

## 104 年公務人員特種考試身心障礙人員考試

考試別：身心障礙人員考試

等別：三等考試

類科：資訊處理

科目：程式語言

一、請解釋 Java 語言中，類別與介面 (interface) 有那些不同，並請舉例加以說明。(20 分)

擬答：

	類別(class)	介面(interface)
定義	是由相同結構與行為的許多物件所組成的的集合	是抽象類別的一種，其中包含屬性與未實作的方法
宣告方法	<pre>class shape{     int x,y;     int cntArea(){         return x*y;     } }</pre>	<pre>interface shape{     int x=5;int y=10;     int cntArea(); }</pre>
繼承方式	<pre>class abc extends shape{     int a,b;     int cntW(){         return x+a+b+y;     } }</pre>	<pre>class a implements shape{     int a,b;     public int cntArea(){         return x*y;     }     int cntW(){         return x+a+b+y;     } }</pre>
說明	<ol style="list-style-type: none"> <li>1. 關鍵字為 class 且使用 extends 為繼承的關鍵字</li> <li>2. 只允許單一繼承</li> <li>3. 存取修飾子可由使用者自行決定</li> <li>4. 類別 abc 如果與 shape 在同一資料夾中，則可呼叫到 cntArea()和變數 x 和 y</li> <li>5. 在進行 override 時，存取修飾子可以是 public、protected 和 default(不寫)，彈性比 interface 還要大</li> </ol>	<ol style="list-style-type: none"> <li>1. 關鍵字為 interface 且使用 implements 為實作的關鍵字</li> <li>2. 可以用來模擬多重繼承，換言之，實作是可以繼承多個介面</li> <li>3. 在介面中宣告的變數會被加上 public static final，所以必須給值，方法則是會被加上 public abstract，所以不用實作，但實作此介面的一定還要實作這些方法</li> <li>4. 承續第 3，compiler 會自動幫忙加上，所以在實作方法時，存取修飾子必須是 public，因為要遵守 override 的規則</li> </ol>

二、網頁應用程式常見的安全漏洞有跨站請求偽造 (cross-site request forgery)，請說明造成此安全漏洞的原因，以及如何避免此問題之措施。(20 分)

擬答：

(一)主要的攻擊行為是利用使用者合法取得網站使用認證後，透過某些方式偽造網站合法使用者的身分進行非法的存取動作，合法使用者可能在不知道的情況下被引導到駭客設置的網頁。一般而言，使用者進行認證後，伺服器會回傳一個存有驗證成功的資訊，稱為 cookie，客戶端是無法解開看的，但會存在瀏覽器上的，當使用者被引誘至駭客的網頁中後，會以非法的方式要求使用者傳送 cookie 給他，這時就能取得該使用者的認證資訊並以使用者的身分非法的存取該認證之網站

(二)避免方式：

1. 避免使用 GET 方法傳送參數及操作新增、刪除和更新
2. 避免使用 HTTP Referer Header 來驗證來源者
3. 使用 Token 來驗證來源者 (SessionID 加鹽—亂數值)
4. 不使用網於時應儘快登出
5. 不使用瀏覽器記憶功能
6. 一般性瀏覽與存取重要的服務時可用不同之瀏覽器。若無法則至少用不同的視窗

三、為何尾遞迴 (tail recursion) 通常比一般性的遞迴執行上更有效率？請說明之。請以熟悉的程式語言寫出一段使用尾遞迴技巧的程式碼例子。(20 分)

擬答：

尾遞迴是在最後一行才呼叫自己，而一般性的遞迴可能在中間某一行呼叫自己，做完之後還要回到剛呼叫的那邊繼續往下執行，而尾遞迴因為下面已經沒有程式碼了，所以效率比較好，以下是一個計算階層的程式碼：(以 java 語言撰寫)

```
int f(int n){
    if(n==1) return 1;
    else return n*f(n-1);
}
```

四、考慮下列文法，大寫字母代表非終結符 (non-terminal)，其餘皆為終結符 (terminal)，E 為開始變數。

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow ( E ) \mid \text{identifier} \mid \text{number}$

(一)使用最右推導 (rightmost derivation)，推導出  $1 * (2 + 3)$  一句。(6 分)

(二)使用最左推導 (leftmost derivation)，推導出  $1 * (2 + 3)$  一句。(6 分)

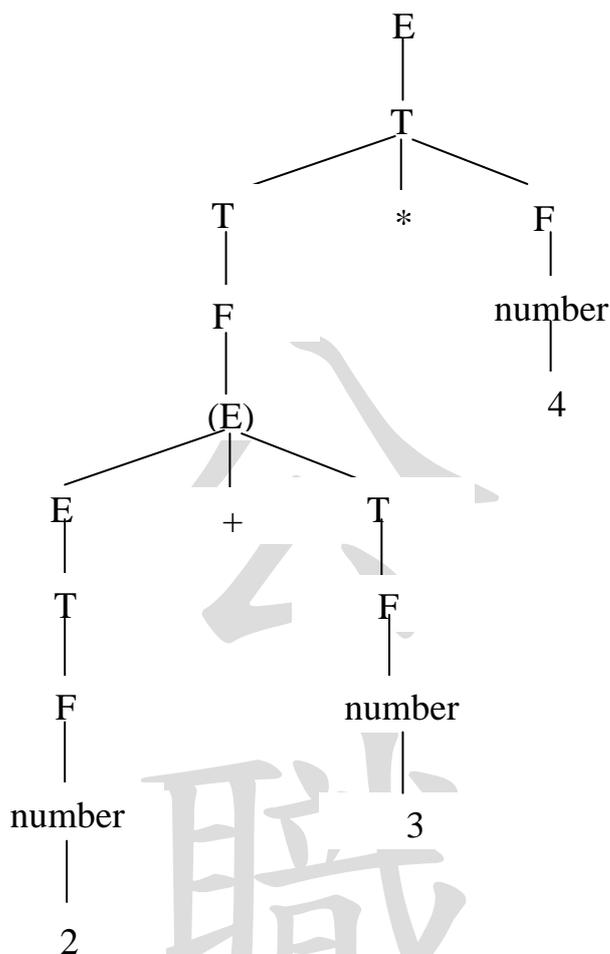
(三)繪製  $(2 + 3) * 4$  一句的剖析樹 (parse tree)。(8 分)

擬答：

最右推導： $E \rightarrow T \rightarrow T * F \rightarrow T * (E) \rightarrow T * (E + T) \rightarrow T * (T + T) \rightarrow T * (\text{number} + \text{number}) \rightarrow T * (2 + 3) \rightarrow F * (2 + 3) \rightarrow \text{number} * (2 + 3) \rightarrow 1 * (2 + 3)$

最左推導： $E \rightarrow T \rightarrow T * F \rightarrow F * F \rightarrow \text{number} * F \rightarrow 1 * F \rightarrow 1 * (E) \rightarrow 1 * (E + T) \rightarrow 1 * (T + T) \rightarrow 1 * (F + F) \rightarrow 1 * (\text{number} + \text{number}) \rightarrow 1 * (2 + 3)$

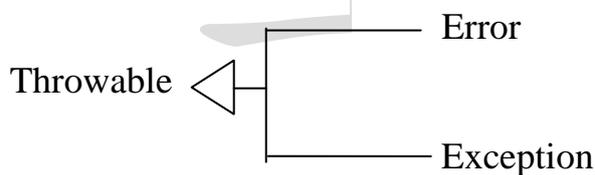
(2+3)\*4 之剖析樹如下:



五、例外處理程序 (exception handlers) 處理的例外有那幾類? 請分別舉例說明。Java 的例外處理程序如何處理例外? (20 分)

擬答:

例外處理程序是當例外發生時, 被要求來進行處理的程式單元, 其例外是指任何由硬體或軟體所偵測到的不正常事件或錯誤, 並需進行殊處理者, 希望透過例外處理程序讓程式能順利執行完畢。在 java 中, 是將例外處理程序寫在 try/catch 區塊中或是透過 throws 丟給上層呼叫的方法去處理, 若一直沒有處理, 最後會由 Throwable 體系中的類別攔截錯誤並顯示訊息, 其中 java 處理例外的類別之架構圖如下:



- (一) Error: 通常指的是系統本身發出的錯訊息
- (二) Exception: 是一個不正常的程式在執行期間所觸發的例外
- (三) 一般而言, 程式行中可能會發生 Error 或 RuntimeException(Exception 的子類別), 在 compiler 不會強制必須實作 try/catch 或 throws, 因為這類型的錯誤是發生在系統資源不足、環境安裝和設定上的問題或是程式邏輯錯誤所導致, 在開發過程中是可以預防的。而其他的 Exception, 如: IOException、FileNotFoundException 就必須先用 try/catch 或 throws 方式將容易發生 Exception 的程式碼包住, 並攔截錯訊息之後再做例外處理