read me

complier hw2

Lex 版本

```
flex 2.6.4
```

bison 版本

```
bison (GNU Bison) 3.5.1
```

作業平台

```
Ubuntu 20.04.2 LTS
```

執行方式

- 請按以下順序
    1. make
    2. make test(如果要一次餵入公開測資)

- make file

```
all:    clean y.tab.c lex.yy.c
    gcc lex.yy.c y.tab.c -ly -lfl -o calc

y.tab.c:
    bison -y -d B073040025.y

lex.yy.c:
    flex B073040025.l
test:
    ./calc < test1.java
    @echo
    ./calc < test2.java
    @echo
    ./calc < test3.java
    @echo
    ./calc < test4.java
    @echo
    ./calc < test5.java
    @echo
    ./calc < test6.java

clean:
    rm -f calc lex.yy.c y.tab.c y.tab.h

```

如何處理這份規格書上的問題

1. 按照作業pdf給的架構，並參考java.doc的架構，寫出自己對這作業理解的架構。
2. 承上，合理編排各種statment的grammar及token位置。
3. 將每個區域放置printf(區域名)的程式碼，以便追蹤tree的建立。
4. 多次詢問助教相關流程。

寫這個作業所遇到的問題

1. 不懂流程，及變數傳遞導致無從開始。
    - 尋問助教，及上網找尋相關知識。
2. newline此一token造成，配對grammar上的困難。
    - 以print方式尋找錯誤點，並以所寫lines架構，統一規格以不混淆。
3. 在寫method declare 的文法與variable declare的文法前綴衝突，導致無法配對。
    - 將衝突的前綴提出，包成另一個grammar，後將衝突的前綴改成一個，所寫的新grammar
4. 只有寫對變數的++or--，忘了寫對數字的。

輸出

```
./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:  }

./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*/
>>>>>>>>>>: duplicate declare at line 7, Var name "x"
line 7: int x ;
line 8: void clear (  )
line 9: {
line 10: x = 0 ;
line 11: y = 0 ;
line 12:  }
line 13:  }

./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 ;
>>>>>>>>>>: declare format error at line 4, char 5(not counting space)
line 4: int xy ;
line 5: /*Need  ',' before y*/
line 6: float w ;
line 7:  }
line 8: class Test {
line 9: int d ;
>>>>>>>>>>: need ';' at EOL at line 10, char 17(not counting space)
line 10: Point p = new Point ( )
line 11: /*Need ';' at EOL*/
line 12: int w , q ;
line 13:  }
```

```
./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*/
>>>>>>>>>>: duplicate declare at line 11, Var name "x"
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:  }
```

```
./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
>>>>>>>>>>: else without if at line 10, char 4(not counting space)
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: 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 */
```

```
./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
>>>>>>>>>>: error boolean expression at line 16, char 13(not counting space)
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:  }
```