104 年公務人員特種考試關務人員考試試題

考試別：關務人員考試
等別：四等考試
類科：資訊處理
科目：程式語言概要

一、靜能類型程式語言需使用某種類型等價(type equivalence)方法以決定類型是否可匹配(compatible)，最常見的是名稱等價和結構等價，請說明這兩種方法。考慮下列程式碼(X:T 表示變數 X 之類型為 T)，在這兩種類型等價方法下，請分別說明其中那些變數的類型為可匹配。(20 分)

```
Type T = array 〔1...10〕of float
X:T
Y:T
Z:array〔1..10〕of integer
```

【擬答】：
名稱等價：
名稱相同 -每種聲明引入了一個新的類型，從所有其他不同的。
例如：

```
TYPE T1 = ARRAY OF CHAR [0..10]
TYPE T2 = ARRAY OF CHAR [0..10]
```

類型 T1 和 T2 是不等價的。
Java 和 Ada 語言使用的名稱相同。
結構等價：
結構等價 -如果它們由相同成分的兩種類型的相同。
例如：

```
TYPE T1 = RECORD
            ANSWER：INTEGER;
            B：ARRAY OF CHAR [0..10]
          END:
TYPE T2 = RECORD
            ANSWER：INTEGER;
            B：ARRAY OF CHAR [0..10]
          END;
```

類型 T1 和 T2 是相同的。
ALGOL-68，的 Modula-3，C，ML 使用結構類型等價。
在程式中，如以名稱等價來看，則 X 與 Y 可以匹配；結構等價來看也只有 X 與 Y 可以匹配。

二、物件導向語言中，對宣告之實體(如屬性、方法)的可見性(visibility)有那三種常見的限制？請用你熟悉的語言舉一個簡單的例子解釋這三種限制造成的不同可見性。(20 分)

【擬答】：
類別的欄位與方法都對有不同的可見性（visibility）等級，包括公開（public）、私有（private）與保護（protected）。不同的可見性等級，限制不同背景環境（context）下所能用到的欄位與方法。
成員的屬性可分為三種：

㈠ public：能以點運算子直接存取。
㈡ private：只有類別的函數成員才可以存取。
㈢ protected：可供該類別及其衍生類別的函數成員存取。

繼承方式對不同的成員屬性：

繼承方式	基底類別	衍生類別
public	Public	public
	Private	無法被繼承
	Protected	protected
private	Public	private
	Private	無法被繼承
	Protected	private
protected	Public	protected
	Private	無法被繼承
	Protected	protected

繼承方式與成員屬性

```
#include<iostream . h>
class A
{
   public : int  pub ;
          void  fa ( )
     {
          int  a ;
          a = pri ;
          a = pro ;
          a = pub ;
     }
   private : int  pri ;
   protected : int  pro ;
} ;
class  B : public A
{
  public : void  f ( )
    {
       int  a ;
       a = pri ; // not  accessible
       a = pro ;
       a = pub ;
    }
} ;
class  C : private  A
{
   public : void  f ( )
     {
        int  a ;
        a = pri ; // not  accessible
        a = pro ;
        a = pub ;
     }
};
class  D : protected  A
{
```

```
            public : void  f ( )
              {
                 int  a ;
                 a = pri ; // not  accessible
                 a = pro ;
                 a = pub ;
              }
         } ;
         void  main ( )
         {
            int  a ;

            B    objB ;
            a = ojbB . pri ; // not  accessible
            a = ojbB . pro ; // not  accessible
            a = ojbB . pub ; // not  accessible

            C    objC ;
            a = ojbC . pri ; // not  accessible
            a = ojbC . pro ; // not  accessible
            a = ojbC . pub ; // not  accessible

            D    objD ;
            a = ojbD . pri ; // not  accessible
            a = ojbD . pro ; // not  accessible
            a = ojbD . pub ; // not  accessible
         }
```

要讓衍生類別的函數成員能夠存取基底類別的成員，又有資料保護的功能，需要將基底類別的成員屬性指定為 protected，並使用 public 繼承。

三、請用上下文無關文法(context-free grammar)寫出一套文法規則，以產生與正規表示式(regular expression)，a*(ba*ba*)，完全相同的語言(*符號代表可重複零到無數次)。再用你所寫的文法規則，用最右推導(rightmost derivation)，推導出 babaaabb 一句。(20 分)

【擬答】：

S→AABB

A→ba | baB

B→aa | bb

推導

S →AAB → babaBB → babaaabb

四、下列的 C 程式碼有何問題?如果忽略警告而逕行執行程式，在許多系統上，該程式將顯示重複的行為，列印出 0123456789，為什麼?也解釋為什麼在其他的系統上，該程式的執行結果可能會有所不同，甚至結果是無法確定的。(20 分)

```
Void foo(){
  int i;
  printf("%d"，i++);
 }
int main(){
```

```
        int j;
        for(j=1;j<=10;j++)foo();
    }
```

【擬答】：

在 Void foo()沒宣告這個程式段的型態；當主程式把變數 j 的值傳過去時，也沒有一個變數可以接收該值，所以程式會出現錯誤訊息。

但是有的 compiler 會讓 j 與 i 共用位址，這時就會重複印出 0123456789。

一般來說 C compiler 的預設是 call by value，也就是會各給 j 與 i 一個位址，在這種情況下，foo()宣告接收變數與該變數的型態就會出現錯誤。

五、假設你要寫一段程式碼，來管理數個並行執行緒(thread)之間共享的一雜湊表(hash table)，而雜湊表的操作必符合原子性(atomicity)。你可以使用一個互斥鎖(mutual exclusion lock)來保護整個表，你也可以用一個鎖分別保護每個雜湊表的桶(bucket)。請分別說明這兩種做法的優點和缺點。(20 分)

【擬答】：

㈠用一個互斥鎖

優點：當多個執行緒要來共用這張雜湊表時，只有一個執行緒可以使用到這張雜湊表，可以確保雜湊表的安全性。

缺點：沒有使用到雜湊表的執行緒就會讀不到雜湊表裡面的資料，會造成程式錯誤，或等待很久。這就失去執行緒平行處理的優點

㈡一個鎖分別保護每個雜湊表的桶(bucket)

優點：因為是保護一個鎖分別保護每個雜湊表的桶(bucket)，所以當多個執行緒要使用雜湊表時，只要桶值不一樣，所有執行緒都可以在同一時間使用雜湊表。

缺點：當兩個或兩個以上的執行緒要讀相同的桶值時，也是一樣只有一個執行緒可以使用，其他執行緒一樣要等待。並且當有執行緒去修改桶值時，下一個執行緒也會讀到錯誤的資料。

這兩個都不是好的演算法，較可行的演算法是將每一個桶值均用一個佇列 (Queue)來管制當兩個或兩個以上的執行緒要讀相同的桶值時，用先進先出的方法來做。也可以保留執行緒平行處理的優點。