Objects and Classes

Part 9 – JAR Files & Documentation

Chapter 4, Core Java, Volume I

Contents

- JAR Files
- Creating JAR Files
- Manifest Files
- Executable JAR Files
- Documentation Comments
- Class and Field Comments
- Method Comments
- Other Comments
- Comment Extraction

JAR Files

- The Java™ Archive (JAR) file format enables you to bundle multiple files into a single archive file.
- Typically a JAR file contains the class files and auxiliary resources associated with applications (such as image and sound files).
- JAR files are compressed, using the familiar ZIP compression format.

- Common JAR file Operations (Using `jar` Tool)
  - Creating a JAR File
  - Viewing the Contents of a JAR File
  - Extracting the Contents of a JAR File
  - Running JAR-Packaged Software
  - etc.

Creating a JAR File

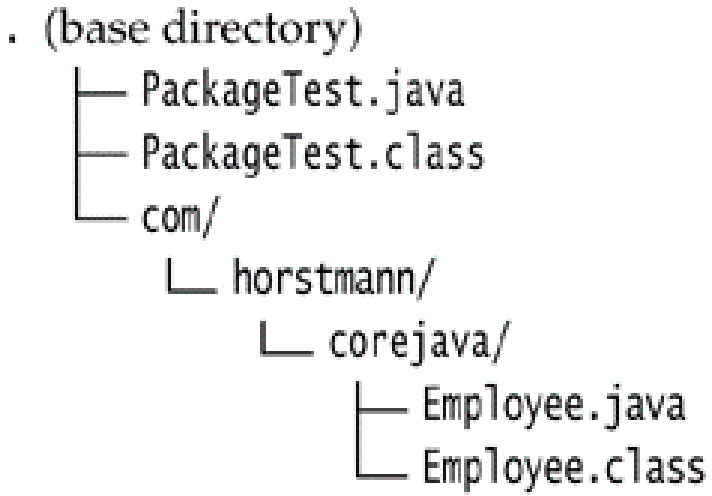
- Typical Command Syntax

  jar options jarFileName file1 file2 …

  e.g. jar cvf EmployeeTest.jar EmployeeTest.class Employee.class

- If any of the specified files are directories, the jar program processes them recursively.

  jar cvf PackageTest.jar PackageTest.class com



Manifest Files

- The manifest is a special file that can contain information about the files packaged in a JAR file.
- By tailoring this "meta" information that the manifest contains, you enable the JAR file to serve a variety of purposes (such as security, package versioning, etc).

- Manifest File (META-INF₩MANIFEST.MF) Structure
  - main section – applies to the whole jar file (e.g. Main-Class)
  - indivisual section – applies to individual files, packages, etc.(e.g. Content-Type)

- Each section can contain attribute entries represented as so-called "name-value" pairs.

  e.g. Main-Class attribute

  Main-Class : PckageTest

```
Manifest-Version: 1.0
Created-By: 1.8.0_181 (Oracle Corporation)
lines describing the archive

Name: PackageTest.class
lines describing the file
Name: com/horstmann/corejava/
lines describing this package
```

Executable JAR Files

- **JAR Files as Applications**
  - You can run JAR packaged applications with the Java command.
  - You can only specify one JAR file, which must contain all of the application-specific code.

```
java -jar PackageTest.jar
```

- Executable JAR files need to specify the entry point of the programs
  - Using jar command option

```
jar cvfe PackageTest.jar PackageTest PackageTest.class com
```

  - Editing MANIFEST-MF file

```
Manifest-Version: 1.0
Created-By: 1.8.0_181 (Oracle Corporation)
Main-Class: PackageTest
```

Do not add a .class extension!

Documentation Comments

- javadoc tool generates HTML documentation extracting information about:
  - Public classes and interfaces
  - Public and protected fields
  - Public and protected constructors, and methods
  - Packages

- Syntax for Docuementation Comments
  - Comments delimited by /** ... */ inside source files.
  - Contains free-form text which can include tags.
  - A tag starts with an @, such as @since or @param
  - The first sentence should be a summary statement.
  - Can use HTML tags such as `<em>...</em>` for formatting, and even `<img...>` to include images
  - For monospaced code, use `{@code ...}`

Class and Field Comments

```
/**
 * A {@code Card} object represents a playing card, such
 * as "Queen of Hearts". A card has a suit (Diamond, Heart,
 * Spade or Club) and a value (1 = Ace, 2 . . . 10, 11 = Jack,
 * 12 = Queen, 13 = King)
 */
public class Card
{
   /**
    * The "Hearts" card suit
    */
   public static final int HEARTS = 1;
   ...
}
```

Method Comments

- Special Tags for Method Comment
  - @param variable description
  - @return description
  - @throws exception description

```
/**
* Raises the salary of an employee.
* @param byPercent the percentage by which to raise the salary (e.g. 10 means 10%)
* @return the amount of the raise
*/
public double raiseSalary(double byPercent)
{
   double raise = salary * byPercent / 100;
   salary += raise;
   return raise;
}
```

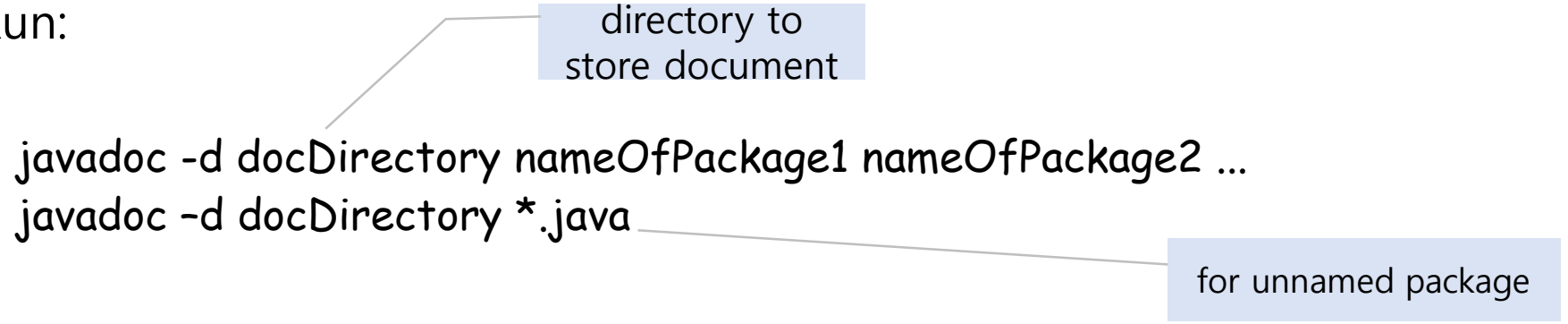
General Comments

- @author *name* *(with -author option)*
- @version *text* *(with -version option)*
- @since *text*
- @deprecated *text*
- @see *reference – adds a hyperlink in the "see also" section*
  - *reference* can be one of the following:
    - "*package.class#feature* (e.g. @see com.horstmann.corejava.Employee#raiseSalary(double))
    - ... (e.g. @see Core Java Book)
    - "text" (e.g. @see "Core Java Volume 2")
- Can include {@link *package.class#feature*} anywhere in a comment.

- Package comments : two choices
  - Supply package-info.java, containing a Javadoc comment /** ... */ preceding a package statement.
  - Supply an HTML file named package.html.

Comment Extraction

- Change to the source directory.
- Run:



- Some useful options
  - -**version** : to include @**version**
  - -**author** : to include @**author**