

物件導向程式設計 使用Java Language

Prof. Hsin-Mao Huang (黃信貿)
Dept. of Computer & Information Science
Aletheia University
Taipei, Taiwan, ROC
Fall 2016

Course Overview



- Instructor: Prof. Hsin-Mao Huang (黃信貿)
 - Office Room: 909
 - Email: xmhuang@mail.au.edu.tw
 - Phone#: 26212121 – 3212
 - Office Hours: 星期一：5,6,7節；星期二：5,6,7節；星期三：6,7節
- Course Website:
 - ILms課程網站

Text Book



- 書名：程式邏輯訓練
入門與運用 使用 JAVA
SE8
- 作者：李春雄
- 出版社：上奇資訊

程式邏輯訓練 入門與運用
使用 **JAVA SE 8**



Course Overview

- Grading (tentative)
 - 40% Homework & Programming
 - 60% Midterm & final exams
- What is your duty?
 - Read the book, do the homework, and attend the class
 - Ask for any questions...

學習程式設計的技巧

- 瞭解語法
- 親自打課本上的範例並且瞭解程式內容
 - 不是一行一行的看懂，而是把整個程式在做什麼事情看懂
- 嘗試的修改範例的參數並預期產生的結果
 - 如果預期錯誤在仔細想想那裡想錯了，直到瞭解程式內容的意義
- 嘗試將範例結合
- 學程式沒有捷徑，練習練習再練習...

簡介 (Introduction)



- 程式撰寫的生命週期分成下列幾個步驟：
 1. 分析(Analysis)問題
 2. 找出解決問題的步驟，這些步驟也就是解決問題的演算法(Algorithm)
 3. 將每個步驟用程式碼表達出來並且要合乎此程式語言的語法規則，稱為程式撰寫(Programming)
 4. 執行，看看能不能正確解出問題。這步驟稱為測試(Testing)
 5. 執行結果如果不正確，則回到第三個步驟，檢查原先撰寫的程式是否有邏輯上的錯誤並加以修正，一直到程式的執行結果正確為止。這個步驟稱為除錯(Debug)

簡介 (Introduction)

- 語言

- 人類拿來互通訊息及溝通的方式。也是文化傳承與累積的重要因素。
 - 華文語系、英語語系、拉丁語系...等等

- 電腦語言

- 電腦可以判讀語言，換句話說，為電腦看的懂得語言
- 這些語言都是由010101所組合而成的
 - For example: Hello.. \leftrightarrow 48 65 6C 6C 6F 2E 2E
 - 電腦實際上處理的資料格式是01001000 01100101 01101100 01101100 01101111 00101110 00101110

簡介 (Introduction)



- 程式語言

- 程式語言語法為人類語言語法相近，所以可讀性高
- 電腦無法瞭解，無法執行
- 必須透過編譯器(Compiler)將程式編譯成機械碼才可在電腦執行。

Java 程式

```
#import java.io.*;
public class test {
    public static void main(String args[])
    {
        System.out.println("Hello...");
    }
}
```

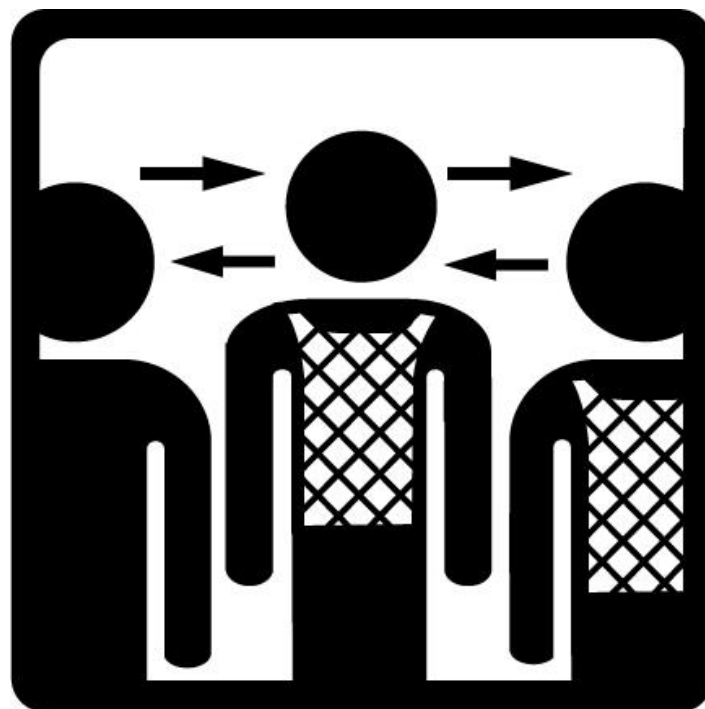
編譯器

執行檔

```
0101010101010100101010100101010
1111110101001010101010101010101
1010010101001010100101001010100
0001010100111100101010001001001
1110000011010001101010101010101
0101010010001001010010010001001
0101001010010101001010010101001
```


簡介 (Introduction)

- 編譯器在程式語言與電腦語言間所扮演的角色與翻譯官相同。



Java的沿革 (Java History)

- Java為美國昇陽(Sun Microsystems)公司所開發的程式語言
- Java語言起源於1990年12月的一個內部計畫
 - 此計畫目的提供替代C++/C程式語言，但是其API並不適用，工程師Patrick Naughton決定改用NeXT。同時Patrick獲得參與了Stealth計畫的工作並把此技術帶入此計畫中。
 - 不久Stealth計畫改名為Green計畫，並且加入了James Gosling和Mike Sheridan。
 - Green計畫的目的是要開發下一代智慧應用程式技術，企圖在家電市場佔據主要地位。

Java的沿革 (Java History)

- Green計畫

- 原先考慮使用C++，然而有下列幾個原因捨棄C++

- 此計畫主要開發嵌入式系統程式，因此具有有限資源的限制，然而C++太耗資源
 - C++太複雜導致開發者經常發生錯誤
 - C++缺乏記憶體垃圾回收系統，導致開發者必須手動管理系統記憶體，這是個很大的挑戰而且容易出錯
 - C++也缺乏移動式設備的安全性、分散式程式設計和多執行緒等功能

Java的沿革 (Java History)

- Oak的由來

- 為了擺脫C++的困擾，Bill Joy設想結合Mesa和C發展新的程式語言。並且他建議以C++為基礎發展物件導向環境
- 最初，Gosling企圖修改和延伸C++(他稱為C++ ++ --)但很快就放棄並且佔成發展全新的程式語言，稱為Oak。以他辦公室外的橡樹(Oak)命名
- 1992年夏天已可以展示Green OS、Oak language、程式庫和硬體
- 9/3 1992 應用Oak在稱為Star 7的PDA上
 - 具有圖形化介面和智慧代理程式” Duke”
- 1992年11月，Green計畫變成一家名為FirstPerson公司，為Sun的子公司

Java的沿革 (Java History)

- Oak的失敗
 - FirstPerson公司參與時代華納機上盒(set-top box)徵求提議書。FirstPerson使用Oak發展set-top box
 - 有線電視業者任為FirstPerson公司開發的set-top box平台給了用戶太多控制權，因此敗於SGI
 - 此平台無法產生利益最後並回Sun

Java的沿革 (Java History)

- 隨著Internet的興起與Mosaic瀏覽器的出現，原先團隊在1994年6、7月間三天的腦力激盪下決定改變目標，將原先應用在嵌入式系統的Oak技術應用在Internet上
- 商標註冊時發現Oak已被註冊，因此將Oak改名為Java

Java語言的特性

- 簡單 (Simple)
 - 架構簡單、易學的程式語言
 - 就語法而言，Java是一個『純淨』的C++語言
 - 捨棄運算子多載(operator overloading)、多重繼承(multiple inheritance)...等功能
- 物件導向 (Object-Oriented)
- 網路功能 (Network-Savvy)
- 強韌 (Robust)
 - Java程式穩定性高，在編譯時，Java編譯器會嚴格檢查程式碼，找出可能產生的錯誤。產生在各種不同的環境下能夠穩定執行的程式。

Java語言的特性

- 安全性 (Secure)
 - Java被設計用於分散式網路程式開發，因此Java對安全性格外要求
 - 採用公開金鑰加密方式(public key encryption)
- 跨平台 (Architecture Neutral)
 - Java既然被設計成能夠網路程式開發，因此必須能用於不同類型之CPU和OS中
 - 使用Bytecode來達到跨平台的目的
- 程式直譯 (Interpreted)
 - 在電腦上安裝Java Virtual Machine(JVM)，透過JVM的直譯器來執行Bytecode檔。
 - 利用just in time編譯器來提高執行效率

Java語言的特性

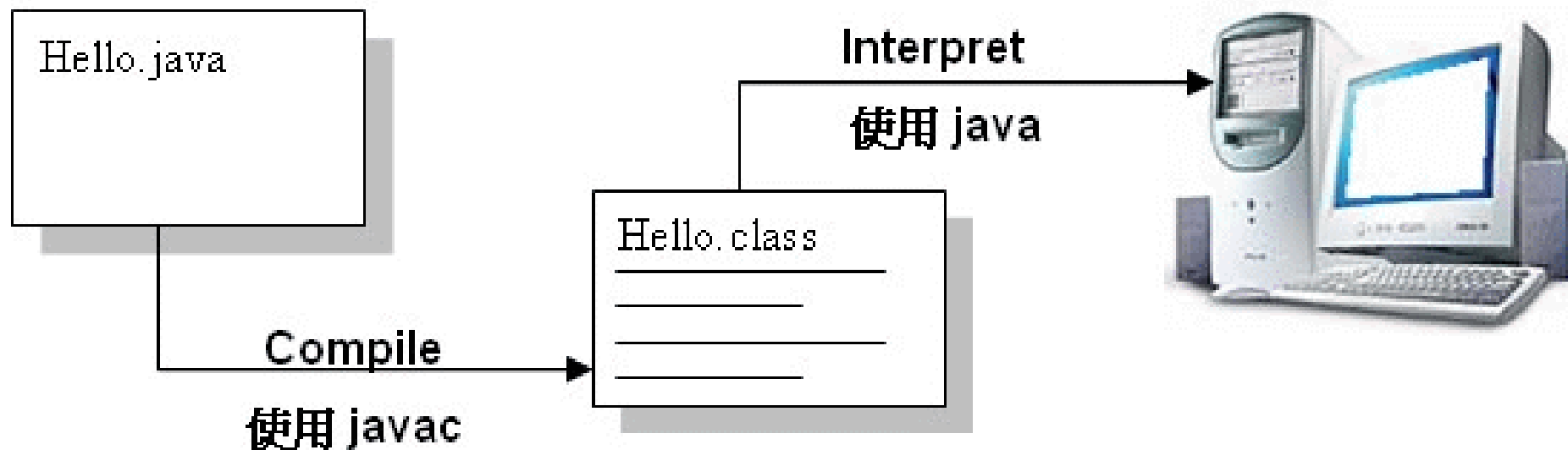
- 高效率 (High-performance)
 - Bytecode的執行效率並不比C與C++來的差
- 多執行緒 (MultiThread)
 - 簡單的程式設計方式即可容易的達到同時執行多項工作

Java執行方式

- 程式執行方式分為兩種：
 - 編譯器：程式撰寫完後需要透過編譯器將程式碼轉譯成機器碼方可執行。例如：C、C++...等
 - 優點：在編譯時可以最佳化，進而提高執行效率
 - 缺點：每次修改程式碼必須重新編譯
 - 直譯器：程式執行時將程式碼一行一行的由直譯器解譯成機器碼在執行。例如：Basic、ASP...等
 - 優點：程式修改可直接執行
 - 缺點：執行效能差

Java執行方式

- Java的執行過程包含了編譯器和直譯器



又稱bytecode檔)

- 一直譯器把中介碼檔案的程式碼直譯成可執行的機器碼。

Java執行方式



撰寫JAVA程式

```
//這是JAVA的第一個程式
public class welcome{
    public static void main(String[] args){
        System.out.println("歡迎使用JAVA");
    }
}
```

*.class檔,位元組檔Bytecode

```
00000000h: CA FE BA BE 00 00 00 31 00 1D 0A 00 06 00 0F 09 ;
00000010h: 00 10 00 11 08 00 12 0A 00 13 00 14 07 00 15 07 ;
00000020h: 00 16 01 00 06 3C 69 6E 69 74 3E 01 00 03 28 29 ;
00000030h: 56 01 00 04 43 6F 64 65 01 00 0F 4C 69 6E 65 4E ;
00000040h: 75 6D 62 65 72 54 61 62 6C 65 01 00 04 6D 61 69 ;
00000050h: 6E 01 00 16 28 5B 4C 6A 61 76 61 2F 6C 61 6E 67 ;
00000060h: 2F 53 74 72 69 6E 67 3B 29 56 01 00 0A 53 6F 75 ;
00000070h: 72 63 65 46 69 6C 65 01 00 0C 77 65 6C 63 6F 6D ;
00000080h: 65 2E 6A 61 76 61 0C 00 07 00 08 07 00 17 0C 00 ;
00000090h: 18 00 19 01 00 10 E6 AD A1 E8 BF 8E E4 BD BF E7 ;
000000a0h: 94 A8 4A 41 56 41 07 00 1A 0C 00 1B 00 1C 01 00 ;
000000b0h: 07 77 65 6C 63 6F 6D 65 01 00 10 6A 61 76 61 2F ;
```

執行Java程式

1-2 Java版本差異



- Java自從發表後，不同的版本都能增加更多實用的功能。最值得注意是1999年發表的Java 1.2版，這也是較多人所熟悉的「Java 2」，此時，Java區分為四個不同的版本：
 - J2EE (企業版，Enterprise Edition)：主要提供企業伺服器端應用程式的開發需求。
 - J2SE (標準版，Standard Edition)：主要提供桌上型電腦應用程式的開發需求。
 - J2ME (精簡版，Micro Edition)：主要是提供資訊家電應用程式的開發需求。
 - Java Card：主要是提供smart card市場的需求。
- 2004年所發表的J2SE 1.5版 (又稱為J2SE 5.0、Java SE 5.0，代號Tiger)也是值得注意的版本，除了內含功能的增多之外，語法也同時更新。
- 而2006年發行的Java標準版本6.0(Java SE6，也稱為Mustang)則是目前初學Java者最常使用的版本。

Java Virtual Machine (JVM)



- 執行Java程式的是虛擬機器JVM
 - JVM包含直譯器，也就是說程式執行的地方是在JVM
 - Bytecode檔案無法直接在作業系統下執行，而是在JVM下執行
 - JVM即是讓Java達到跨平台目的元件
 - 每個作業系統有自己專屬的JVM程式
 - Bytecode檔可以在所有不同作業系統的JVM上執行而達到一次編譯，到處執行的目的

Java Runtime Environment (JRE)

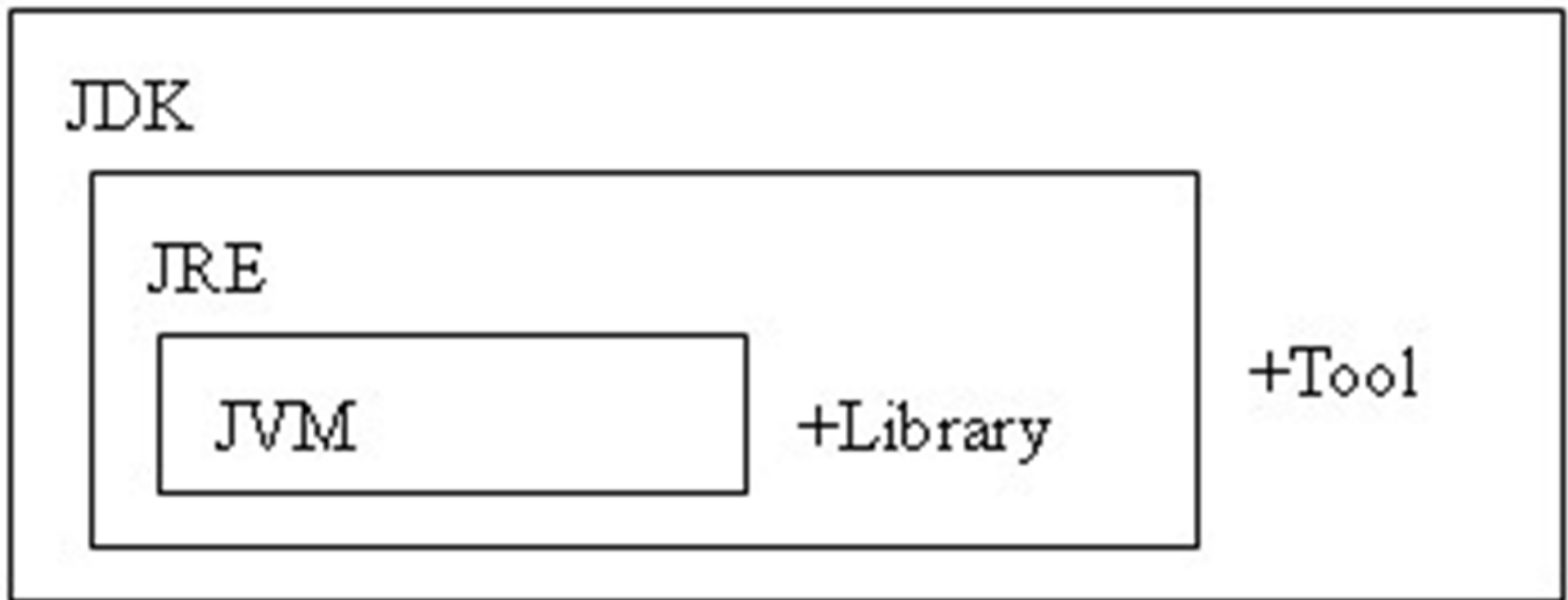


- JRE是讓電腦系統執行Java應用程式的執行環境
- JRE包含JVM和標準類別程式庫(Class Library)
- 專門為執行用戶設計，並不為開發者設計的環境
 - 不包含編譯器

Java Development Kit (JDK)

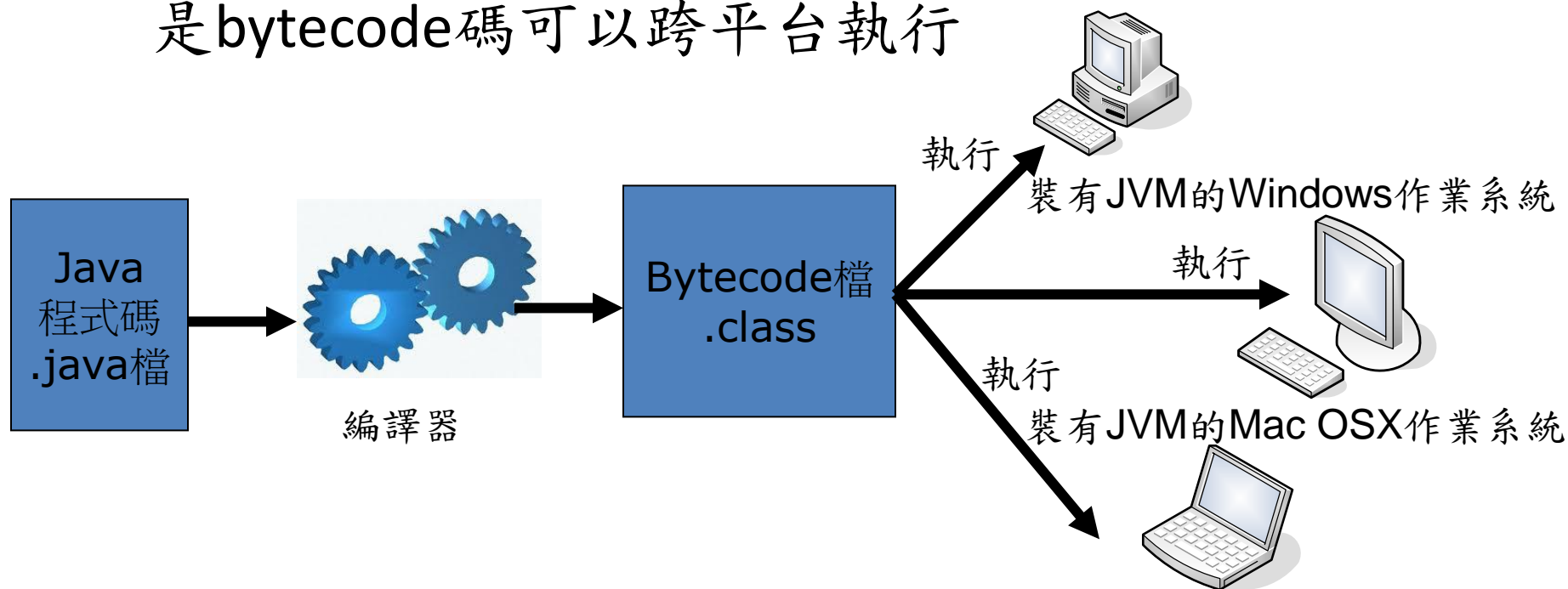
- 又稱SDK (Software Development Kit)
 - 1.4版後使用SDK替代JDK
- 針對Java程式開發者所設計的軟體開發工具包
- 基本元件
 - javac: 為Java編輯器，將Java程式轉譯成bytecode檔
 - jar: 製造Java類別壓縮檔的工具
 - javadoc: java文件產生器，程式設計師在程式中加上合乎規範的javadoc註解。當程式撰寫完成時即可透過javadoc來建立Java Platform API文件。此文件為HTML格式。
 - jdb: debugger工具

JVM, JRE, JDK三者關係



Java跨平台的特異功能

- Java透過又編譯又直譯的執行方式來達到跨平台的目的。
 - 跨平台不是指執行檔可以在任何平台上執行，而是bytecode碼可以跨平台執行



如何下載Java執行環境所需檔案



- 要開發Java程式前，我們要先下載Java SE8，並安裝相關的檔案。
 - 我們可以由Java的官方網站免費取得Java SE8。請連上「<http://java.sun.com/>」網站，並在該頁面顯示後，找到相關的Java SE8下載連結：