

Java^{SE 7} 技術手冊

- 涵蓋 OCP/JP (原 SCJP) 考試範圍
- Coin 專案、JSR166y、JDBC 4.1、NIO.2 等 Java SE 7 新功能介紹
- JDK 基礎與 IDE 操作交相對照
- 提供 Lab 檔案與操作錄影教學

CHAPTER 1

- Java平台概論



學習目標

- 簡介Java版本遷移
- 認識Java SE、Java EE、Java ME
- 瞭解JVM、JRE與JDK
- 下載、安裝JDK

Java不只是語言

- 從1995年至今，已經超過15個年頭
- 不僅是個程式語言，也代表瞭解決問題的平台（Platform）
- 代表了原廠、各個廠商、社群、開發者與使用者溝通的成果

Java誕生

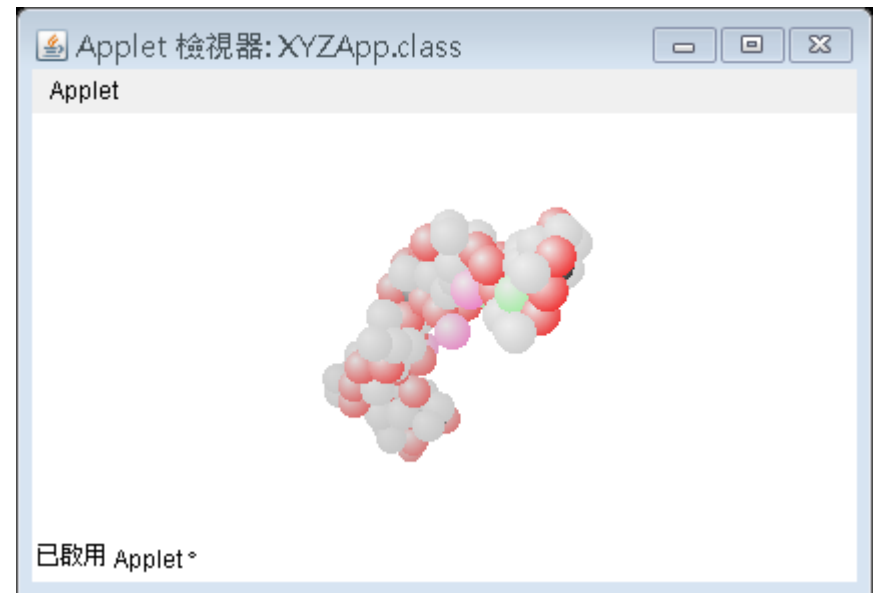
- Java最早是Sun公司「綠色專案」(Green Project) 中撰寫Star7應用程式的程式語言，當時名稱不是Java，而是取名為Oak
- 後來發現Oak名稱已經被註冊了，工程師們邊喝咖啡邊討論著新名稱，最後靈機一動而改名為Java

Java誕生

- 一開始是為了消費性數位產品而設計，而這類小型裝置通常有著有限記憶體與運算資源
 - Java本身會見到許多為了節省資源而作的設計，像是動態載入類別檔案、字串池（String pool）等特性
- 全球資訊網（World Wide Web）興起，Java Applet成為網頁互動技術代表

Java誕生

- 1995年5月23 日，正式將Oak改名為Java，Java Development Kits（當時JDK全名）1.0a2版本正式對外發表
- 1996年Netscape Navigator 2.0也正式支援Java，Microsoft Explorer亦開始支援Java



版本演進

- Sun在1998/12/4年發佈Java 2 Platform，簡稱J2SE 1.2
- Java開發者版本一開始是以Java Development Kit名稱發表，簡稱JDK，而J2SE則是平台名稱，包含了JDK與Java程式語言

版本演進

- Java平台標準版約每兩年為週期推出重大版本更新
- 「Java 2」這個名稱也從J2SE 1.2一直延用至之後各個版本

版本演進

- 2004/9/29發表的Java平台標準版的版號不是1.5，而直接跳到5.0，稱為J2SE 5.0
- 2006/12/11發表的Java平台標準版，除了版號之外，名稱也有了變化，稱為Java Platform, Standard Edition 6，簡稱Java SE 6
- JDK6全名則稱為Java SE Development Kit 6，也就是不再像以前Java 2帶有"2"這個號碼

版本	代碼名稱	釋出日期
JDK 1.1.4	Sparkler (煙火)	1997/9/12
JDK 1.1.5	Pumpkin (南瓜)	1997/12/3
JDK 1.1.6	Abigail (聖經故事人物名稱)	1998/4/24
JDK 1.1.7	Brutus (羅馬政治家名稱)	1998/9/28
JDK 1.1.8	Chelsea (足球俱樂部名稱)	1999/4/8
J2SE 1.2	Playground (遊樂場)	1998/12/4
J2SE 1.2.1	無	1999/3/30
J2SE 1.2.2	Cricket (蟋蟀)	1999/7/8
J2SE 1.3	Kestrel (紅隼)	2000/5/8
J2SE 1.3.1	Ladybird (瓢蟲)	2001/5/17
J2SE 1.4.0	Merlin (魔法師名稱)	2002/2/13
J2SE 1.4.1	Hopper (蚱蜢)	2002/9/16
J2SE 1.4.2	Mantis (螳螂)	2003/6/26
J2SE 5.0	Tiger (老虎)	2004/9/29
Java SE 6	Mustang (野馬)	2006/12/11
Java SE 7	Dolphin (海豚)	2011/7/28

江山易主

- 從Java SE 6之後，Java開發人員足足等了四年多，才等到新版本的推出 ...
- 2010年Oracle宣佈併購Sun ...
- 2010年底JCP（Java Community Process，稍後即會說明這個組織為何）終於通過了Java SE 7與Java SE 8的規劃地圖（Roadmap） ...
- Java SE 7正式於2011/7/28釋出

三大平台

- Sun公司在1999年6月美國舊金山的Java One大會上，公佈了新的Java體系架構
 - J2SE (Java 2 Platform, Standard Edition)
 - J2EE (Java 2 Platform, Enterprise Edition)
 - J2ME (Java 2 Platform, Micro Edition)

三大平台

- Java SE 6後Java不再帶有“2”這個號碼
 - J2SE → Java SE
 - J2EE → Java EE
 - J2ME → Java ME

Java SE



JCP與JSR

- Java不僅是程式語言，而是標準規範！
- Sun公司於1998年組成了JCP（Java Community Process）
 - 一個開放性國際組織，目的是讓Java演進由Sun非正式地主導，成為全世界數以百計代表成員公開監督的過程

JCP與JSR

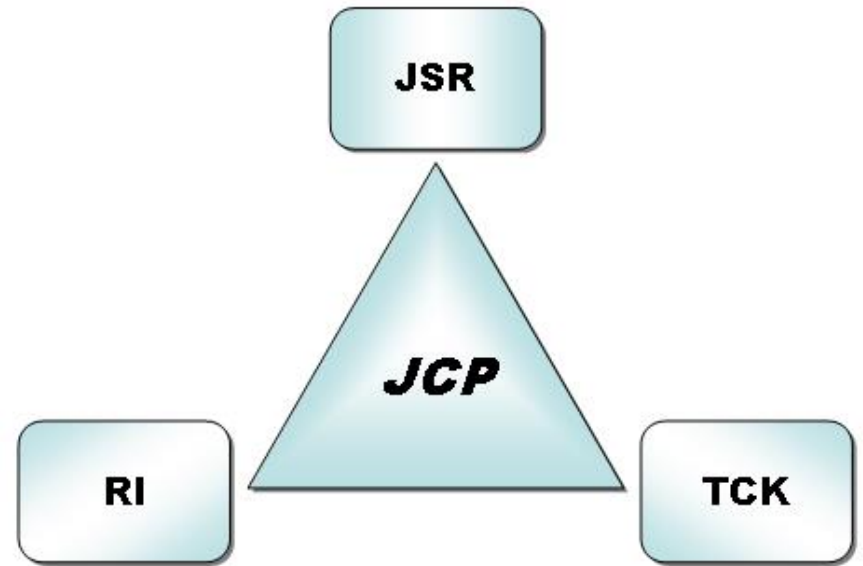
- 任何想要提議加入Java的功能或特性，必須以JSR（Java Specification Requests）正式文件的方式提交
- JSR必須經過JCP執行委員會（Executive Committee）投票通過，方可成為最終標準文件
- 有興趣的廠商或組織可以根據JSR實現產品

JCP與JSR

- 若JSR成為最終文件後，必須根據JSR實作出免費且開發原始碼的參考實現，稱為RI（Reference Implementation）
- 提供TCK（Technology Compatibility Kit）作為技術相容測試工具箱
 - 方便其它想根據JSR實現產品的廠商或組織參考與測試相容性

JCP與JSR

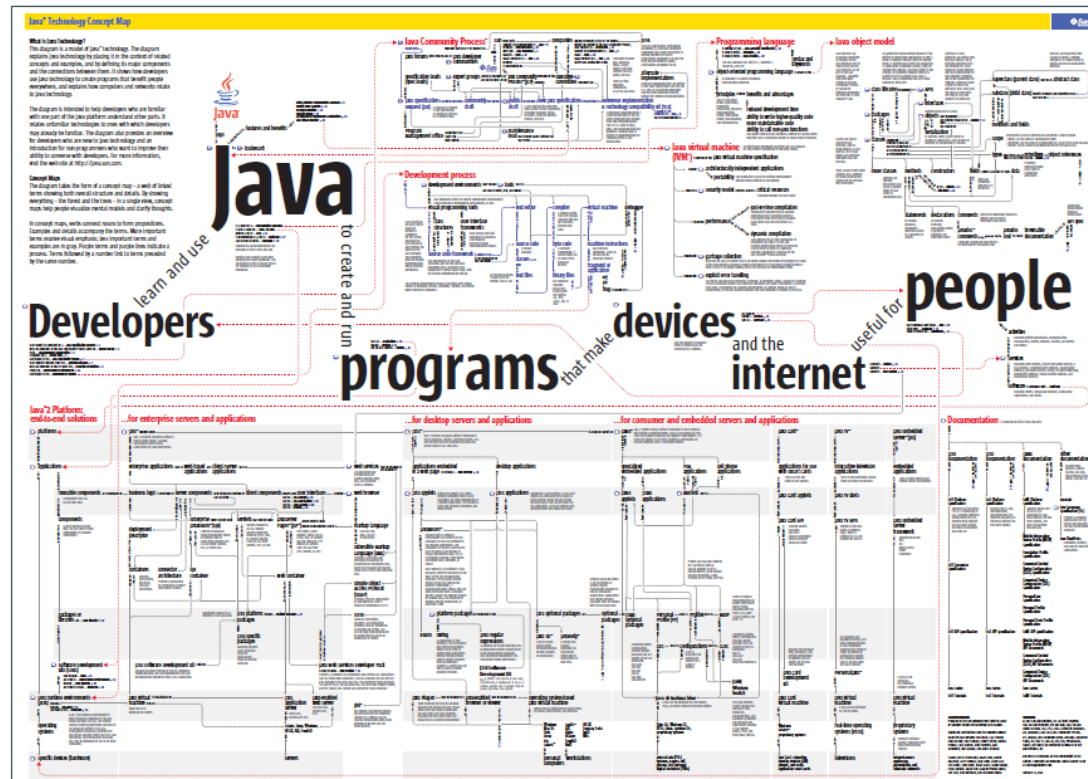
- 無論是Java SE、Java EE或Java ME，都是業界共同訂製的標準
- JSR作為正式標準規範文件，不同的技術解決方案標準規範會給予一個編號
- 在JSR規範的標準之下，各廠商可以各自實作成品



JCP與JSR

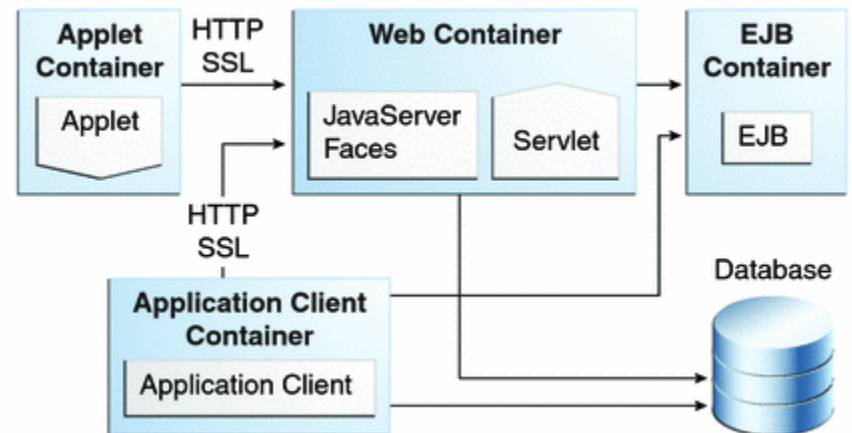
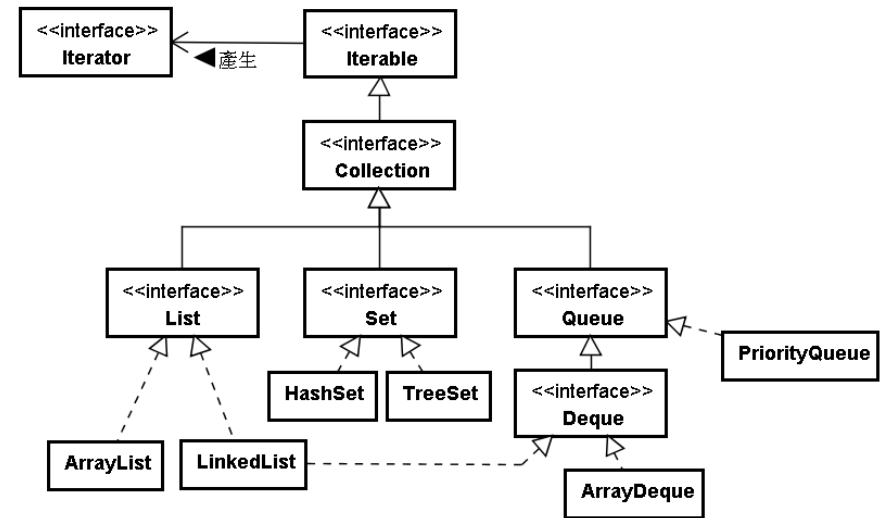
- Java SE 7的主要規範是在JSR 336文件之中，而Java SE平台中的特定技術，則再規範於特定的JSR文件之中...
 - <http://jcp.org/en/jsr/detail?id=336>

- <http://java.sun.com/new2java/javamap/intro.html>



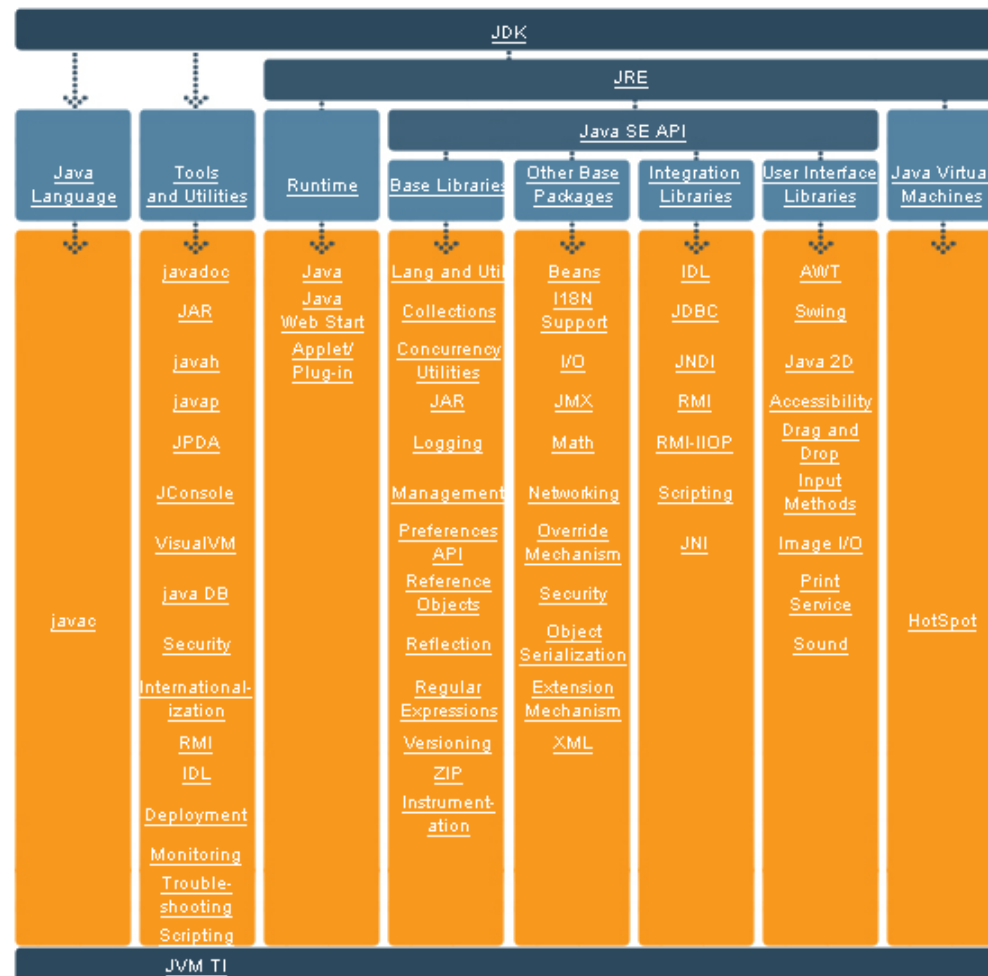
建議的學習路徑

- 深入瞭解JVM/JRE/JDK
- 理解封裝、繼承、多型
- 掌握常用Java SE API架構
- 學習容器觀念
- 研究開放原始碼專案
- 學習設計模式與重構
- 熟悉相關開發工具



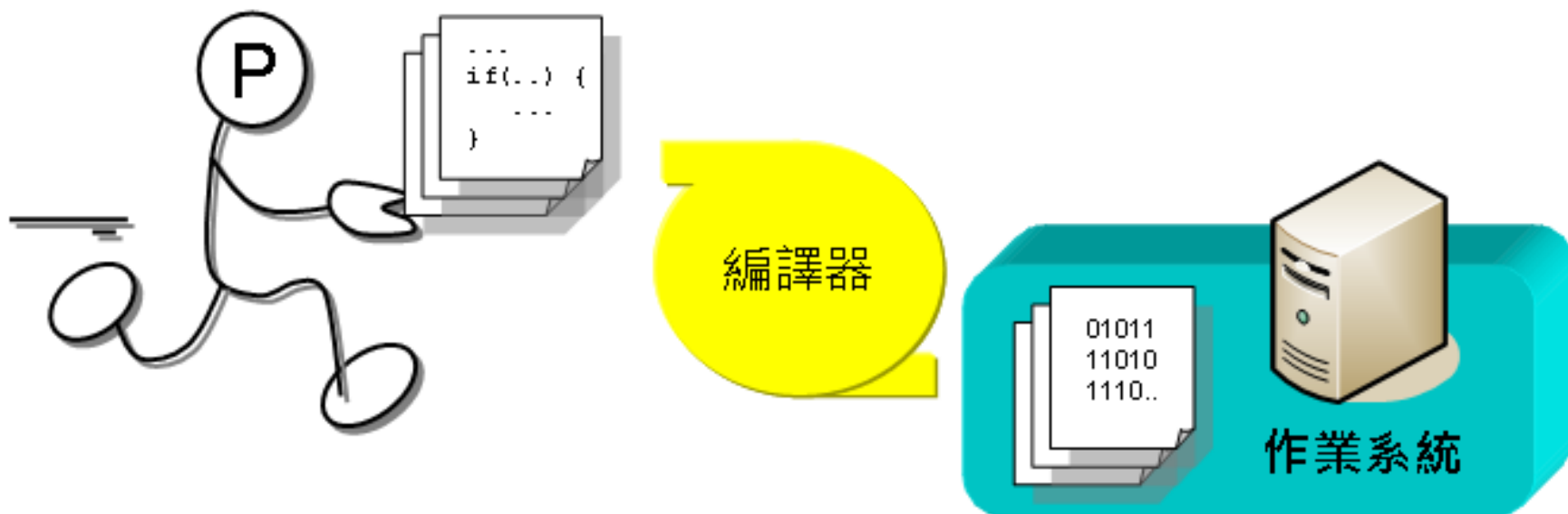
JVM/JRE/JDK

Java SE Platform at a Glance

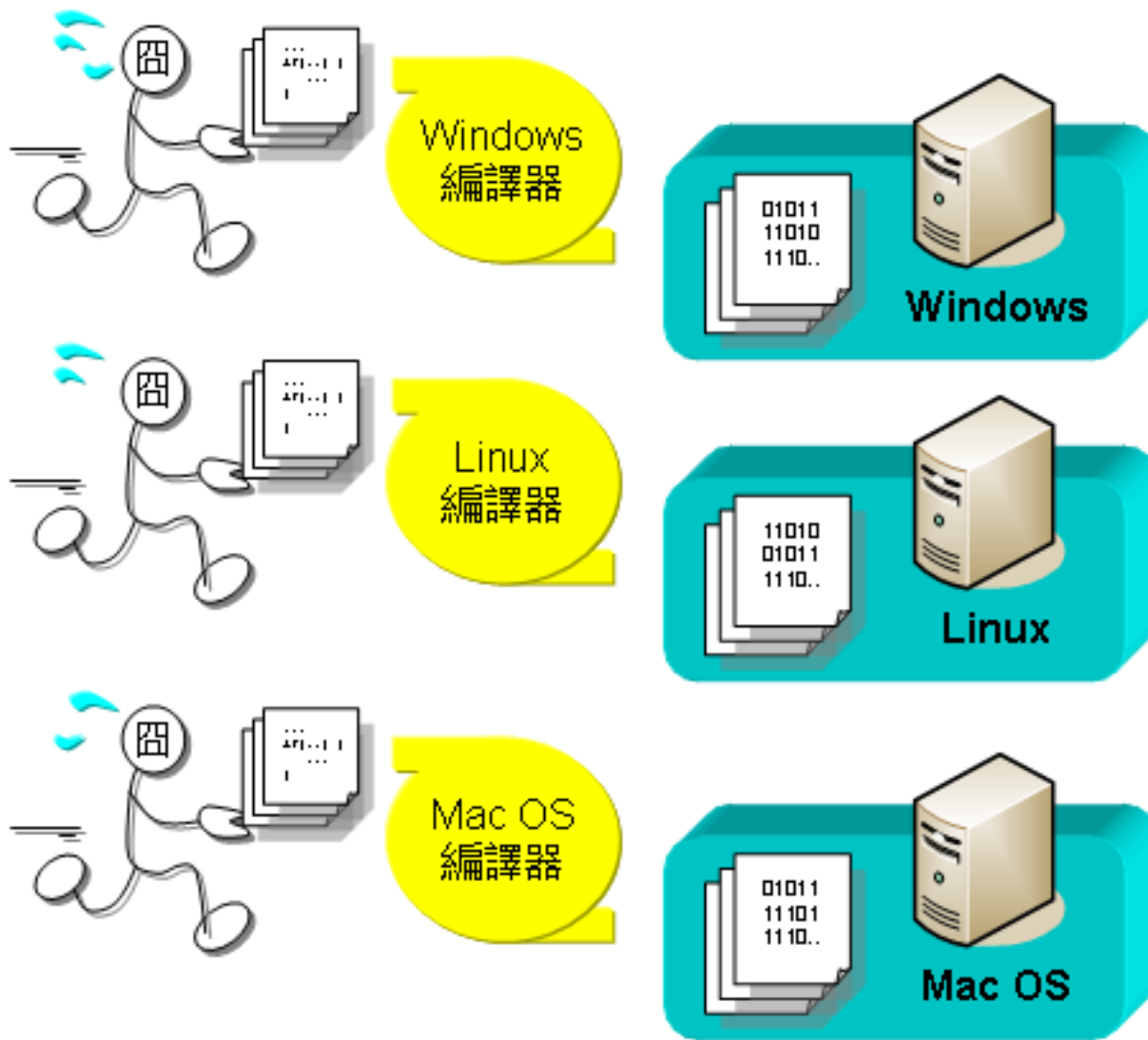


什麼是JVM？

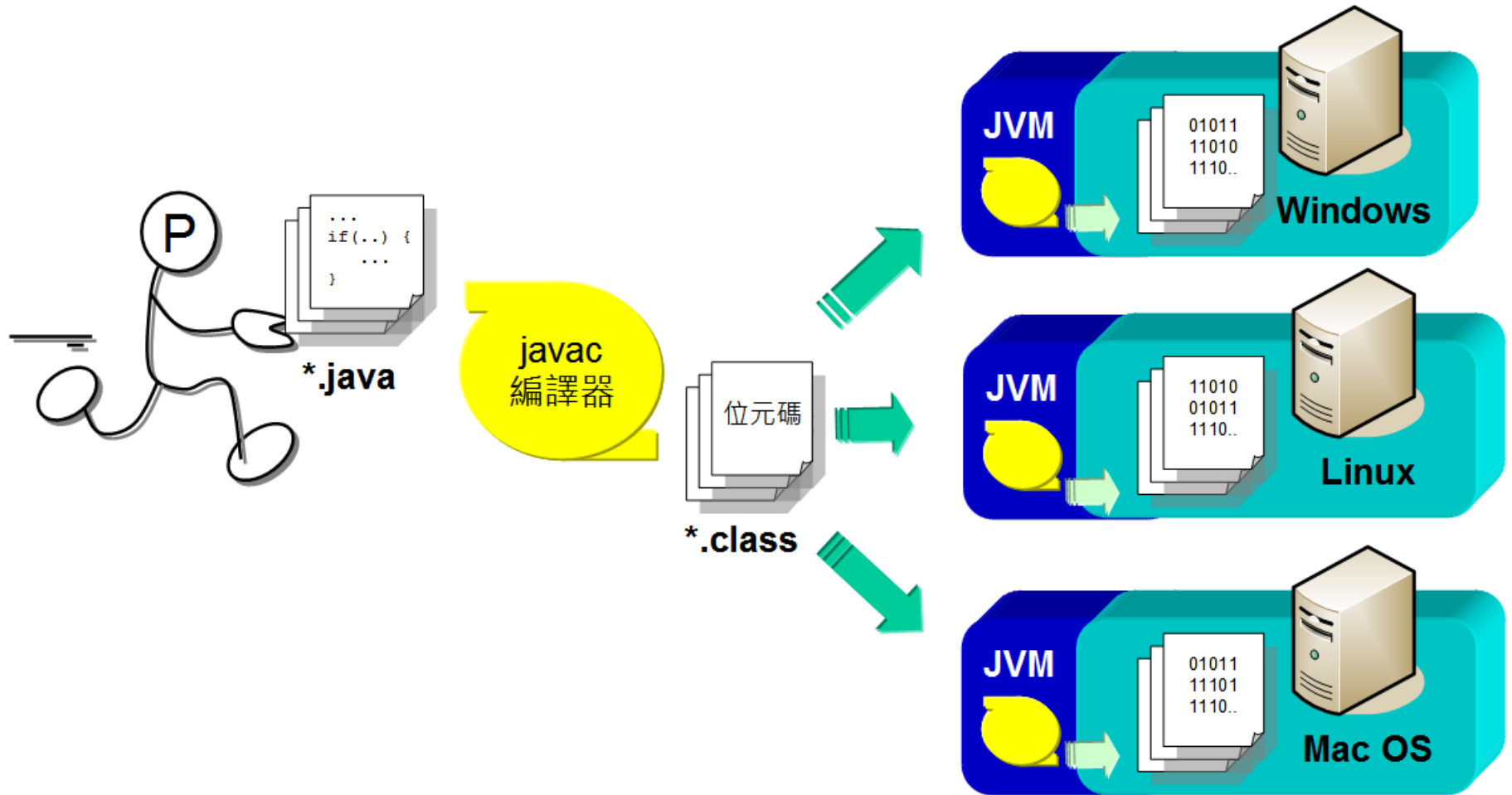
- JVM讓Java可以跨平台，但是跨平台是怎麼一回事？



什麼是JVM？



什麼是JVM？



什麼是JVM？

- 對Java程式而言，只認識一種作業系統，這個系統叫JVM，位元碼檔案（副檔名為.class的檔案）就是JVM的可執行檔。
 - Java程式理想上，並不用理會真正執行於哪個平台，只要知道如何執行於JVM就可以了
 - JVM實際上就相當於Java程式的作業系統，JVM就負責了Java程式的各種資源管理

區分JRE與JDK

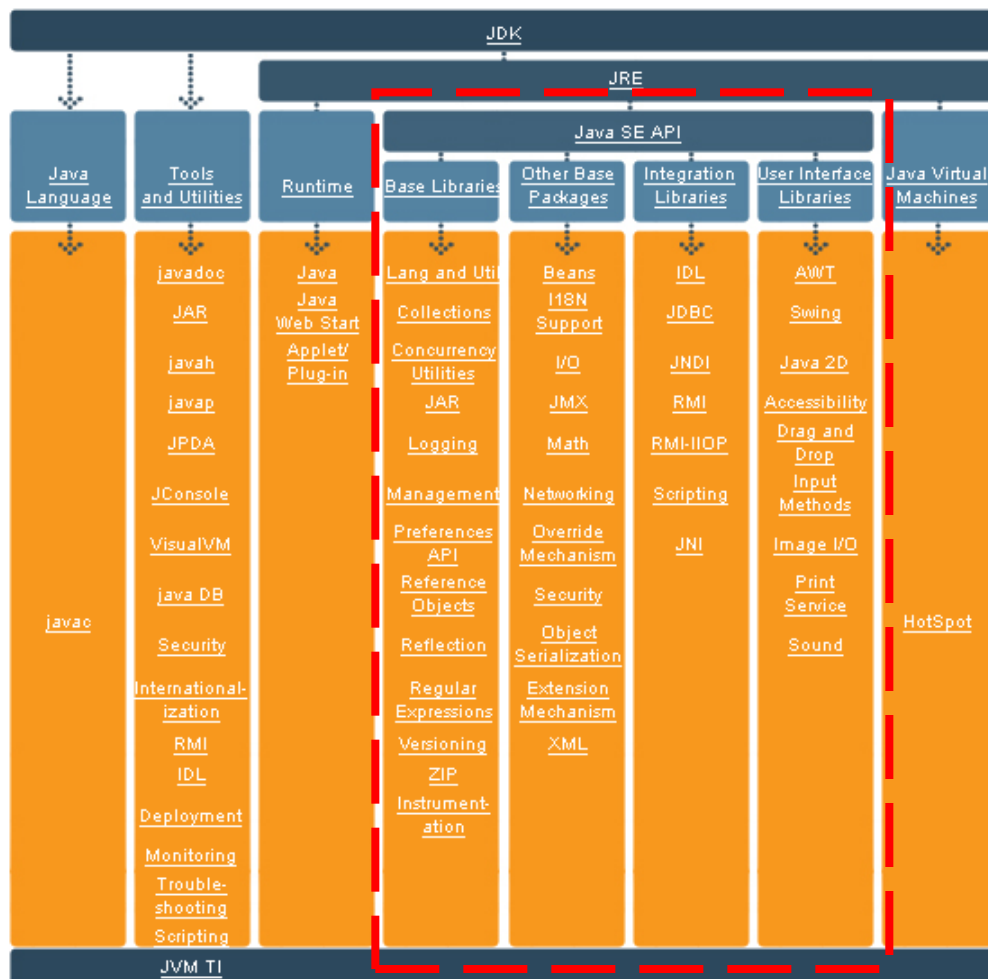
```
System.out.println("Hello World");
```

- System、out、println這些名稱，都是標準中規範的名稱
- 實際上必須要有人根據標準撰寫出System.java，編譯為System.class
- 你才能在撰寫第一個Java程式時，使用System類別（Class）上out物件（Object）的println（）方法（Method）

區分JRE與JDK

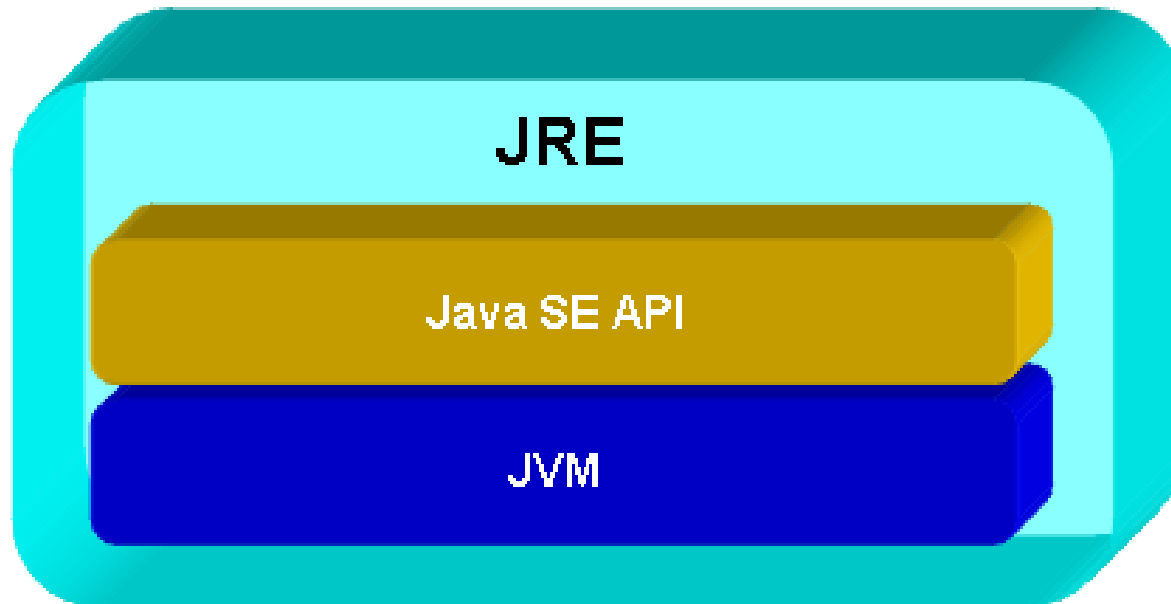
- 廠商必須根據相關的JSR標準文件，將標準程式庫實作出來
- 如此你撰寫的第一個Java程式，在Sun、IBM、Apache等廠商實作的JVM上運行時，引用如System這些標準API，你的第一個Java程式，才可能輕易地運行在不同的平台

Java SE Platform at a Glance



區分JRE與JDK

- Java Runtime Environment就是Java執行環境，簡稱JRE，包括了Java SE API與JVM

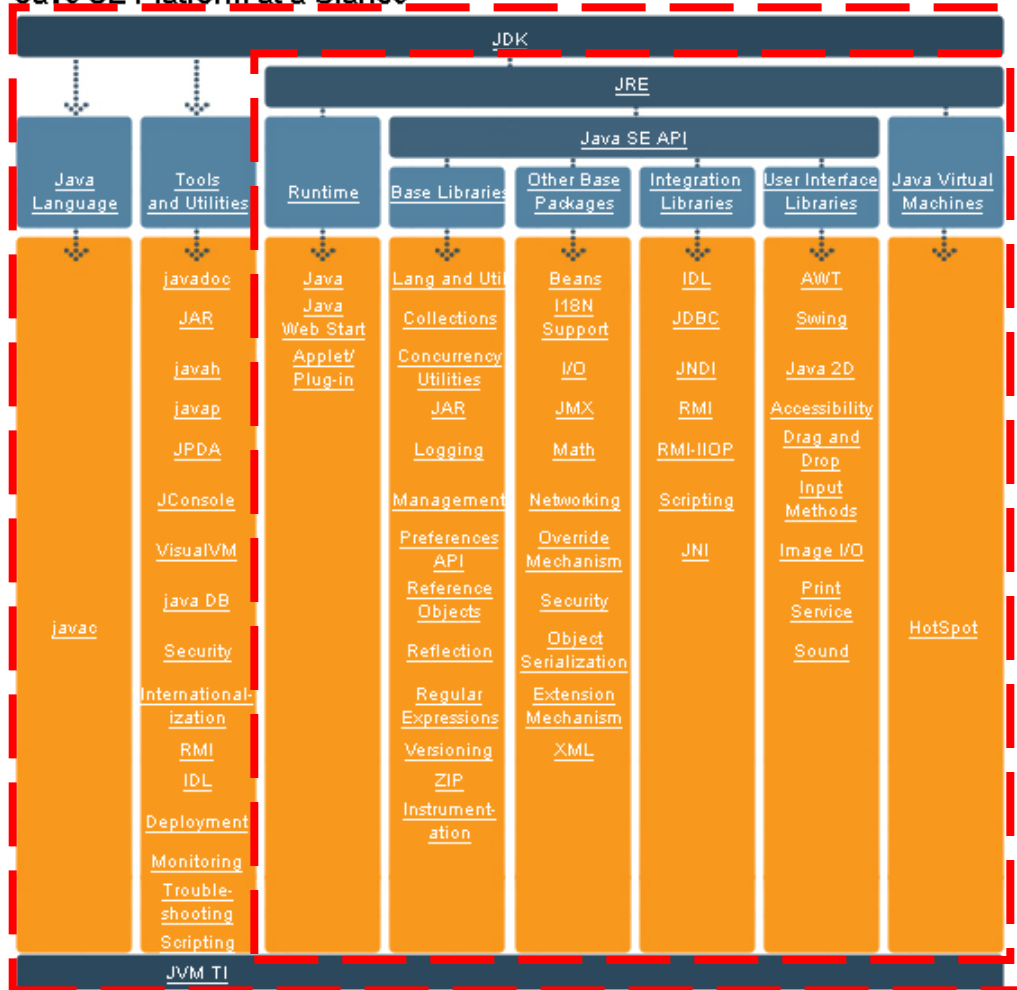


區分JRE與JDK

- 像編譯器這樣的工具程式是由誰提供？答案就是JDK，全名為~~Java Developer Killer~~！呃！不對！是Java Development Kit！

區分JRE與JDK

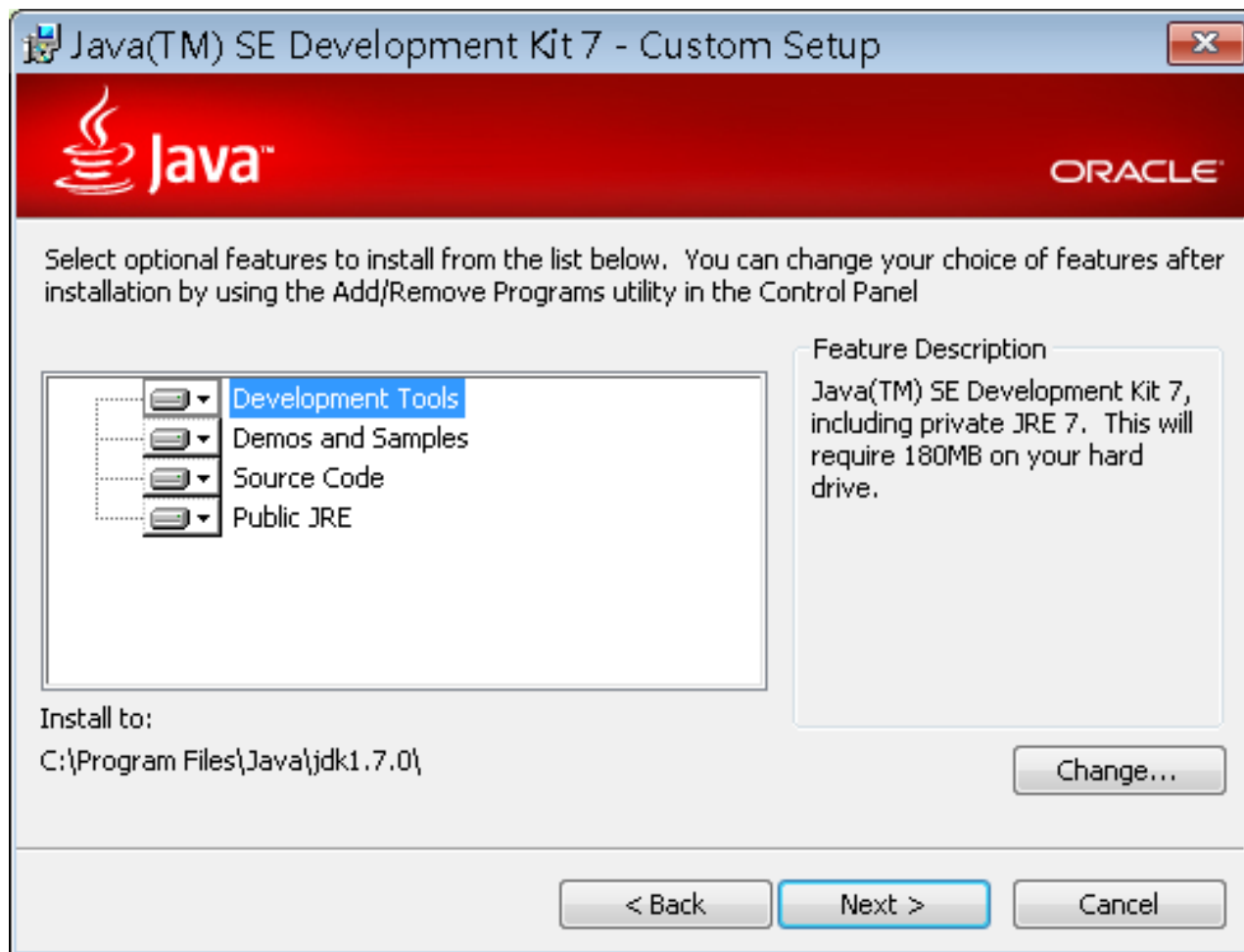
Java SE Platform at a Glance



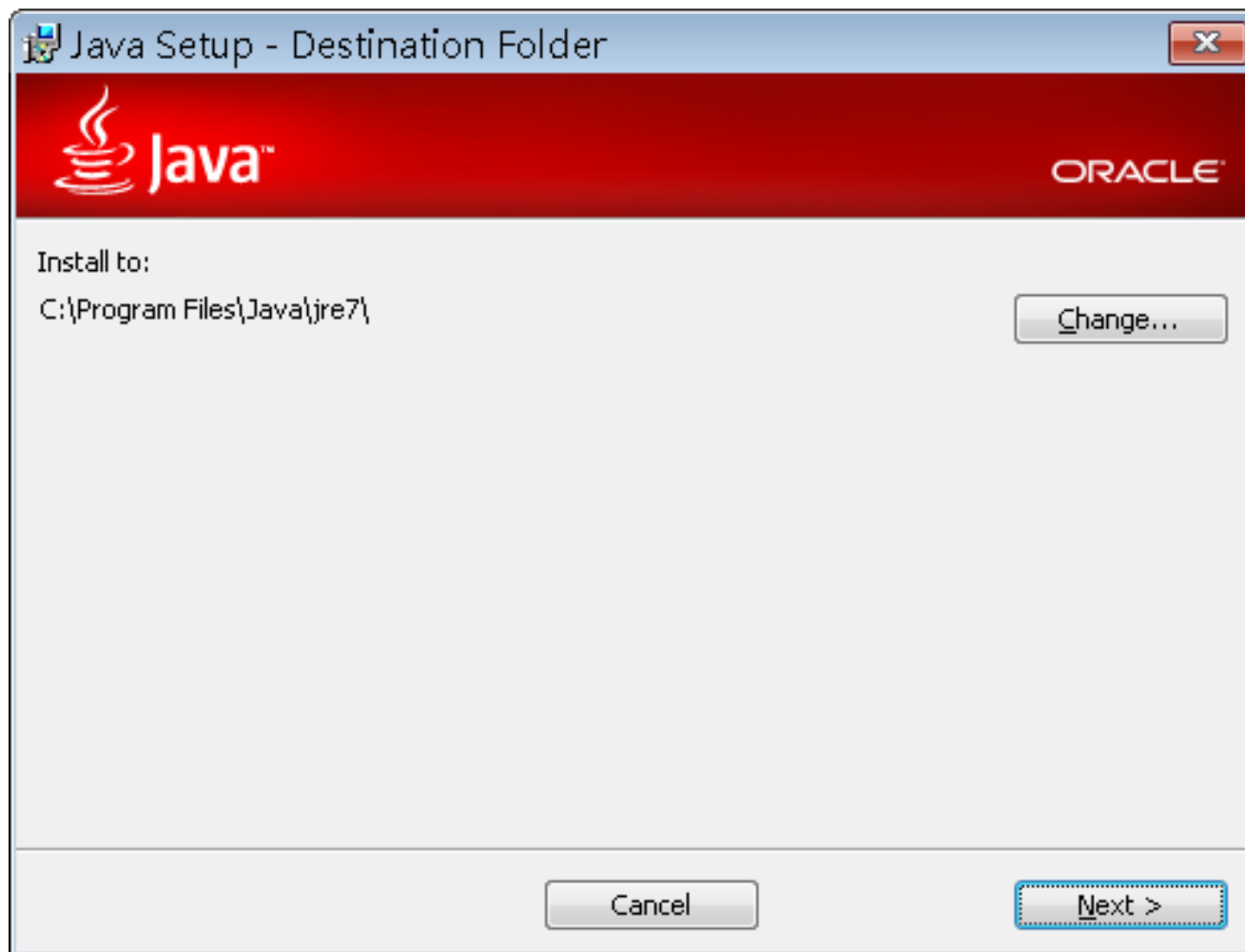
區分JRE與JDK

- JDK包括了Java程式語言、工具程式與JRE
- JRE則包括了部署技術、Java SE API與JVM
- 撰寫Java程式才需要JDK，如果你的程式只是想讓朋友執行？那他只要裝JRE就可以了

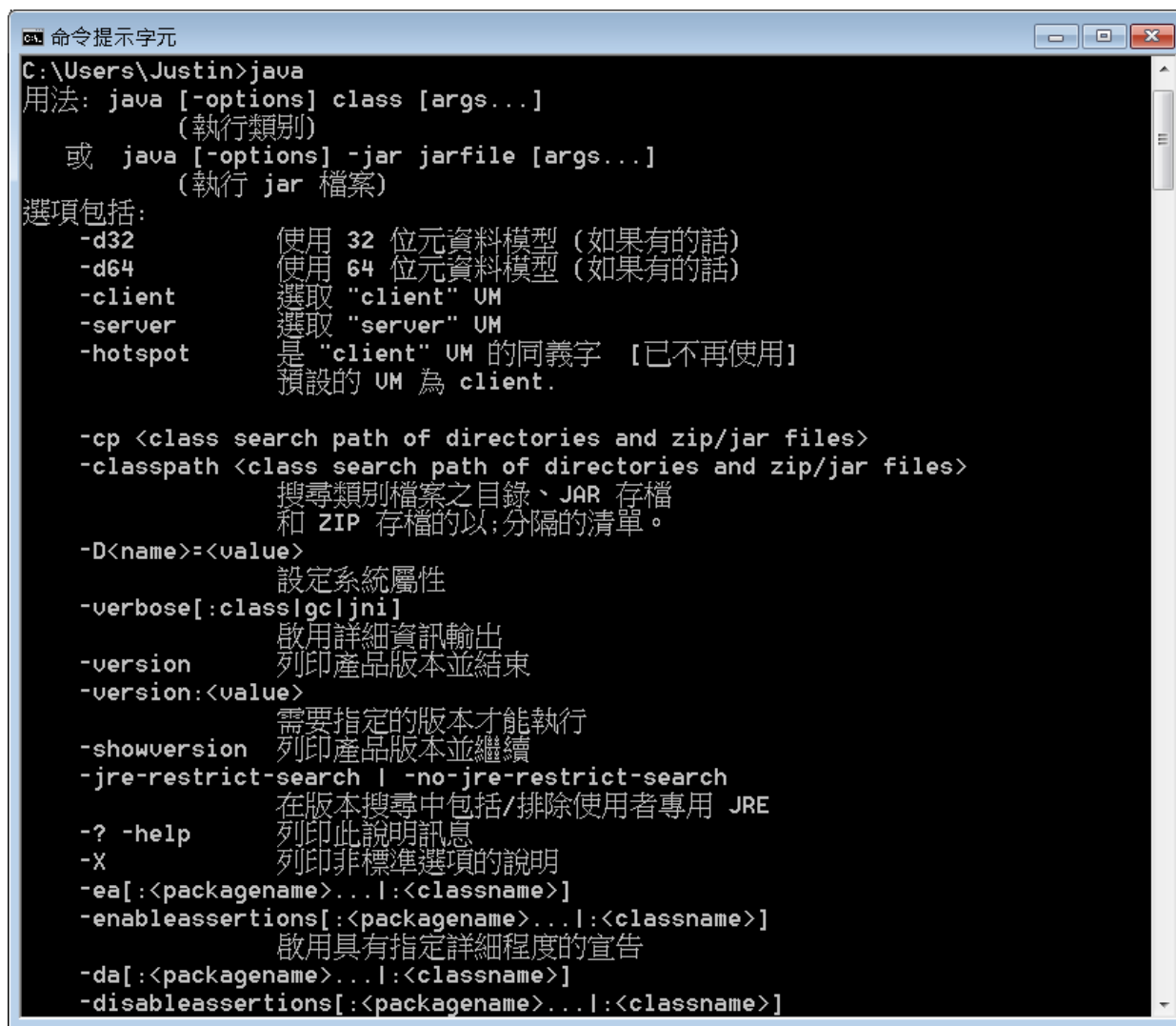
下載、安裝JDK



下載、安裝JDK



下載、安裝JDK



```
命令提示字元
C:\Users\Justin>java
用法: java [-options] class [args...]
       (執行類別)
或 java [-options] -jar jarfile [args...]
       (執行 jar 檔案)

選項包括:
  -d32          使用 32 位元資料模型 (如果有的話)
  -d64          使用 64 位元資料模型 (如果有的話)
  -client       選取 "client" VM
  -server       選取 "server" VM
  -hotspot      是 "client" VM 的同義字 [已不再使用]
                預設的 VM 為 client.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
                搜尋類別檔案之目錄、JAR 存檔
                和 ZIP 存檔的以;分隔的清單。
  -D<name>=<value>
                設定系統屬性
  -verbose[:class|gc|jni]
                啟用詳細資訊輸出
  -version      列印產品版本並結束
  -version:<value>
                需要指定的版本才能執行
  -showversion  列印產品版本並繼續
  -jre-restrict-search | -no-jre-restrict-search
                在版本搜尋中包括/排除使用者專用 JRE
  -? -help      列印此說明訊息
  -X            列印非標準選項的說明
  -ea[:<packagename>...[:<classname>]]
  -enableassertions[:<packagename>...[:<classname>]]
                啟用具有指定詳細程度的宣告
  -da[:<packagename>...[:<classname>]]
  -disableassertions[:<packagename>...[:<classname>]]
```

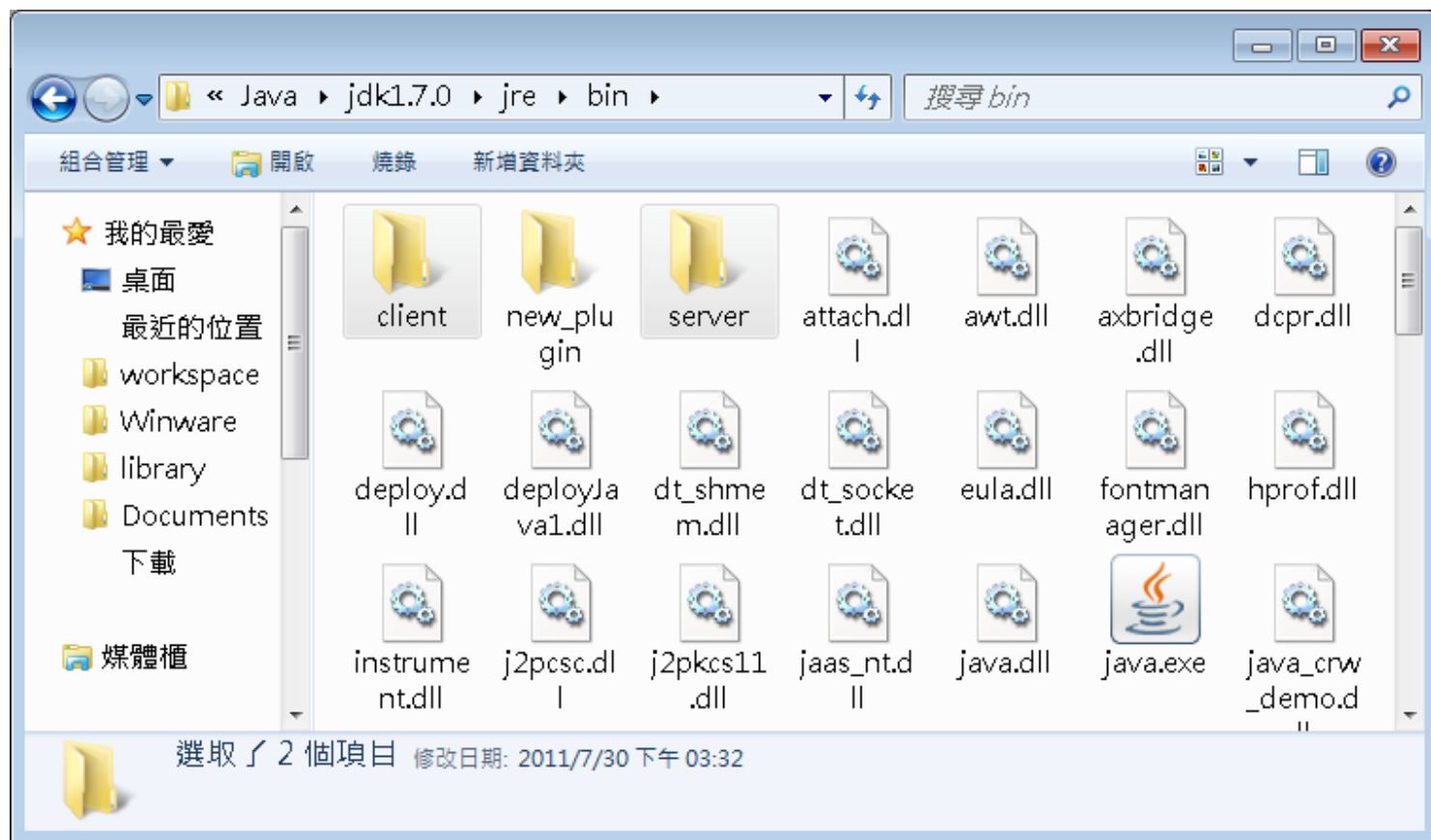
認識JDK安裝內容

- 假設JDK與Public JRE各安裝至「C:\Program Files\Java\jdk1.7.0\」及「C:\Program Files\Java\jre7\」

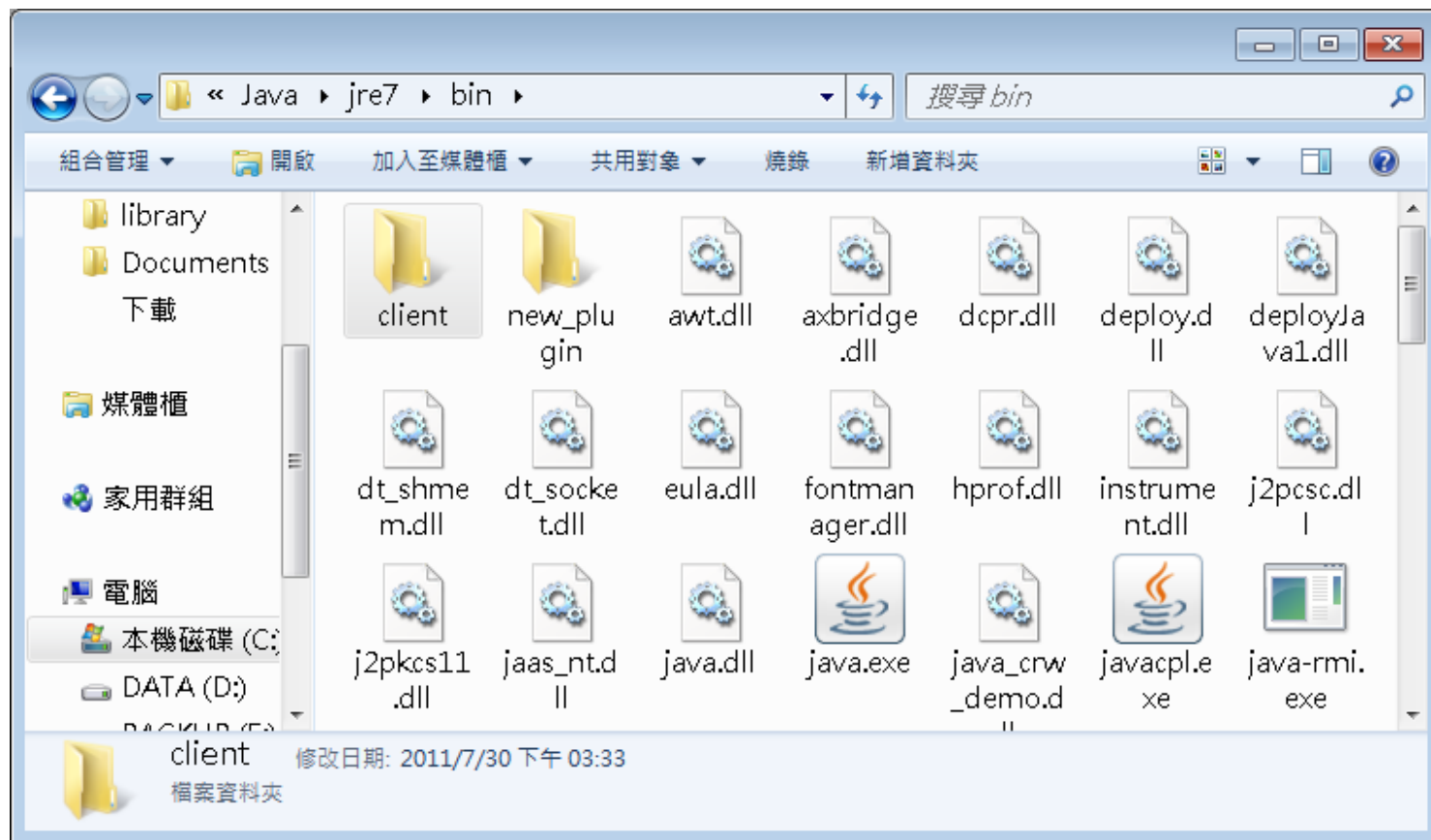
認識JDK安裝內容

- Public JRE是給Java程式執行的平台
- JDK本身也附帶JRE，這個JRE是位於JDK安裝資料夾的「jre」資料夾下，也就是在「C:\Program Files\Java\jdk1.7.0\jre」中，通常稱為Private JRE，主要是開發Java程式時測試之用

認識JDK安裝內容



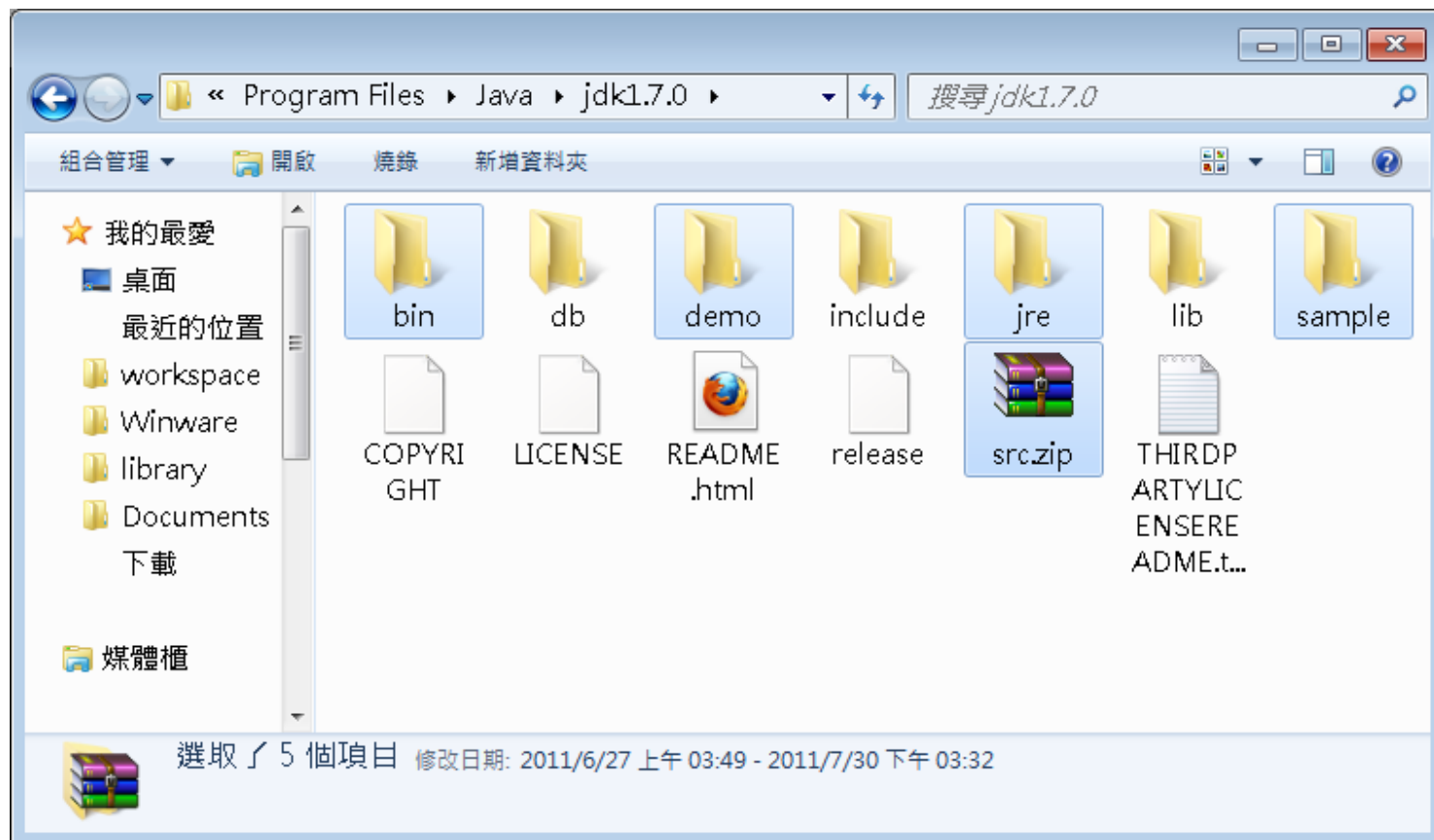
認識JDK安裝內容



認識JDK安裝內容

- server與client選項的差別在於使用的VM不同
 - 執行Java客戶端程式時預設會使用client VM，使用較少記憶體以取得較快的啟動速度（因為客戶端程式，通常想要有較快的畫面回應）
 - 使用server VM的話會花比較長的啟動時間及耗用較多的記憶體，為的是啟動Java程式後可以獲得較好的執行效能

認識JDK安裝內容



認識JDK安裝內容

- 那麼Java SE API編譯好的.class檔放在哪呢？
 - 無論是Private JRE或Public JRE的資料夾中，都會有個lib目錄，其中會有個**rt.jar**檔案，JAR（Java Archive）檔案是zip壓縮格式，可以使用解壓縮軟體開啟，你就會看到許多編譯好的.class檔案