

Readme

學號：B073021024

姓名：錢承

1. Lex/Yacc 版本

Flex 2.6.4

bison (GNU Bison) 3.5.1

2. 作業平台

Ubuntu 20.04.2 LTS (Linux 5.8.0-53-generic)

3. 執行方式

Zsh Terminal 輸入以下指令：

```
make ; ./demo < 檔案名稱
```

4. 處理規格書上的問題

Data Types and Declarations:

建立 `dec_list`, 包含一個或多個 `dec`(如變數、函數、`class`)。每一個宣告定義為 `dec`, 包含變數、`array`、`main`、`method(function)`、`class`、`object` 等。其中 `main` 可能無 `type`, 所以需要特別處理。

Classes and Objects:

`Class` 包含 `Fields` 和 `Method`, 兩者用 `{ }` 包起來。其中 `Fields` 可以直接使用 `dec_list`, 而且可有可無, 必須分開處理。而 `Method` 可能包含 `main`, 需要處理。

Objects:

直接建立即可, 無須特別處理。

Fields:

直接利用 `dec_list` 即可。

Methods(function):

開頭可能有 `public`、`protected`、`private`, 需要建立 `method_type` 處理。而函數的 `type` 可能有 `viod`, 需要特別處理。函數可能有 `argument`, 也可能沒有, 需要分開處理。

Compound:

可以包含 `statement`, 所以利用遞迴處理。也可能包含宣告, 可以使用 `dec_list` 處理。

Simple:

主要出現在 while 或 for 下方不需要使用 {} 包起來的 statement，可以包含 print、assignment、expressions 等等。

Conditional:

處理 if else 指令，if 裡面需要包含 bool 運算，如 <、>、<=、>= 等等。而 if 和 else 下方包含 simple 或 compound。

Loop:

包含 for 和 while。while 裡面需要包含 bool 運算，下方包含 simple 或 compound。for 分為 3 個部分：第一個，包含一連串的變數或 array 的 assignment 或宣告，我使用 idlist 和 TYPE idlist 處理。第二個，需要包含 bool 運算。第三個，變數或 array 的運算(expression)。

Return:

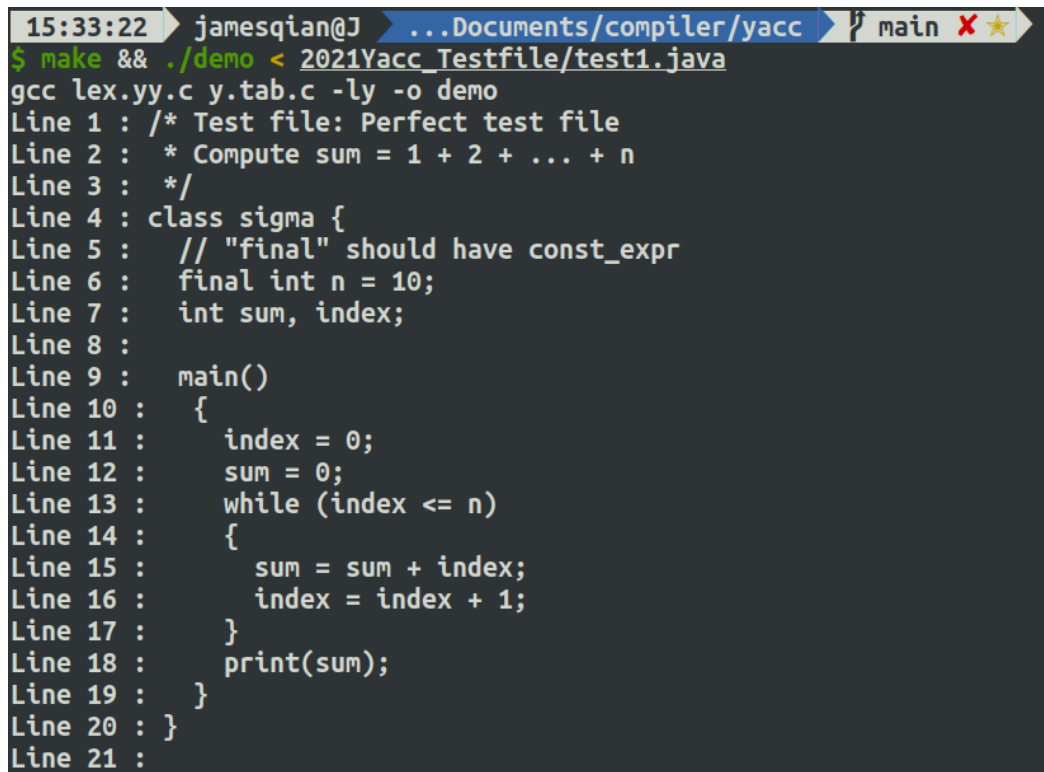
直接建立即可，無須特別處理。

method call(function call):

需要處理裡面可能包含變數、常數、運算，直接用即可。

5. 所有測試檔執行出來的結果

(1) test1.java



```
15:33:22 jamesqian@J ...Documents/compiler/yacc main X★
$ make && ./demo < 2021Yacc_Testfile/test1.java
gcc lex.yy.c y.tab.c -ly -o demo
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 : }
Line 21 :
```

(2) test2.java

```
15:33:28 jamesqian@J ...Documents/compiler/yacc ? main X★
$ make && ./demo < 2021Yacc_Testfile/test3.java
gcc lex.yy.c y.tab.c -ly -o demo
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 :     int x y ;
Line 4, 1st char: 12, a syntax error at "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 :     int w,q;
Line 12, 1st char: 8, a syntax error at "int"
Line 13 : }
Line 14 :
```

(3) test3.java

```
15:33:25 jamesqian@J ...Documents/compiler/yacc ? main X★
$ make && ./demo < 2021Yacc_Testfile/test2.java
gcc lex.yy.c y.tab.c -ly -o demo
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 : }
Line 14 :
```

(4) test4.java

```
15:33:30 jamesqian@J ...Documents/compiler/yacc main X★
$ make && ./demo < 2021Yacc_Testfile/test4.java
gcc lex.yy.c y.tab.c -ly -o demo
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 :         char x;
> 'x' is a duplicate identifier.
Line 12 :     {
Line 13 :         boolean w;
Line 14 :     }
Line 15 :         /*Another w in the same scope*/
Line 16 :         int w;
> 'w' is a duplicate identifier.
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 : }
Line 24 :
```

(5) test5.java

```
15:33:37 jamesqian@J ...Documents/compiler/yacc main X★
$ make && ./demo < 2021Yacc_Testfile/test5.java
gcc lex.yy.c y.tab.c -ly -o demo
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 :                 else
Line 10, 1st char: 21, syntax error at "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 :                 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 */
Line 27 :
```

(6) test6.java

```
15:33:40 jamesqian@J ...Documents/compiler/yacc main X ★
$ make && ./demo < 2021Yacc_Testfile/test6.java
gcc lex.yy.c y.tab.c -ly -o demo
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 :         while(*
Line 16, 1st char: 16, syntax error at "*".
*/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 :
Line 26 : }
Line 27 :
```