handle many different files. Note that the
web deployment framework described in this chapter, Java Web Start
has been discontinued since Java 11.