

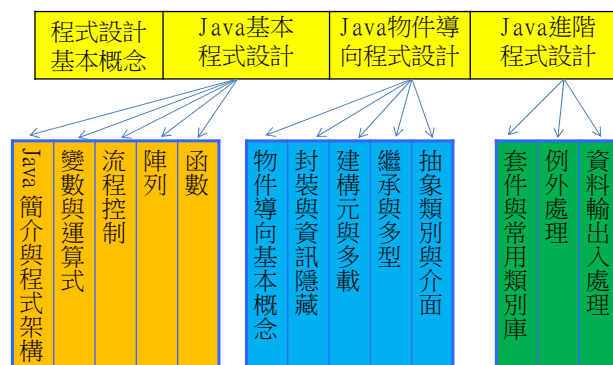
第一章

認識Java

- 1.1 Java的發展歷史
- 1.2 Java的虛擬機器(JVM)
- 1.3 Java未來的發展
- 1.4 撰寫第一個Java程式

1

Java的學習路徑



2

數據來源：<https://www.tiobe.com/tiobe-index/>

最新程式語言排行榜出爐(20210916)

Sep 2021	Sep 2020	Change	Programming Language	Ratings	Change
1	1		C	11.83%	-4.12%
2	3		Python	11.67%	+1.20%
3	2		Java	11.12%	-2.37%
4	4		C++	7.13%	+0.01%
5	5		C#	5.78%	+1.20%
6	6		Visual Basic	4.62%	+0.50%
7	7		JavaScript	2.55%	+0.01%
8	14		Assembly language	2.42%	+1.12%
9	8		PHP	1.85%	-0.64%
10	10		SQL	1.80%	+0.04%
11	22		Classic Visual Basic	1.52%	+0.77%
12	17		Groovy	1.46%	+0.48%
13	15		Ruby	1.27%	+0.03%
14	11		Go	1.13%	-0.33%
15	12		Swift	1.07%	-0.31%
16	16		MATLAB	1.02%	-0.07%

3

Very Long Term History

To see the bigger picture, please find below the positions of the top 10 programming languages of many years back. Please note that these are *average* positions for a period of 12 months.

Programming Language	2021	2016	2011	2006	2001	1996	1991	1986
C	1	2	2	2	1	1	1	1
Java	2	1	1	1	3	15	-	-
Python	3	5	6	8	25	24	-	-
C++	4	3	3	3	2	2	2	6
C#	5	4	5	7	13	-	-	-
Visual Basic	6	13	-	-	-	-	-	-
JavaScript	7	7	10	9	9	20	-	-
PHP	8	6	4	4	10	-	-	-
SQL	9	-	-	-	37	-	-	-
Assembly language	10	11	-	-	-	-	-	-
Ada	31	28	17	16	18	7	3	2
Lisp	33							

4

Java的歷史

- Java於1995年誕生，由美國加州的昇陽電腦公司（Sun Microsystems, Inc.）所推出
 - Java語言的前身是Oak，即橡樹之意
 - Sun將Oak修改成為Java，並將這種技術轉移到Web上
 - Web於1995年開始盛行，Java也隨之一炮而紅
 - 2009年，Oracle以74億美元買Sun Microsystems。
- Internet的發展史
(<https://web.ntnu.edu.tw/~499700326/Internethistory.htm>)

5

Java的由來



- 昇陽的Green Team設計新的程式語言「Oak」，也就是Java之前的名稱。
- 但是，為什麼Oak又要改名字為Java呢?這是因為當時Green Team要把Oak拿去註冊為商標時，發現Oak已經被別家公司註冊過了，因此不能用這個名字。
- 在Green Team多次討論之後，便決定以小組成員常去的那家咖啡店的店名來命名這個程式語言（Java本來是一種咖啡豆的名稱）。
- 其實您可以發現，Java的標誌是一杯咖啡的圖樣呢!

6

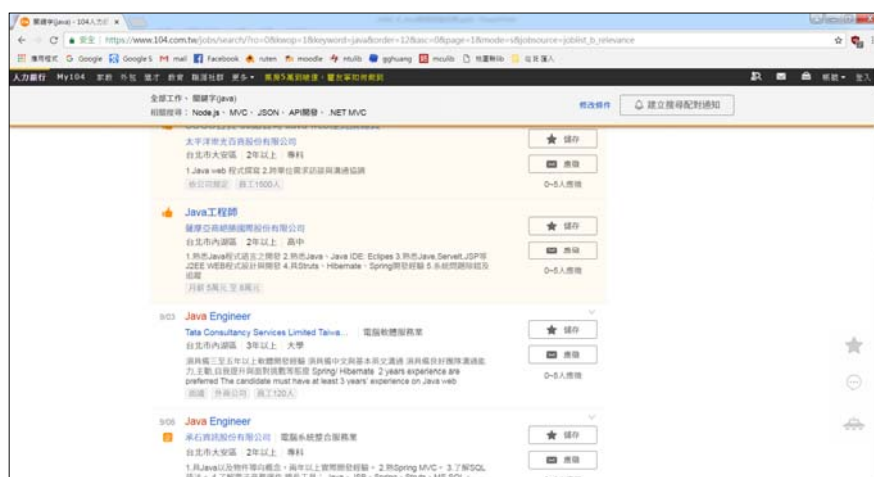
課堂作業：查詢Java相關工作機會

複製下述題目，完成之後，上傳至moodle規定上傳區
檔名：學號姓名.docx

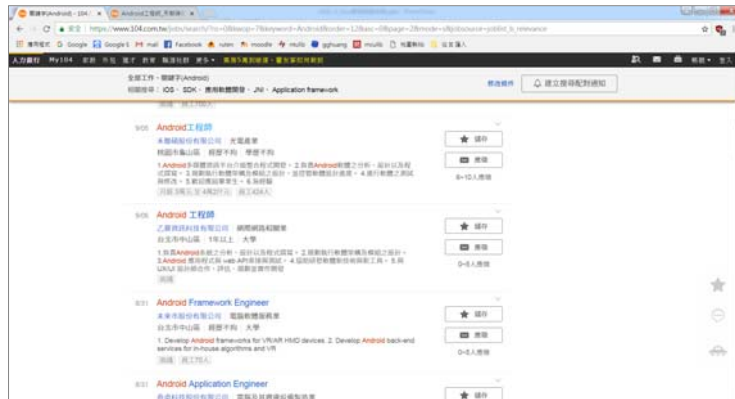
1. 請同學上104人力銀行(<http://www.104.com.tw/>)查Java的關鍵字，共有多少工作機會。
2. 想像工作機會
請根據上述的查詢結果中，找尋一個薪資+地點+工作條件符合你期待的工作。
3. 具備能力
根據你所描述的工作，你需要具備哪些工作能力，才能成功獲取這個工作呢？
4. 找完之後，你有沒有什麼了解或收穫？

7

104-Java關鍵字(5347)



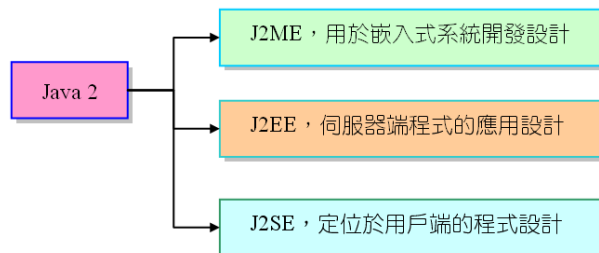
104-Android關鍵字(3262)



1.1 歷史的回顧

Java的應用領域

- 下圖為Java的三大應用領域：



在Java 為了解決不同的需求
分為下列三種不同的版本

- **JSE(Java Platform, Standard Edition)**
 - 是Java的標準版，包含Java SDK、Java API、JRE(Java Runtime Environment, Java執行環境)、除錯工具...等。 **本書即是安裝JSE**。
 - **JME(Java Platform, Micro Edition)**
 - 用於消費性電子商品及嵌入式系統。例如：可設計手機遊戲、PDA應用程式、汽車衛星導航系統...等
- 
- **JEE(Java Platform, Enterprise Edition)**
 - 是Java的企業版，用於企業應用程式，擁有J2SE的功能、EJB(Enterprise Java Beans)、JSP、支援XML及CORBA技術...等。



11

Java的實際應用範圍



Java的特點 (1/3)

- 嬌小且完美的語言
 - Java的語法簡潔、勻稱，容易撰寫且易於維護
- 具物件導向的功能
 - Java具有物件導向的特點
 - 它是全新的語言，因而可以將物件導向的特性發揮到極至
- 完全支援網際網路
 - Java所撰寫的程式很容易地在Browser裡呈現
 - 可藉此讓所有能夠上網的電腦都能執行Java程式

13

Java的特點 (2/3)

- 通用的語言
 - 凡是C/C++所能做到的事，Java也都能做到
- 跨平台的語言
 - 可在不修改程式碼的情況下，便能在不同的作業系統執行
- 具有豐富的函數庫
 - Java語言背後有龐大的函數庫支撐
 - 這些函數庫包含了繪圖函數庫、圖形使用者介面函數庫（graphical user interface）、網路設計函數庫...等等

14

Java的特點 (3/3)

- 特殊的處理機制
 - 「多執行緒」機制可在同一時間執行不同的程序
 - 「垃圾收集」機制可將無用的變數所佔用之記憶體釋放
 - 「例外處理」機制可依情況拋出例外，使得程式不會因此中斷執行

15

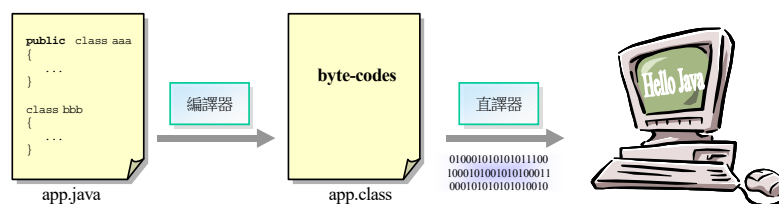
編譯與直譯

- 程式語言可分編譯式與直譯式兩種
 - 常見的編譯式語言如 C、FORTRAN與COBOL等均是
 - 直譯式的語言如 BASIC、JavaScript、VBScript、Python、Ruby 及其它Script語言
- Java程式的執行則是先編譯，後直譯

16

Java的位元碼

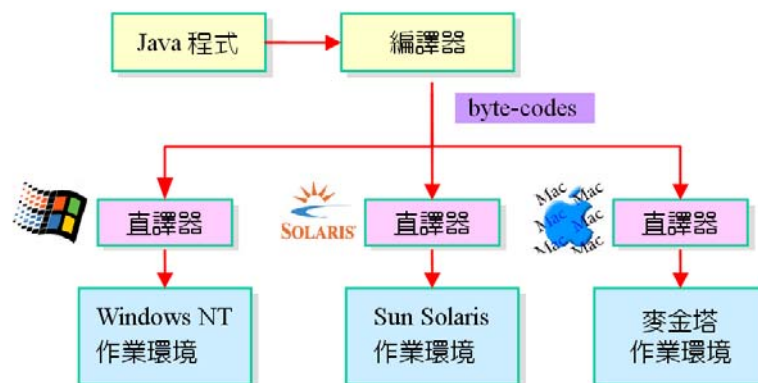
- Java的編譯與執行的程序：
 - 將程式碼編譯成與平台無關（platform-independent）的機器碼，稱之為「位元碼」（byte-codes）
 - 經編譯後，可在裝有JVM(Java virtual machine)上的平台直接執行
- 下圖說明Java相關的執行流程：



17

byte-codes 的執行

- byte-codes最大的好處是--可跨越平台來執行：



18

Java程式的分類

- Java程式可分為下列兩種：
 - (1) Java應用程式 (Java application)
 - (2) 應用在www上的小程序 (Java applet)

19

簡單的程式範例 (1/3)

- 下面是app1_1的程式碼
 - 經編譯和執行後，會在Dos視窗上顯示Hello Java!!字串：

```
01 // app1_1, 簡單的 java application
02 public class app1_1
03 {
04     public static void main(String args[])
05     {
06         System.out.println("Hello Java!!");
07     }
08 }
```

//兩撇是註解
//大小寫有差異

20

```

package ch01;

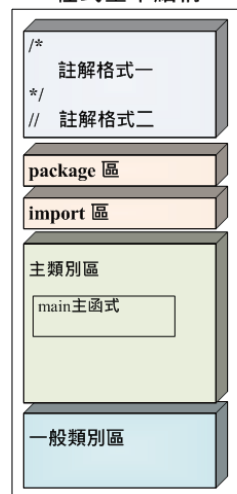
public class app1_1
{
    public static void main(String args[])
    {
        System.out.println("Hello Java!!");
    }
}

//兩撇是註解
//大小寫有差異
/* app1_1 OUTPUT-----
Hello Java!!
-----*/

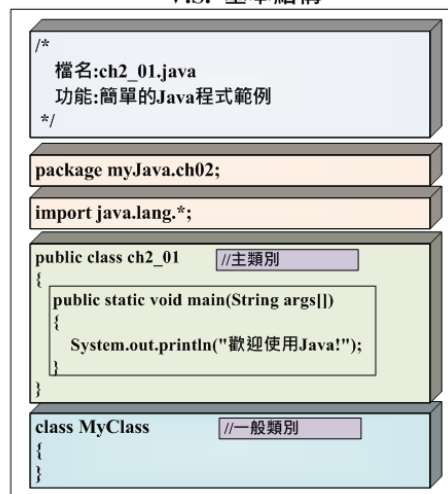
```

Java程式的基本結構

程式基本結構



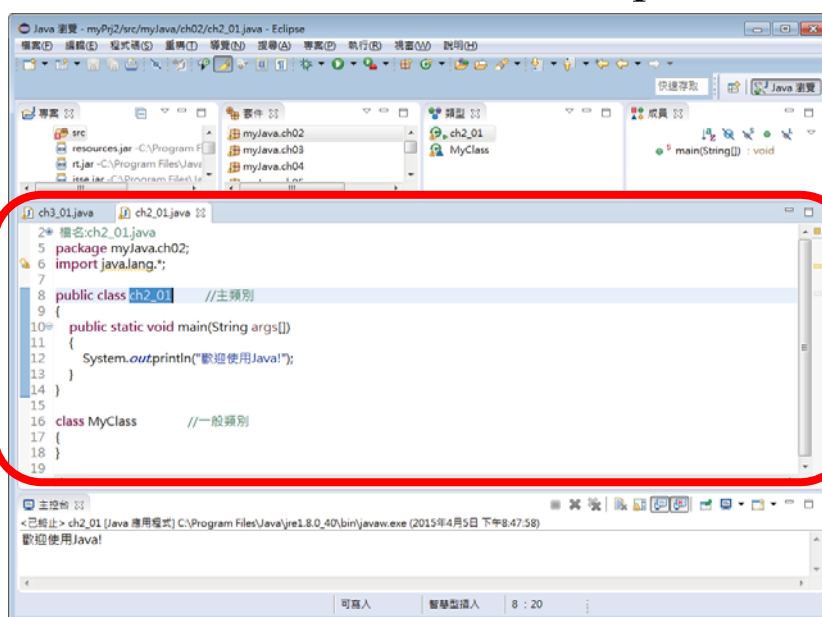
V.S. 基本結構



各程式區塊說明

- **package區**
 - 是用來宣告該檔案中的類別屬於哪一個Package（類別庫），必須出現在程式的最上方，且只能有一個。
- **Import區**
 - 在開發Java程式時，或多或少我們會使用到一些現成的類別(class)，以縮短程式開發的時程。這些現成的類別可能是由JDK或其他廠商所提供的，也可能是自己先前已開發完成的。無論如何，想要引用這些類別，就必須先將其在import區中宣告
- **主類別區**
 - Java是一個物件導向語言，以類別做為單位，所以每一個Java程式必須有一個主類別才能夠被執行。

Java開發環境介紹-Eclipse



找錯誤1

```
01 public class ex1_1 {  
02     public static void main(String[] argv) {  
03         System.out.println("//我要列印的訊息"我的Java程式");  
04     }  
05 }
```

找錯誤2

```
01 public class ex1_2 {  
02     public static void Main(String[] argv) {  
03         System.out.println("我的Java程式");  
04     }  
05 }
```

找錯誤3

```
01 public class ex1_3 {  
02     public static void main(String[] argv) {  
03         System.out.println("我的Java程式")  
04         system.out.println("怎麼會有錯？");  
05     }  
06 }
```

課堂練習(一)

(1) 練習第一個小程式

– 檔名 **ex1_4.java**

- 在output中顯示你的「姓名」、「綽號」與「興趣」字串

– 練習完請上傳檔案**ex1_4.java**至moodle

姓名：xxx
綽號：xxx
興趣：xxx

課堂練習(二)

ex1_5.java

請撰寫一個 Java 程式, 執行後可以在螢幕上顯示以下這首唐詩：

春眠不覺曉, 處處聞啼鳥
夜來風雨聲, 花落知多少

課堂練習(三)

ex1_6.java

請撰寫一個 Java 程式, 執行後可以在螢幕上顯示以下圖形：

```
*  
* *  
* * *  
* * * *  
* * * * *
```