

B112040003 張景旭

Lex 版本: flex 2.6.4

Yacc 版本: bison (GNU Bison) 3.8.2

OS: ubuntu 22.04

執行方式：

1. Compile :

```
$ make clean
```

```
$ make
```

2. Execute:

```
$ ./myjavac <inputfile>
```

```
e.g. $ ./myjavac ./test1.java
```

你/妳如何處理這份規格書上的問題

- 在文法 `<Program> ::= class <ClassName> { <MemberList> }` 中，

定義 Java 類別的基本架構。包含：

- class 關鍵字遺漏
- { 或 } 括號缺漏
- 類別主體 `<MemberList>` 為空時，視情況顯示提示
- 不合法的類別名稱識別字
- 漏寫分號、括號不對稱、參數遺漏、重複宣告等問題，讓語法分析器具備基礎的錯誤恢復能力。

這些錯誤會被攔截處理，印出錯誤訊息，且不會導致 parser 直接終止。

- 在文法 $\langle \text{Declaration} \rangle ::= \langle \text{Type} \rangle \langle \text{IDList} \rangle ;$ 中，檢查變數宣告的格式是否正確，包括漏寫分號、錯用 $=$ 或 $::=$ 賦值符號等錯誤情況，並透過語意分析階段回報具體錯誤訊息。
- 在 $\langle \text{IDList} \rangle ::= \text{ID} \mid \langle \text{IDList} \rangle , \text{ID}$ 中，結合 symbol table 功能，對每個出現的 ID 進行重複性檢查，若發現相同變數名稱重複宣告，將產生錯誤提示，並持續分析後續語法。
- 在賦值運算 $\langle \text{Assignment} \rangle ::= \langle \text{LHS} \rangle = \langle \text{Expression} \rangle$ 中，偵測是否使用了非 Java 標準賦值符號（例如 $::=$ ），若出現類 Pascal 語法，將回報不合法的賦值操作。
- 在邏輯與算術運算式中，進行基本型別一致性檢查。例如： $\text{int} + \text{boolean}$ 或 $\text{float} \&\& \text{string}$ 等情況，將產生型別錯誤訊息。此外，也會對非法常數格式（如 $12., .34e+, 1.2.3$ ）進行過濾。
- 在條件語句 $\langle \text{IfStmt} \rangle ::= \text{if} (\langle \text{Expr} \rangle) \langle \text{Stmt} \rangle$ 與 $\langle \text{WhileStmt} \rangle ::= \text{while} (\langle \text{Expr} \rangle) \langle \text{Stmt} \rangle$ 中，檢查括號是否完整封閉，並提示漏寫右括號或語法結構不完整的情況。
- 在 $\langle \text{MethodDecl} \rangle$ 中，檢查是否有缺漏的引數括號或回傳型別，並對未正確結束的 method block（例如缺少 $\}$ ）提出錯誤提示。

遇到問題

1. lex 把 多行 comment 都視做同一行字串，對於多行 comment 要分成標示出不同不同行號需透過字串行數及陣列處理
2. lex 把 ' ' 和 '\t' 都視做長度為一，所以針對 '\t' 計算位置會有問題，需另行處理
3. Yacc 在做 recovery 時的訊息判別，會因為 comment 在其上下列，造成行號標示錯誤，必須另行判斷
4. 可利用 yacc 指令使用特殊記號名稱 error 進行錯誤處理。設計 error recovery 時，如果此位置發生輸入錯誤，剖析器會執行 ，可以快速判別文法錯誤位置，利於程式設計

結果：

test1.java

```
edward@ubuntu3:~/compiler/myjavav3$ ./myjavac 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: }
```

test2.java

```
edward@ubuntu3:~/compiler/myjavav3$ ./myjavac 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*/
line 7:  int x ;
>'x' is a duplicate identifier.
line 8:  void clear ( )
line 9:  {
line 10:  x = 0 ;
line 11:  y = 0 ;
line 12:  }
line 13: }
```

test3.java

```
edward@ubuntu3:~/compiler/myjavav3$ ./myjavac 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 , a syntax error at 'y'
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 10, char: 12, statement without semicolon
line 11: /*Need ';' at EOL*/
line 12: int w , q
line 13: }
edward@ubuntu3:~/compiler/myjavav3$ |
```

test4.java

```
edward@ubuntu3:~/compiler/myjavav3$ ./myjavac 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' in the next line is a duplicated identifier in the current scope.***
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: }
line 18: }
line 19: class Test
line 20: {
line 21: /*Another p, but in different scopes*/
line 22: Point p = new Point ( ) ;
line 23: }
```

test5.java

```
edward@ubuntu3:~/compiler/myjavav3$ ./myjavac 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
*****Else Without If at line 10, char 4*****
line 10: else
line 11: i = 1
line 12: }
line 13: for ( x ; x < 5 ; x++ ) {
line 14: y++ ;
line 15: //-----FUNCTION CALL
line 16: x = add(x , y) ;
line 17: x = z(x , y) ;
line 18: }
line 19: }
line 20: print ( "x:" + x + "y:" + y ) ;
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 */
edward@ubuntu3:~/compiler/myjavav3$
```

test6.java

```
edward@ubuntu3:~/compiler/myjavav3$ ./myjavac 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
*****Invalid Boolean Expression at line 16,char 9 *****
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: }
line 23: }
line 24:
line 25: }
```