

《程式語言》

試題評析

今年程式題皆以Java命題，並包含許多實務上的應用考題，雖然都屬於基本題，但與以往慣例出現文法或特殊語言明顯不同，對於有程式設計實務經驗的考生而言較為吃香。

第一題：以Java中的介面命題，考生除了需要能應用介面的特性外，還要能理解題目中所畫的圖，作答可能因為小瑕疵而不易拿滿分。

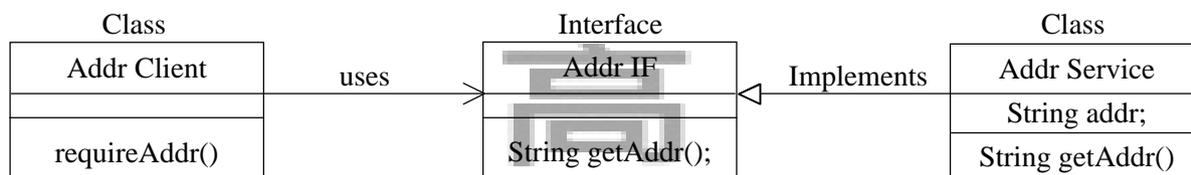
第二題：注重Web Service的概念與應用，雖然題目幾乎不會出現在以往的程式語言考題，可是卻是資訊管理常見考古題，因此全科準備的考生應可以拿到不錯的分數。

第三題：用遠高於以往的配分考出AJAX，雖屬於考古題，仍然需要對AJAX的運作有較深刻的體驗才能拿到高分。

第四題：考「多型」的概念，屬於考古題，考生若能充分解釋應可拿到高分。

整體而言，今年的考題沒有出現太艱澀的題目，且幾乎都是出現過的考古題，程度好的考生應可拿到80分以上，而程度中等的考生要拿到60分也絕非難事。

一、下圖為開發程式時常見之Interface設計模型（Design Pattern），其目的是要透過一個服務介面來將服務內容區隔出來，以達到使用者（Client）透過介面呼叫服務的方式不受服務內容修正的影響，試以Java程式指令將此設計模型實作出來。（20分）



答：

以Java程式實作如下：

```

1 public class AddrClient {
2     void requireAddr(AddrIF ai) {           //"uses" 介面
3         System.out.println("addr is: "+ai.getAddr());
4     }
5 }
6 public static void main(String[] args) { //測試AddrClient物件
7     AddrClient ac = new AddrClient();
8     ac.requireAddr(new AddrService()); //可依照需求替換服務內容
9 }
10 }
11
12 interface AddrIF {                       //宣告介面
13     String getAddr();                     //宣告介面中的方法(不實作)
14 }
15
16 class AddrService implements AddrIF{     //"implements" 介面
17     String addr;
18     public String getAddr() {            //實作getAddr方法
19         return addr;                    //一定要宣告為public，因為
20     }                                     //介面中的方法預設都是public
21     AddrService() {                      //建構子，用來初始化addr
22         addr = "台北市開封街一段2號8樓 (高上高普特考)"; // (考試時可不寫)
23     }
24 }
  
```

【版權所有，重製必究！】

程式輸出：

addr is：台北市開封街一段2號8樓(高上高普特考)

【參考書目】

- 1.金乃傑，《程式語言》講義，第四回，P.28-31。
- 2.金乃傑，《程式語言》總複習，P.43-46

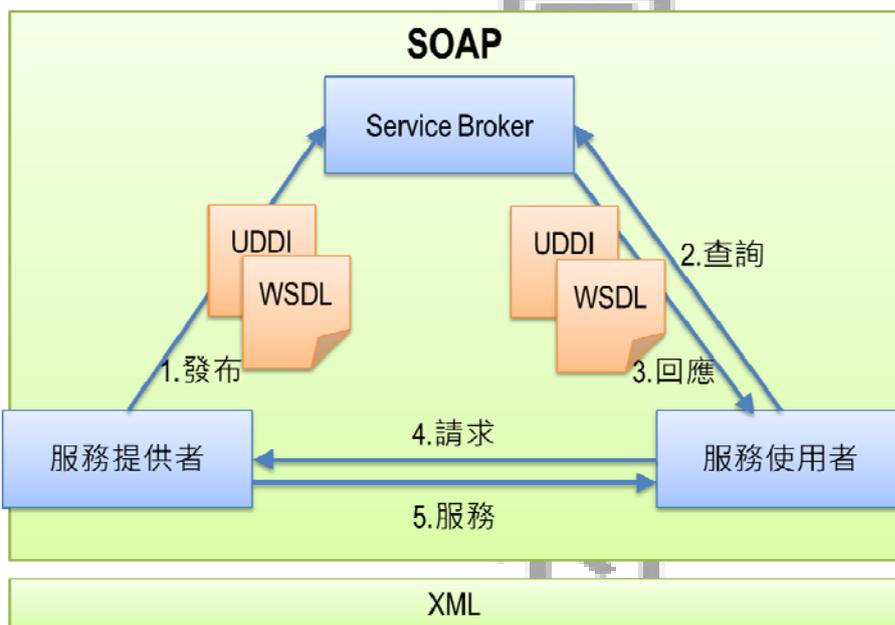
二、(一)試說明Web Service技術之組成部分及特性。(20分)(二)對於政府而言，您認為政府可以如何運用此技術以提升政府的效能或施政滿意度。(10分)

答：

(一)Web Service是在Web環境上，讓各種不同平台與程式語言的軟體元件，能彈性的、動態的快速整合、彼此互通、呼叫的一種開放性結構與標準。由以下技術組成：

- 1.XML (Xtensible Markup Language)：通用、標準化的資料結構描述語言。可使不同程式傳送資料時不需經過耗時的轉換，就能快速交換與解讀。
- 2.SOAP (Simple Object Access Protocol)：Web Service交換結構化資訊的協定。其去除網頁伺服器對XML文件格式化頁面的工作，讓不同應用程式間透過HTTP等通訊協定，以XML互相交換彼此的資料，使與程式語言、平台和硬體無關。
- 3.UDDI (Universal Description Discovery and Integration)：透過SOAP查詢的XML文件，是一種描述規範。用戶端可向UDDI伺服器尋找所需的服務，企業新增的網路服務亦可在UDDI上註冊以供其他用戶端查詢。
- 4.WSDL (Web Service Description Language)：透過SOAP查詢的XML文件，其描述了Web Services服務的存取方法和使用協議的細節，讓使用者讀取、了解並使用這些服務，通常用來輔助建構服務存取的程式碼。

這些技術的互動關係如下：



Web Service的特性如下：

- (1)鬆散耦合 (loosely coupled)：透過模組化，每個元件都可以動態的抽換，相依性低。
- (2)服務導向 (service oriented)：以提供服務為目的。
- (3)自我描述 (self-described)：使用機器看得懂的通用規格UDDI、WSDL，使得程式可以自動化對Web Services存取。
- (4)程式對程式溝通 (Application-to-Application)：不需要使用者從中介入，就能達到提供服務的目的。

【版權所有，重製必究！】

(二)由於Web Service具有跨平台、擴展、安裝成本低及軟體再利用的特性，因此政府部門可以利用Web Service來整合不同單位或組織的平台服務，並利用Web Service的跨平台特性，讓使用不同平台設備的人民都可以利用Web Service與政府系統溝通，獲得服務。例如：台北市政府可以將聯合醫院的線上掛號做成Web Service模組，以便與民眾自行開發的網站、手機APP結合，使得看病掛號的管道更多元，且能混搭成貼心的便民服務。

【參考書目】

金乃傑，《資訊管理與資通安全》講義，第一回，P.21-23。

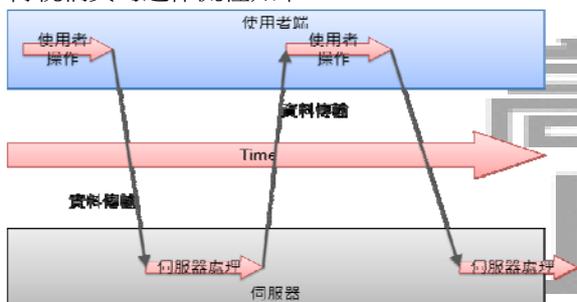
三、在Web 2.0相關技術中，AJAX是一個常被用來建構互動式網頁應用的網頁開發程式技術，試描述其運作方式並指出其優缺點。(25分)

答：

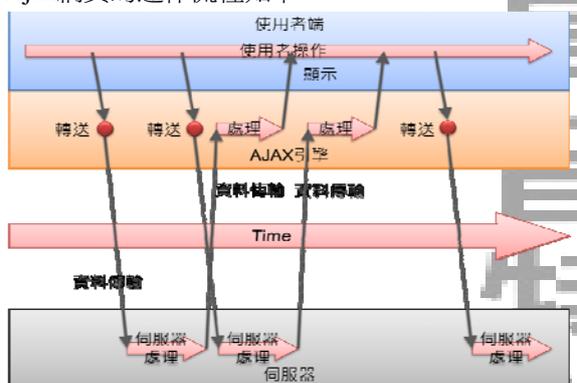
(一)AJAX全名為(Asynchronous JavaScript and XML)，顧名思義是透過javascript非同步的存取XML。AJAX並非單獨的技術，而是由一群現有的技術所組成。通常來說：

- 1.在javascript方面，以XMLHttpRequest物件背景存取網站伺服器的資料，並解析成網頁中所需顯示的格式。在實務上常使用javascript的jQuery函式庫以簡化相關的查詢語法。
- 2.網站伺服器方面，以往因資料交換常以XML封裝(AJAX的X因此得名)，但現在多以json格式將資料包裝。
- 3.而網頁端需要搭配CSS與div標籤，動態控制要在網頁中呈現的資料。

傳統網頁的運作流程如下：



Ajax網頁的運作流程如下：



說明：

傳統網頁必須要等伺服器處理完全部畫面後，使用者才能看到網頁的變化。因此不管什麼操作，都要等待全部的傳輸時間，而伺服器也因為要處理完整個網頁，使得反應較慢。AJAX以非同步的方式，使用javascript動態替換掉網頁的部分內容，使用者可以在伺服器還沒回應上一個要求時，就執行下一個要求。伺服器在處理時，因為只需要處理較小量的資料，在處理時間、傳輸時間均會比傳統網頁來的快速。

【版權所有，重製必究！】

(二)優缺點：

AJAX有以下優點：

- 1.提升使用者互動性 (interactivity)：使用者可以在伺服器還沒處理完上一個要求時，就執行下一個要求，減少使用者的等待時間，使網頁操作更為流暢。
- 2.速度快 (speed)：使用AJAX可以減少每次資料傳輸的量，因此可以減少傳輸時間；在資料處理方面，畫面的配置由AJAX引擎輔助產生，可以使用使用者電腦的運算資源，而伺服器每次也只要回應部分的資料，節省運算整張網頁的時間。

AJAX有以下缺點：

- 1.破壞瀏覽器的瀏覽行為：AJAX使網頁的介面變得模糊，因此可能在使用者按下「重新整理」或「上一頁」時變成不可預料的狀態。例如：資料都消失了，或者無法回到原先預期的頁面。
- 2.使用者難於將某個特定狀態的網頁分享或收藏：也因為網頁內容動態被替換，與傳統網頁不同，網址已不是決定網頁畫面的唯一指標，同一個網址可能因為使用者不同的操作步驟看到不同的畫面。因此在使用者分享頁面資訊、或加入最愛收藏時，可能顯示或儲存與預期不同的資料。
- 3.搜尋引擎不易索引：搜尋引擎透過超連結儲存經過的網頁路徑，但通常不具備執行javascript的功能，因此AJAX的動態畫面搜尋引擎可能無法作索引，而減少網頁的曝光率。

【參考書目】

金乃傑，《程式語言》講義，第五回，P.27-28。

四、何謂物件導向程式語言 (object-oriented programming language) 中之多型 (Polymorphism)？試以下列Java類別定義，透過物件建立與物件內方法呼叫說明之。(25分)

```
public class A1
{
    private int a;
    public int Compute(int x){return x * x + 5000;}
}

public class A2 extends A1
{
    private int a;
    public int Compute(int x){return x * x + 6000;}
}

public class A3 extends A2
{
    private int a;
    public int Compute(int x){return x * x + 7000;}
}
```

答：

- (一)多型：相同的程式碼在執行時其可具有多種執行模式。多型可以讓相同的程式碼，即使經過編譯，亦能在執行時期具有多種執行模式，因此可提供很高的執行彈性，但因其使用動態繫結，增加了程式執行的不可預測性。此外，因為必須在執行時才決定呼叫操作，會降低程式執行效能。
- (二)考慮以下Java程式，我們假設它可以存取試題上所列之類別：

【版權所有，重製必究！】

```

1 class MAIN{
2     public static void main(String[] args)
3     {
4         A1 aArray[] = new A1[3];           //建立一個陣列
5         aArray[0] = new A1();             //裡面存放不同的A
6         aArray[1] = new A2();
7         aArray[2] = new A3();
8
9         for(int i = 0; i < 3; i++){       //以迴圈輸出陣列
10            System.out.println("aArray["+i+"] x is "+aArray[i].Compute(1));
11        }                                   //語法皆相同，但卻依照實際的物件使用
12    }                                       //物件中不同內容的Compute方法
13 }

```

輸出：

```

aArray[0] x is 5001
aArray[1] x is 6001
aArray[2] x is 7001

```

說明：

根據題意，A1是A2的parent；A2是A3的parent。我們在建立物件時，由於A1、A2、A3間存在繼承的關係，A1是最上層物件，因此我們不管建立哪種物件，都可以用A1型態宣告。例如：上述程式中的陣列先以A1型態宣告，但在實際建立成員元素的時候再用不同的類別產生。

在使用物件內的Compute方法時，我們可以發現for迴圈內的語法不會改變，但Java卻可以根據陣列中實際存放的物件而去呼叫對應的方法處理，使相同的程式碼在執行時達到多種執行模式，即所謂的多型。

【參考書目】

1. 金乃傑，《程式語言》講義，第四回，P.30-31。
2. 金乃傑，《程式語言》總複習，P.45-46

高上高普特考

【版權所有，重製必究！】