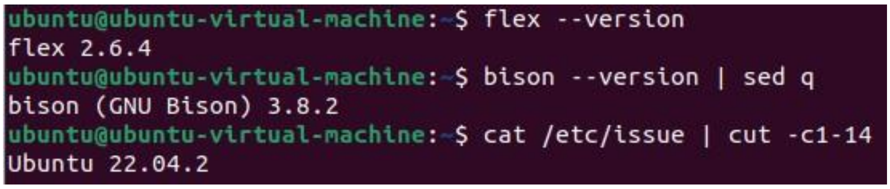
「編譯器製作」作業一 Readme Simple Java – Parsesr

姓名：鄭璟翰、學號：B093040003

一、Lex 版本：flex 2.6.4

Yacc 版本： bison (GNU Bison) 3.8.2

二、作業系統平台： Ubuntu 22.04.2



```
ubuntu@ubuntu-virtual-machine:~$ flex --version
flex 2.6.4
ubuntu@ubuntu-virtual-machine:~$ bison --version | sed q
bison (GNU Bison) 3.8.2
ubuntu@ubuntu-virtual-machine:~$ cat /etc/issue | cut -c1-14
Ubuntu 22.04.2
```

三、執行方式：(1)、make；(2)、./calc < 測試檔案.java

四、如何處理規格書上所遇到的問題

(一)、Scanner 修改

將作業一的 lex 檔案修該為遇到一個 terminal 會回傳該關鍵字，再藉由 Parser 來判斷該關鍵字是否正確；而遇到 Parser 不需要處理的關鍵字則不須回傳。此外，辨識到一個 Pattern 時就將其輸出，遇到換行「\n」時輸出下一行的行數，最後一行則使用 yytext 變數來判斷是否需要輸出行數。

(二)、Symbol Table

將 Symbol Table 移動到 yacc 檔案，並將其擴充至三個維，三個維度分別是 CLASS、FUNCTION 及 VARIABLE，同時定義 enum TYPE 來操控這三個維度。Insert(TYPE, char*)負責將 ID 存入 Symbol Table 之中，IsDeclare(Type, char*)負責檢 ID 是否在 Symbol Table 中。

(三)、處理 Grammar

1.使用 classList 開頭

因為 Java 檔案中所有內容都是以 class 為最基本的單位，使用 classList 作為文法的開頭。

2.重複組成的語法

程式當中有許多內容是可以重複出現，如 class、classbody 等等，可以使用 left recursive 的語法(複數->複數 單數)來達到此文法。

3.依照規格書上的提示展開文法

(1)field 表示宣告，可以分成一變數、Array 及 Object 三種。

(2)method 除了使用一般 ID 宣告的變數，也可以使用"main"宣告。

(3)if-else 的寫法為了避免 ambiguous，將所以宣告刪除。

(4)和 Variable ID 相關的語法可能要加上"[]"，因為其可能為陣列。

4.Variable 重未使用(加分)

宣告 VariableUsed 變數紀錄該變數是否使用過，在 Variable 進入 insert(TYPE, char*)時紀錄為 false，而當被 IsDeclare(TYPE, char*)詢問或是其他文法有使用到十節紀錄為 true，Scope 結束檢查是否使用過。

五、寫這份作業遇到的問題

(一)、class declaration 判斷

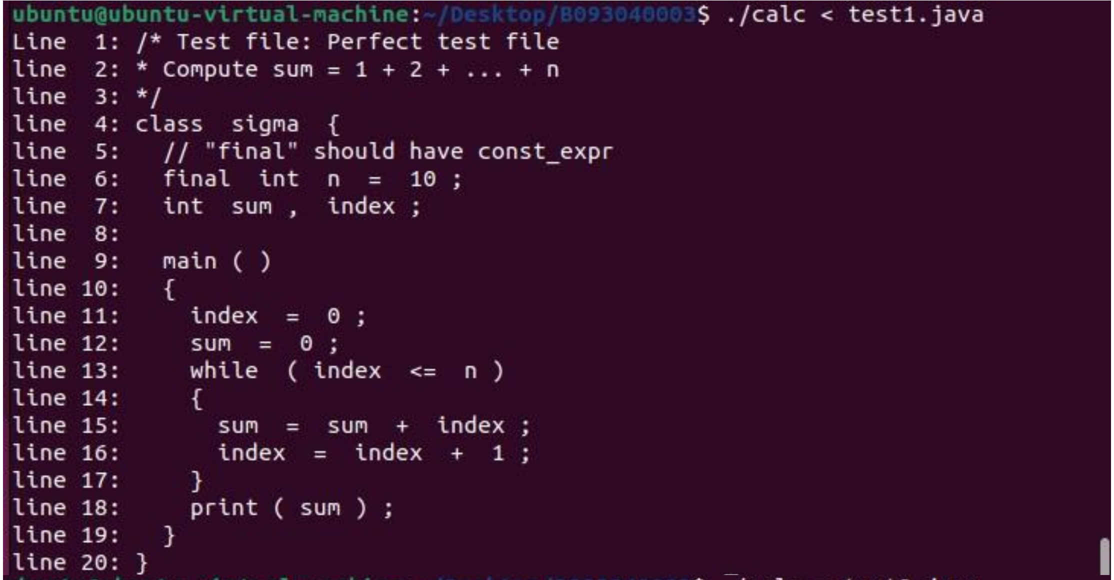
和 FUNCTION ID、VARIABLE ID 相比，Java CLASS 是可以在宣告前就使用，因此無法直接判斷該變數是否已經宣告過。解決方法為當遇到尚未宣告的 class name 時，先存入 Error ID 中，直到 Scope 結束時再判斷該變數是否宣告過。

(二)、規格書所提供的測試資料與執行結果和預期中不同。

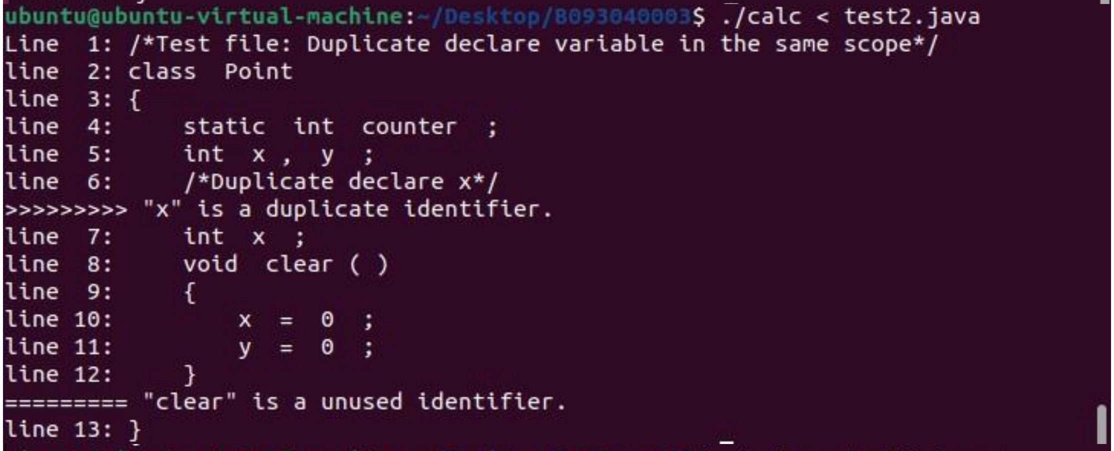
將於第六部分提及。

六、測試資料結果：

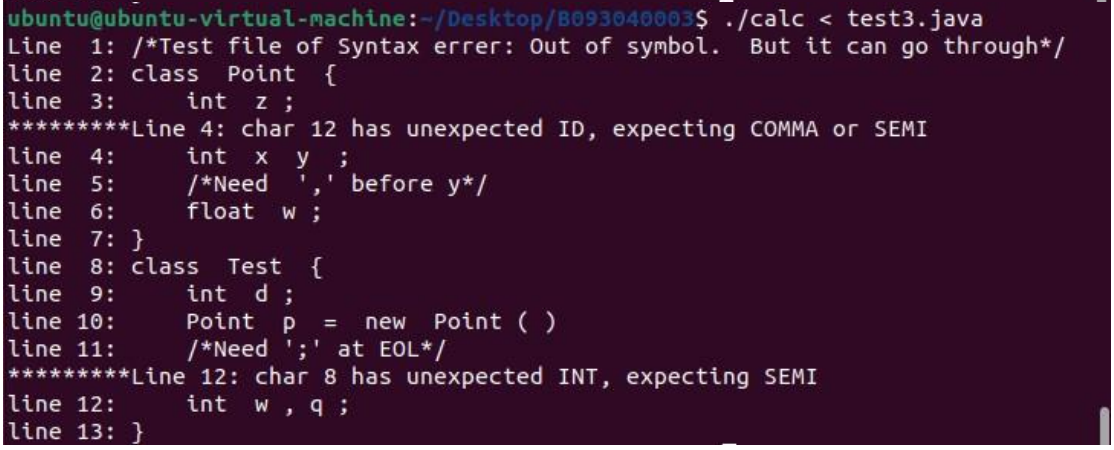
test1~3.java



```
ubuntu@ubuntu-virtual-machine:~/Desktop/B093040003$ ./calc < test1.java
Line  1: /* Test file: Perfect test file
line  2: * Compute sum = 1 + 2 + ... + n
line  3: */
line  4: class  sigma  {
line  5:   // "final" should have const_expr
line  6:   final  int  n  =  10 ;
line  7:   int  sum ,  index ;
line  8:
line  9:   main ( )
line 10:   {
line 11:     index  =  0 ;
line 12:     sum  =  0 ;
line 13:     while  ( index  <=  n )
line 14:     {
line 15:       sum  =  sum  +  index ;
line 16:       index  =  index  +  1 ;
line 17:     }
line 18:     print ( sum ) ;
line 19:   }
line 20: }
```

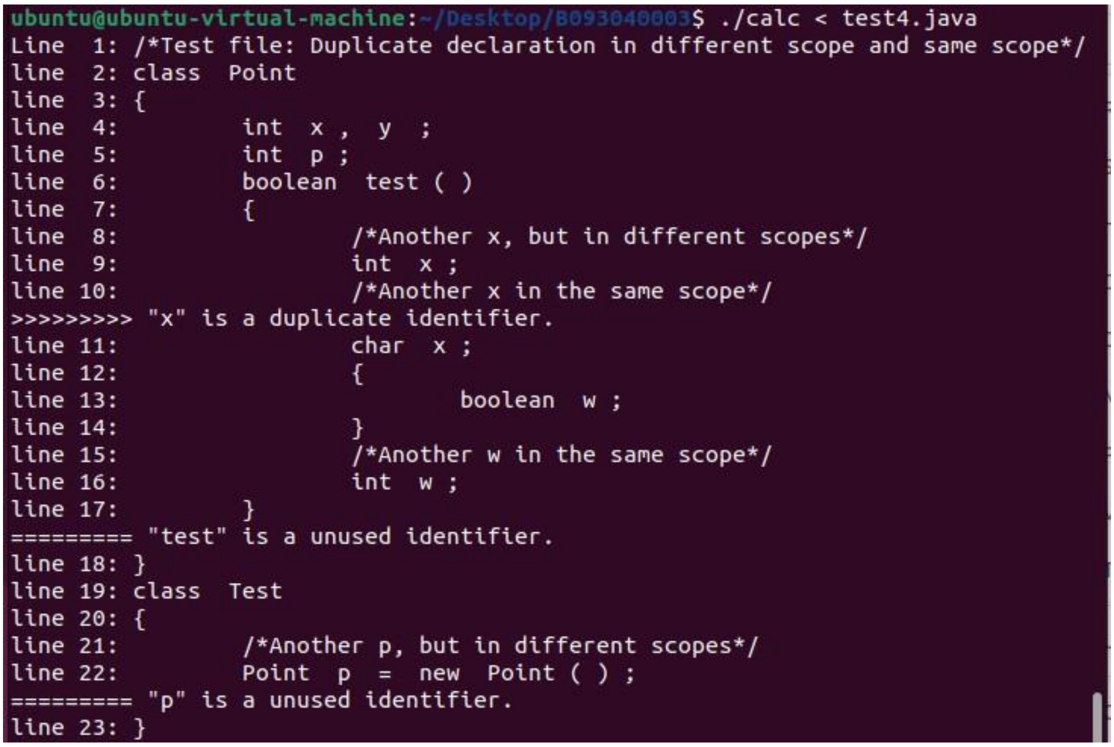


```
ubuntu@ubuntu-virtual-machine:~/Desktop/B093040003$ ./calc < test2.java
Line  1: /*Test file: Duplicate declare variable in the same scope*/
line  2: class  Point
line  3: {
line  4:     static  int  counter  ;
line  5:     int  x ,  y  ;
line  6:     /*Duplicate declare x*/
>>>>>>>>> "x" is a duplicate identifier.
line  7:     int  x  ;
line  8:     void  clear ( )
line  9:     {
line 10:         x  =  0  ;
line 11:         y  =  0  ;
line 12:     }
========= "clear" is a unused identifier.
line 13: }
```



```
ubuntu@ubuntu-virtual-machine:~/Desktop/B093040003$ ./calc < test3.java
Line  1: /*Test file of Syntax error: Out of symbol.  But it can go through*/
line  2: class  Point  {
line  3:     int  z ;
*********Line 4: char 12 has unexpected ID, expecting COMMA or SEMI
line  4:     int  x  y  ;
line  5:     /*Need  ',' before y*/
line  6:     float  w ;
line  7: }
line  8: class  Test  {
line  9:     int  d ;
line 10:     Point  p  =  new  Point ( )
line 11:     /*Need ';' at EOL*/
*********Line 12: char 8 has unexpected INT, expecting SEMI
line 12:     int  w , q ;
line 13: }
```

test4.java

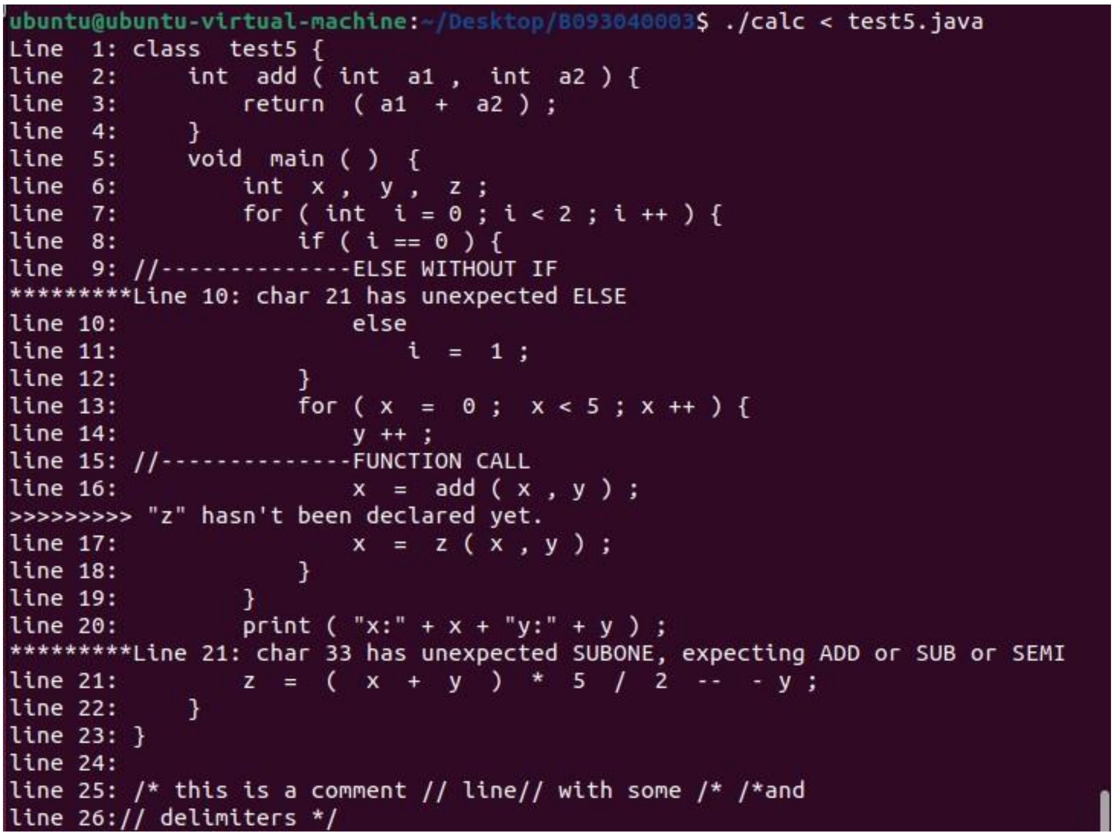


```
ubuntu@ubuntu-virtual-machine:~/Desktop/B093040003$ ./calc < test4.java
Line  1: /*Test file: Duplicate declaration in different scope and same scope*/
line  2: class  Point
line  3: {
line  4:        int  x ,  y  ;
line  5:        int  p ;
line  6:        boolean  test ( )
line  7:        {
line  8:               /*Another x, but in different scopes*/
line  9:               int  x ;
line 10:               /*Another x in the same scope*/
>>>>>>>>> "x" is a duplicate identifier.
line 11:               char  x ;
line 12:               {
line 13:                      boolean  w ;
line 14:               }
line 15:               /*Another w in the same scope*/
line 16:               int  w ;
line 17:        }
========= "test" is a unused identifier.
line 18: }
line 19: class  Test
line 20: {
line 21:        /*Another p, but in different scopes*/
line 22:        Point  p  =  new  Point ( ) ;
========= "p" is a unused identifier.
line 23: }
```

test5.java

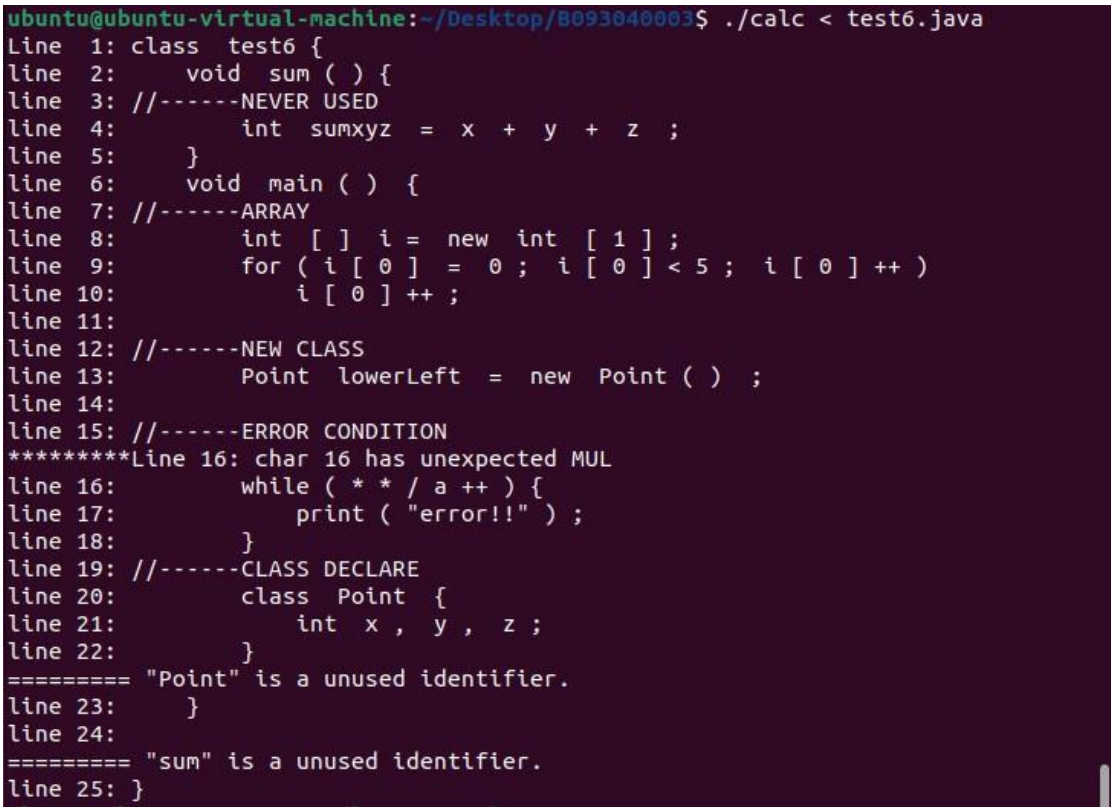
Line 17 的 ID “z”依照文法是屬於一個 FUNCTION ID，但是並沒有宣告過 z 這個 method 或 function，只有宣告過”z”這個 VARIABLE ID，因此我將其視為一個沒有宣告的 ID。

Line21 的”-- -y”是一個錯誤的 expression，因此判斷為錯誤。



```
ubuntu@ubuntu-virtual-machine:~/Desktop/B093040003$ ./calc < test5.java
Line  1: class  test5 {
line  2:    int  add ( int  a1 ,  int  a2 ) {
line  3:        return  ( a1  +  a2 ) ;
line  4:    }
line  5:    void  main ( )  {
line  6:        int  x ,  y ,  z ;
line  7:        for ( int  i = 0 ; i < 2 ; i ++ ) {
line  8:            if ( i == 0 ) {
line  9: //-------------ELSE WITHOUT IF
*********Line 10: char 21 has unexpected ELSE
line 10:               else
line 11:                   i  =  1 ;
line 12:            }
line 13:            for ( x  =  0 ;  x < 5 ; x ++ ) {
line 14:               y ++ ;
line 15: //-------------FUNCTION CALL
line 16:               x  =  add ( x , y ) ;
>>>>>>>>> "z" hasn't been declared yet.
line 17:               x  =  z ( x , y ) ;
line 18:            }
line 19:        }
line 20:        print ( "x:" + x + "y:" + y ) ;
*********Line 21: char 33 has unexpected SUBONE, expecting ADD or SUB or SEMI
line 21:        z  =  (  x  +  y  )  *  5  /  2  --  - y ;
line 22:    }
line 23: }
line 24:
line 25: /* this is a comment // line// with some /* /*and
line 26:// delimiters */
```

test6.java



```
ubuntu@ubuntu-virtual-machine:~/Desktop/B093040003$ ./calc < test6.java
Line  1: class  test6 {
line  2:     void  sum ( ) {
line  3: //------NEVER USED
line  4:         int  sumxyz  =  x  +  y  +  z  ;
line  5:     }
line  6:     void  main ( )  {
line  7: //------ARRAY
line  8:         int  [ ]  i =  new  int  [ 1 ] ;
line  9:         for ( i [ 0 ]  =  0 ;  i [ 0 ] < 5 ;  i [ 0 ] ++ )
line 10:             i [ 0 ] ++ ;
line 11:
line 12: //------NEW CLASS
line 13:         Point  lowerLeft  =  new  Point ( )  ;
line 14:
line 15: //------ERROR CONDITION
*********Line 16: char 16 has unexpected MUL
line 16:         while ( * * / a ++ ) {
line 17:             print ( "error!!" ) ;
line 18:         }
line 19: //------CLASS DECLARE
line 20:         class  Point  {
line 21:             int  x ,  y ,  z ;
line 22:         }
========= "Point" is a unused identifier.
line 23:     }
line 24:
========= "sum" is a unused identifier.
line 25: }
```