B093040024
郭晏涵

編譯器製作 Homework 2

Readme of Simple Java - Parser

Lex, Yacc 版本：

flex 2.6.4

bison (GNU Bison) 3.8.2

作業平台：

Ubuntu 22.04.2 LTS

執行方式：

$ make all

$ ./calc < test1.java

如何處理這份規格書上的問題：

1. **Symbol Table Implementation**

   由於需要在 yacc 的階段才能判斷出此 identifier 是屬於 variable、method name 還是 class name，因此將 symbol table 移至 yacc 實作，並且將 symbol table 分為 variable table、function table 和 class table，來儲存不同型態的 identifier。

2. **Syntactic Definitions**

   此部分於 Parser 的註解內說明。

3. **Semantic Definition**

   必須檢查同一個 scope 內，不能宣告兩個相同的的變數，而 variable、method 和 class 的宣告是分開來檢查的，也就是說同一個 scope 內可以宣告各一個相同名稱的 variable 和 method。

   為了判斷 scope 的範圍，我使用 variable scope、function scope 和 class scope 三個 array，分別記錄從 symbol table 的第幾個 element 開始是一個新的 scope，當離開一個 scope 時，會將 symbol table 中屬於此 scope 的 element 移除，並將 scope array 的最後一個 element 刪除。辨認此變數是否重複宣告的方法是檢查 symbol table 是否已存在位於此 scope 的 element，而 scope 的範圍則由 scope array 最後一個 element 得知。

4. **Error and Recovery**

Parser 在遇到沒有定義的 input 時，可使用 error 這個特殊的保留字進行錯誤處理，error 會蒐集錯誤的 input，並使用 yyerror 將錯誤訊息印出。但此方法只適用於語法錯誤，變數重複宣告或是使用到未宣告的變數等語意錯誤需要額外處理。

過程中遇到的問題：

1. **規格書相關問題**

某些作業規格書上的文法會導致 shift/reduce conflict，例如 simple statement 的文法中包含 name++和 name--，但在 factor 的文法中又出現 identifier PostfixOp，如此一來 ID ++就會同時符合兩種文法定義。

規格書上定義 method call 屬於一種 stamente，但在 factor 中也有出現 mathod invocation 的文法，同樣會造成 shift/reduce conflict。

格書上 test6.java 的輸出似乎有誤， line 17 呼叫了 z 這個 method，但先前並沒又定義 z 這個 method，只有宣告名稱為 z 的 variable，line 21 的 2 --也應該顯示語法錯誤，因此我在這部份補上 error message。

2. **Java 可以先宣告 Object 後定義 Class**

先將用來建立 object 的 class 名稱記錄在 array 中，每當離開一個 scope 時，檢查此 scope 是否有出現這些 class 的 definition。因此必須處理完所有 input 才能確定 object 的建立是否合法。

測試檔執行結果：

test1.java

```
cse132@cse132-virtual-machine:~/Documents/Compiler/yacc$ ./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: }
```

test2.java

```
cse132@cse132-virtual-machine:~/Documents/Compiler/yacc$ ./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*/
******Line 7: '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: }
line 13: }
```

test3.java

```
cse132@cse132-virtual-machine:~/Documents/Compiler/yacc$ ./calc < test3.java
line  1: /*Test file of Syntax errer: Out of symbol.  But it can go through*/
line  2: class Point {
line  3: int z ;
******Line 4: 1st 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: 1st char 8 has unexpected INT, expecting SEMI******
line 12: int w , q ;
line 13: }
```

test4.java

```
cse132@cse132-virtual-machine:~/Documents/Compiler/yacc$ ./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*/
******Line 11: '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: }
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

```
cse132@cse132-virtual-machine:~/Documents/Compiler/yacc$ ./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: 1st 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 ) ;
******Line 17: '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: 1st char 33 has unexpected DSUB, expecting SEMI or ADD or SUB******
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

```
cse132@cse132-virtual-machine:~/Documents/Compiler/yacc$ ./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: 1st 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: }
line 23: }
line 24:
line 25: }
```