

Lex版本:

flex 2.6.4

作業平台:

Ubuntu 22.04

執行方式:

1. 先進到包含作業檔案的資料夾
2. make(若出現make: Nothing to be done for 'all'.的訊息, 輸入make clean後, 再重新輸入一次make)
3. 觀看開放測資之結果
./B113040047 < TestFile_Lab1_2025/Test1.java
./B113040047 < TestFile_Lab1_2025/Test2.java
./B113040047 < TestFile_Lab1_2025/Test3.java
4. 觀看自訂測資之結果
./B113040047 < TestFile_Lab1_2025/OwnTest1.java (針對string的測資)
./B113040047 < TestFile_Lab1_2025/OwnTest2.java (針對ID的測資)
./B113040047 < TestFile_Lab1_2025/OwnTest3.java (針對integer的測資)
./B113040047 < TestFile_Lab1_2025/OwnTest4.java (針對+/-的測資)

你如何處理這份規格書上的問題:

Lexical Definitions

- 會傳給parser的tokens:
1. Symbols
以regular expression: `[,;:\(\)\[\]\{\}]`進行判斷, 並輸出判斷到的symbol資訊。
 2. Arithmetic, Relational, and Logical Operators
以regular expression: `\+(\+)?|-(-)?|*|V|%| |=|<|<=|>|>=|!=|==|&|\||!` 進行判斷, 並輸出判斷到的operator資訊。
 3. Reserved words
以regular expression:
`boolean|break|bute|case|catch|class|const|continue|default|do|double|else|extends|false|final|finally|float|for|if|implements|int|long|main|new|print|private|protected|public|return|short|static|string|switch|this|true|try|void|while` 進行判斷。並輸出判斷到的reserved words資訊。
 4. Identifiers
以regular expression: `[_a-zA-Z0-9]*`進行判斷, 其中, 會把不合法的ID也讀入, 例如1a, 因此會在Lex Rules的部份額外進行判斷, 把不合法的ID選出, 最後輸出錯誤訊息或是判斷到的ID資訊。
 5. Integer Constants
以regular expression: `[+-]?([0-9]*|0x[0-9A-F]*)`進行判斷, 能夠接收八進位、十進位、十六進位的整數。針對overflow進行額外判斷, 若數值範圍超過2147483647~-2147483648, 則視為overflow, 出現錯誤訊息。也針對數字前方的'-'是減號還是負號('+'也進行相同的判斷)進行額外的判斷, 作法是使用"isOperator"變數紀錄前面一個讀

到的內容是否為ID、integer、float或是')', 若符合這四種內容, 則將其視為減號, 否則視為負號。處理完後, 輸出錯誤訊息或是判斷到的integer constant資訊。

6. Float Constants

以regular expression:

`[+-]?(((([0-9]+)\.([0-9]+))|(\.([0-9]+))|([0-9]+\.))([0-9]+)([eE][+-]?[0-9]+)?(f|d)?`進行判斷, 此regular expression會將double和float都視為float, 並且支援科學記號表示法。另外, 和integer constants相同, 會判斷數字前方的'-'是減號還是負號('+'也進行相同的判斷)。處理完後, 輸出錯誤訊息或是判斷到的float constant資訊。

7. String Constants

以regular expression: `"\.*"`進行判斷, 並且再額外判斷string內的雙引號前面是否加上反斜線, 若沒有, 則視為不合法的string, 並紀錄哪一個雙引號前面未加上反斜線, 而若有加上反斜線, 則會把該反斜線去除, 讓string能正常表示雙引號。另外使用regular expression: `(\.*\)|(\.*\\")|(\.*"[^"]*"*)`來捕捉部份非法字串, 主要針對兩種錯誤string進行判斷, 包含以單引號標示的string, 以及缺少第二個雙引號的string。處理完後, 輸出錯誤訊息或是判斷到的string constant資訊。

● 不會傳給parser的tokens:

1. Whitespace

以regular expression: `[\t]`進行判斷。不輸出任何資訊, 僅紀錄字元數量增加。

2. Comments

以regular expression: `*.*\\V|*.*(\\n.*)*\\V|\\V.*`進行判斷, 並輸出判斷到的Comment, 另外, 由於comment可能出現跨行的情況, 因此若comment內存在'\\n', 會將目前進行判斷的行數額外加一。

Recovery:

以下的幾種Recovery的處理方式都寫於Lexical Definitions的說明中。

- 為了盡可能處理所有的input, 上述的token判斷會額外讀入不符合規定的tokens, 例如identifier和string constants都進行的此處理, 並在lex rules中進行處理, 提出錯誤token的資訊。
- 針對符合規定, 但不符合該token類型的基本要求的token進行判斷, 例如integer constants若發生overflow的狀況, 雖然這個token符合integer constants的規定, 但仍會將其視為不合法的token
- 針對'+'和'-', 對於是正號或是加號以及是減號或是負號進行判斷。

Symbol Table:

使用hash的方式除存identifier。

table中儲存的ID是一個struct的結構, 裡面包含三個資訊:

1. order: 代表此ID是第幾個被發現的ID。
2. ID: 代表此ID的內容。
3. next: 代表此ID的下一個order的ID存於table中的哪個位置。

hash function的計算方式會取ID的第一個字元的ASCII*80, 將其放到table中此值的位置。

若發生collision, 選擇linear probing來解決, 即是到發生collision的位置的下一格檢查是否已放入ID, 若無, 則放到該位置, 若有, 則再檢查下一格。

以下說明symbol table的四個function之實現方式:

1. create(): 分配記憶體空間給symbol table

- lookup(s): 到ID的第一個字元的ASCII*80的位置尋找是否有符合的字串, 若沒有, 則到下一格尋找, 直到尋找到符合的字串, 或是將symbol table的所有位置都尋找過。
- insert(s): 先判斷s是否存在於symbol table中, 若不存在, 則將其放到ASCII*80的位置, 若發生collision, 則以linear probing的方式尋找下一個格子。同時會將前一個放入table的Identifier struct之next改為s所放入的位置。
- dump(): 配合Identifier struct所儲存的next資訊, 將ID依照放入table的順序一一印出。

你寫這個作業所遇到的問題

- 為了提升放入以及搜尋的速度, 使用hash table作為symbol table, 但在輸出所有ID時, 無法依照儲存的順序輸出。最後是將下一個被除存的ID於symbol table的位置也存入symbol table才解決此問題。
- 負號和減號的判斷因為不知道哪些情況會被視為負號, 哪些情況又該視為減號, 使用java online compiler嘗試了非常多種狀況, 最後才整理出前述的幾種會被視為減號的可能。

測試檔的執行結果:

Test1.java:

```
chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/lab1$ ./B113040047 < TestFile_Lab1_2025/Test1.java
Line: 2, 1st char: 1, "public" is a "reserved word".
Line: 2, 1st char: 8, "class" is a "reserved word".
Line: 2, 1st char: 14, "Test1" is a "ID".
Line: 2, 1st char: 20, "{" is a "symbol".
Line: 3, 1st char: 5, "public" is a "reserved word".
Line: 3, 1st char: 12, "static" is a "reserved word".
Line: 3, 1st char: 19, "int" is a "reserved word".
Line: 3, 1st char: 23, "add" is a "ID".
Line: 3, 1st char: 26, "(" is a "symbol".
Line: 3, 1st char: 27, "int" is a "reserved word".
Line: 3, 1st char: 31, "a" is a "ID".
Line: 3, 1st char: 32, "," is a "symbol".
Line: 3, 1st char: 34, "int" is a "reserved word".
Line: 3, 1st char: 38, "b" is a "ID".
Line: 3, 1st char: 39, ")" is a "symbol".
Line: 3, 1st char: 41, "{" is a "symbol".
Line: 4, 1st char: 9, "return" is a "reserved word".
Line: 4, 1st char: 16, "a" is a "ID".
Line: 4, 1st char: 18, "+" is an "operator".
Line: 4, 1st char: 20, "b" is a "ID".
Line: 4, 1st char: 21, ";" is a "symbol".
Line: 5, 1st char: 5, "}" is a "symbol".
Line: 7, 1st char: 5, "public" is a "reserved word".
Line: 7, 1st char: 12, "static" is a "reserved word".
Line: 7, 1st char: 19, "void" is a "reserved word".
Line: 7, 1st char: 24, "main" is a "reserved word".
Line: 7, 1st char: 28, "(" is a "symbol".
Line: 7, 1st char: 29, ")" is a "symbol".
Line: 7, 1st char: 31, "{" is a "symbol".
Line: 9, 1st char: 9, "int" is a "reserved word".
Line: 9, 1st char: 13, "c" is a "ID".
Line: 9, 1st char: 14, ";" is a "symbol".
Line: 10, 1st char: 9, "int" is a "reserved word".
Line: 10, 1st char: 13, "a" is a "ID".
Line: 10, 1st char: 15, "=" is an "operator".
Line: 10, 1st char: 17, "5" is a "integer".
Line: 10, 1st char: 18, ";" is a "symbol".
Line: 11, 1st char: 9, "c" is a "ID".
Line: 11, 1st char: 11, "=" is an "operator".
Line: 11, 1st char: 13, "add" is a "ID".
Line: 11, 1st char: 16, "(" is a "symbol".
Line: 11, 1st char: 17, "a" is a "ID".
Line: 11, 1st char: 18, "," is a "symbol".
Line: 11, 1st char: 20, "10" is a "integer".
Line: 11, 1st char: 22, ")" is a "symbol".
Line: 11, 1st char: 23, ";" is a "symbol".
Line: 12, 1st char: 9, "if" is a "reserved word".
Line: 12, 1st char: 12, "(" is a "symbol".
Line: 12, 1st char: 13, "c" is a "ID".
Line: 12, 1st char: 15, ">" is an "operator".
Line: 12, 1st char: 17, "10" is a "integer".
```

```

Line: 12, 1st char: 19, ")" is a "symbol".
Line: 13, 1st char: 13, "print" is a "reserved word".
Line: 13, 1st char: 18, "(" is a "symbol".
Line: 13, 1st char: 19, "c = " is a "string".
Line: 13, 1st char: 26, "+" is an "operator".
Line: 13, 1st char: 28, "-" is an "operator".
Line: 13, 1st char: 29, "c" is a "ID".
Line: 13, 1st char: 30, ")" is a "symbol".
Line: 13, 1st char: 31, ";" is a "symbol".
Line: 14, 1st char: 9, "else" is a "reserved word".
Line: 15, 1st char: 13, "print" is a "reserved word".
Line: 15, 1st char: 18, "(" is a "symbol".
Line: 15, 1st char: 19, "c" is a "ID".
Line: 15, 1st char: 20, ")" is a "symbol".
Line: 15, 1st char: 21, ";" is a "symbol".
Line: 16, 1st char: 9, "print" is a "reserved word".
Line: 16, 1st char: 14, "(" is a "symbol".
Line: 16, 1st char: 15, "Hello World" is a "string".
Line: 16, 1st char: 28, ")" is a "symbol".
Line: 16, 1st char: 29, ";" is a "symbol".
Line: 18, 1st char: 5, "}" is a "symbol".
Line: 20, 1st char: 1, "}" is a "symbol".

```

Symbol table:

```

1. Test1
2. add
3. a
4. b
5. c

```

Test2.java:

```

chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/lab1$ ./B113040047 < TestFile_Lab1_2025/Test2.java
" is a "comment".: 1, "/* this is a comment // line */ /* with /* delimiters */ before the end
Line: 3, 1st char: 1, "public" is a "reserved word".
Line: 3, 1st char: 8, "class" is a "reserved word".
Line: 3, 1st char: 14, "Test2" is a "ID".
Line: 3, 1st char: 20, "{" is a "symbol".
Line: 4, 1st char: 5, "int" is a "reserved word".
Line: 4, 1st char: 9, "i" is a "ID".
Line: 4, 1st char: 11, "=" is an "operator".
Line: 4, 1st char: 13, "-100" is a "integer".
Line: 4, 1st char: 17, ";" is a "symbol".
Line: 5, 1st char: 5, "double" is a "reserved word".
Line: 5, 1st char: 12, "d" is a "ID".
Line: 5, 1st char: 14, "=" is an "operator".
Line: 5, 1st char: 16, "12.25e+6" is a "float".
Line: 5, 1st char: 24, ";" is a "symbol".
Line: 7, 1st char: 5, "public" is a "reserved word".
Line: 7, 1st char: 12, "static" is a "reserved word".
Line: 7, 1st char: 19, "void" is a "reserved word".
Line: 7, 1st char: 24, "main" is a "reserved word".
Line: 7, 1st char: 28, "(" is a "symbol".
Line: 7, 1st char: 29, ")" is a "symbol".
Line: 7, 1st char: 31, "{" is a "symbol".
Line: 8, 1st char: 1, "/* this is a comment // line with some /* and
// delimiters */" is a "comment".
Line: 10, 1st char: 5, "}" is a "symbol".
Line: 11, 1st char: 1, "}" is a "symbol".

```

Symbol table:

```

1. Test2
2. i
3. d

```

Test3.java:

```
chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/Lab1$ ./B113040047 < TestFile_Lab1_2025/Test3.java
Line: 2, 1st char: 1, "public" is a "reserved word".
Line: 2, 1st char: 8, "class" is a "reserved word".
Line: 2, 1st char: 14, "Test3" is a "ID".
Line: 2, 1st char: 20, "{" is a "symbol".
Line: 3, 1st char: 5, "int" is a "reserved word".
Line: 3, 1st char: 9, "A" is a "ID".
Line: 3, 1st char: 10, ";" is a "symbol".
Line: 4, 1st char: 5, "int" is a "reserved word".
Line: 4, 1st char: 9, "a" is a "ID".
Line: 5, 1st char: 5, "double" is a "reserved word".
Line: 5, 1st char: 12, "b" is a "ID".
Line: 5, 1st char: 13, ";" is a "symbol".
Line: 6, 1st char: 5, "double" is a "reserved word".
Line: 6, 1st char: 12, "A" is a "ID".
Line: 6, 1st char: 13, ";" is a "symbol".
Line: 8, 1st char: 5, "public" is a "reserved word".
Line: 8, 1st char: 12, "Test3" is a "ID".
Line: 8, 1st char: 17, "(" is a "symbol".
Line: 8, 1st char: 18, ")" is a "symbol".
Line: 8, 1st char: 20, "{" is a "symbol".
Line: 9, 1st char: 9, "a" is a "ID".
Line: 9, 1st char: 11, "=" is an "operator".
Line: 9, 1st char: 13, "1" is a "integer".
Line: 9, 1st char: 14, ";" is a "symbol".
Line: 10, 1st char: 9, "A" is a "ID".
Line: 10, 1st char: 11, "=" is an "operator".
Line: 10, 1st char: 13, "2" is a "integer".
Line: 10, 1st char: 14, ";" is a "symbol".
Line: 11, 1st char: 9, "b" is a "ID".
Line: 11, 1st char: 11, "=" is an "operator".
Line: 11, 1st char: 13, "-1.2" is a "float".
Line: 11, 1st char: 17, ";" is a "symbol".
Line: 12, 1st char: 5, "}" is a "symbol".
Line: 13, 1st char: 1, "}" is a "symbol".
```

Symbol table:

1. Test3
2. A
3. a
4. b

OwnTest1.java:

```
chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/lab1/final$ ./B113040047 < TestFile_Lab1_2025/OwnTest1.java
Line: 1, 1st char: 1, "// used to test invalid string" is a "comment".
Line: 2, 1st char: 1, "public" is a "reserved word".
Line: 2, 1st char: 8, "class" is a "reserved word".
Line: 2, 1st char: 14, "OwnTest1" is an "ID".
Line: 2, 1st char: 23, "{" is a "symbol".
Line: 3, 1st char: 5, "public" is a "reserved word".
Line: 3, 1st char: 12, "static" is a "reserved word".
Line: 3, 1st char: 19, "void" is a "reserved word".
Line: 3, 1st char: 24, "main" is a "reserved word".
Line: 3, 1st char: 28, "(" is a "symbol".
Line: 3, 1st char: 29, ")" is a "symbol".
Line: 3, 1st char: 31, "{" is a "symbol".
Line: 4, 1st char: 9, "string" is a "reserved word".
Line: 4, 1st char: 16, "o" is an "ID".
Line: 4, 1st char: 18, "=" is an "operator".
Line: 4, 1st char: 20, "te"sst" is a "string".
Line: 4, 1st char: 28, ";" is a "symbol".
Line: 5, 1st char: 9, "string" is a "reserved word".
Line: 5, 1st char: 16, "a" is an "ID".
Line: 5, 1st char: 18, "=" is an "operator".
""test1'" is a wrong string. Cannot use (') as the quote of a string
Line: 5, 1st char: 27, ";" is a "symbol".
Line: 6, 1st char: 9, "string" is a "reserved word".
Line: 6, 1st char: 16, "b" is an "ID".
Line: 6, 1st char: 18, "=" is an "operator".
""test2\" is a wrong string. The quoting is not closed.
Line: 6, 1st char: 28, ";" is a "symbol".
Line: 7, 1st char: 9, "string" is a "reserved word".
Line: 7, 1st char: 16, "c" is an "ID".
Line: 7, 1st char: 18, "=" is an "operator".
"tes"t3" is a wrong string. Unvalid syntax of '"' (Position: 3).
If you want to include '"' in the string, you have to add '\' immediately before it.
Line: 7, 1st char: 28, ";" is a "symbol".
Line: 8, 1st char: 9, "string" is a "reserved word".
Line: 8, 1st char: 16, "d" is an "ID".
Line: 8, 1st char: 18, "=" is an "operator".
""test4;" is a wrong string. The quoting is not closed.
Line: 9, 1st char: 9, "string" is a "reserved word".
Line: 9, 1st char: 16, "e" is an "ID".
Line: 9, 1st char: 18, "=" is an "operator".
Line: 9, 1st char: 20, "test5" is a "string".
Line: 9, 1st char: 27, ";" is a "symbol".
Line: 10, 1st char: 5, "}" is a "symbol".
Line: 12, 1st char: 1, "}" is a "symbol".

Symbol table:
1. OwnTest1
2. o
3. a
4. b
5. c
6. d
7. e
```

```
1 // used to test invalid string
2 public class OwnTest1 {
3     public static void main() {
4         string o = "te\"st";
5         string a = 'test1';
6         string b = "test2\";
7         string c = "tes"t3";
8         string d = "test4;
9         string e = "test5";
10    }
11
12 }
```

OwnTest2.java:

```
chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/Lab1$ ./B113040047 < TestFile_Lab1_2025/OwnTest2.java
Line: 1, 1st char: 1, "// used to test invalid identifier" is a "comment".
Line: 2, 1st char: 1, "public" is a "reserved word".
Line: 2, 1st char: 8, "class" is a "reserved word".
Line: 2, 1st char: 14, "OwnTest2" is an "ID".
Line: 2, 1st char: 23, "{" is a "symbol".
Line: 3, 1st char: 5, "public" is a "reserved word".
Line: 3, 1st char: 12, "static" is a "reserved word".
Line: 3, 1st char: 19, "void" is a "reserved word".
Line: 3, 1st char: 24, "main" is a "reserved word".
Line: 3, 1st char: 28, "(" is a "symbol".
Line: 3, 1st char: 29, ")" is a "symbol".
Line: 3, 1st char: 31, "{" is a "symbol".
Line: 4, 1st char: 9, "int" is a "reserved word".
Line: 4, 1st char: 13, "abcvsdcgsdfg" is an "ID".
Line: 4, 1st char: 25, ";" is a "symbol".
Line: 5, 1st char: 9, "int" is a "reserved word".
Line: 5, 1st char: 13, "_asdfks123fs$$" is an "ID".
Line: 5, 1st char: 27, ";" is a "symbol".
Line: 6, 1st char: 9, "int" is a "reserved word".
12kjcvijsd is not a valid identifier, because it begin with a digit
Line: 6, 1st char: 24, ";" is a "symbol".
Line: 7, 1st char: 5, "}" is a "symbol".
Line: 9, 1st char: 1, "}" is a "symbol".
```

```
Symbol table:
1. OwnTest2
2. abcvsdcgsdfg
3. _asdfks123fs$$
```

```
1 // used to test invalid identifier
2 public class OwnTest2 {
3     public static void main() {
4         int abcvsdcgsdfg;
5         int _asdfks123fs$$;
6         int 12kjcvijsd;
7     }
8
9 }
```

OwnTest3.java:

```
chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/Lab1$ ./B113040047 < TestFile_Lab1_2025/OwnTest3.java
Line: 1, 1st char: 1, "// used to test invalid integer" is a "comment".
Line: 2, 1st char: 1, "public" is a "reserved word".
Line: 2, 1st char: 8, "class" is a "reserved word".
Line: 2, 1st char: 14, "OwnTest3" is an "ID".
Line: 2, 1st char: 23, "{" is a "symbol".
Line: 3, 1st char: 5, "public" is a "reserved word".
Line: 3, 1st char: 12, "static" is a "reserved word".
Line: 3, 1st char: 19, "void" is a "reserved word".
Line: 3, 1st char: 24, "main" is a "reserved word".
Line: 3, 1st char: 28, "(" is a "symbol".
Line: 3, 1st char: 29, ")" is a "symbol".
Line: 3, 1st char: 31, "{" is a "symbol".
Line: 4, 1st char: 9, "int" is a "reserved word".
Line: 4, 1st char: 13, "num1" is an "ID".
Line: 4, 1st char: 18, "=" is an "operator".
Line: 4, 1st char: 20, "2147483647" is an "integer".
Line: 4, 1st char: 30, ";" is a "symbol".
Line: 5, 1st char: 9, "int" is a "reserved word".
Line: 5, 1st char: 13, "num2" is an "ID".
Line: 5, 1st char: 18, "=" is an "operator".
2147483648 is not a valid integer, because it overflows
Line: 5, 1st char: 30, ";" is a "symbol".
Line: 6, 1st char: 9, "int" is a "reserved word".
Line: 6, 1st char: 13, "num3" is an "ID".
Line: 6, 1st char: 18, "=" is an "operator".
2147483649 is not a valid integer, because it overflows
Line: 6, 1st char: 30, ";" is a "symbol".
Line: 7, 1st char: 9, "int" is a "reserved word".
Line: 7, 1st char: 13, "num4" is an "ID".
```

```

Line: 7, 1st char: 18, "=" is an "operator".
Line: 7, 1st char: 20, "-2147483647" is an "integer".
Line: 7, 1st char: 31, ";" is a "symbol".
Line: 8, 1st char: 9, "int" is a "reserved word".
Line: 8, 1st char: 13, "num5" is an "ID".
Line: 8, 1st char: 18, "=" is an "operator".
Line: 8, 1st char: 20, "-2147483648" is an "integer".
Line: 8, 1st char: 31, ";" is a "symbol".
Line: 9, 1st char: 9, "int" is a "reserved word".
Line: 9, 1st char: 13, "num6" is an "ID".
Line: 9, 1st char: 18, "=" is an "operator".
-2147483649 is not a valid integer, because it overflows
Line: 9, 1st char: 31, ";" is a "symbol".
Line: 10, 1st char: 9, "int" is a "reserved word".
Line: 10, 1st char: 13, "num7" is an "ID".
Line: 10, 1st char: 18, "=" is an "operator".
-1111111111 is not a valid integer, because it overflows
Line: 10, 1st char: 32, ";" is a "symbol".
Line: 11, 1st char: 9, "int" is a "reserved word".
Line: 11, 1st char: 13, "num8" is an "ID".
Line: 11, 1st char: 18, "=" is an "operator".
Line: 11, 1st char: 20, "-1" is an "integer".
Line: 11, 1st char: 22, ";" is a "symbol".
Line: 12, 1st char: 9, "int" is a "reserved word".
Line: 12, 1st char: 13, "num9" is an "ID".
Line: 12, 1st char: 18, "=" is an "operator".
Line: 12, 1st char: 20, "+1" is an "integer".
Line: 12, 1st char: 22, ";" is a "symbol".
Line: 13, 1st char: 5, "}" is a "symbol".
Line: 15, 1st char: 1, "}" is a "symbol".

```

Symbol table:

1. OwnTest3
2. num1
3. num2
4. num3
5. num4
6. num5
7. num6
8. num7
9. num8
10. num9

```

1  // used to test invalid integer
2  public class OwnTest3 {
3      public static void main() {
4          int num1 = 2147483647;
5          int num2 = 2147483648;
6          int num3 = 2147483649;
7          int num4 = -2147483647;
8          int num5 = -2147483648;
9          int num6 = -2147483649;
10         int num7 = -1111111111;
11         int num8 = -1;
12         int num9 = +1;
13     }
14
15 }

```


OwnTest4.java

```
chenze@chenze-TravelMate-P238-M:~/Compiler_Homework/lab1/final$ ./B113040047 < TestFile_Lab1_2025/OwnTest4.java
```

```
Line: 1, 1st char: 1, "// used to test '-' and '+' is a "comment".
```

```
Line: 2, 1st char: 1, "public" is a "reserved word".
```

```
Line: 2, 1st char: 8, "class" is a "reserved word".
```

```
Line: 2, 1st char: 14, "OwnTest4" is an "ID".
```

```
Line: 2, 1st char: 23, "{" is a "symbol".
```

```
Line: 3, 1st char: 5, "public" is a "reserved word".
```

```
Line: 3, 1st char: 12, "static" is a "reserved word".
```

```
Line: 3, 1st char: 19, "void" is a "reserved word".
```

```
Line: 3, 1st char: 24, "main" is a "reserved word".
```

```
Line: 3, 1st char: 28, "(" is a "symbol".
```

```
Line: 3, 1st char: 29, ")" is a "symbol".
```

```
Line: 3, 1st char: 31, "{" is a "symbol".
```

```
Line: 4, 1st char: 9, "int" is a "reserved word".
```

```
Line: 4, 1st char: 13, "a" is an "ID".
```

```
Line: 4, 1st char: 15, "=" is an "operator".
```

```
Line: 4, 1st char: 17, "-1" is an "integer".
```

```
Line: 4, 1st char: 19, ";" is a "symbol".
```

```
Line: 5, 1st char: 9, "int" is a "reserved word".
```

```
Line: 5, 1st char: 13, "b" is an "ID".
```

```
Line: 5, 1st char: 15, "=" is an "operator".
```

```
Line: 5, 1st char: 17, "1" is an "integer".
```

```
Line: 5, 1st char: 18, "-" is a "operator".
```

```
Line: 5, 1st char: 19, "1" is an "integer".
```

```
Line: 5, 1st char: 20, ";" is a "symbol".
```

```
Line: 6, 1st char: 9, "int" is a "reserved word".
```

```
Line: 6, 1st char: 13, "c" is an "ID".
```

```
Line: 6, 1st char: 15, "=" is an "operator".
```

```
Line: 6, 1st char: 17, "+123" is an "integer".
```

```
Line: 6, 1st char: 21, ";" is a "symbol".
```

```
Line: 7, 1st char: 9, "int" is a "reserved word".
```

```
Line: 7, 1st char: 13, "d" is an "ID".
```

```
Line: 7, 1st char: 15, "=" is an "operator".
```

```
Line: 7, 1st char: 17, "-423" is an "integer".
```

```
Line: 7, 1st char: 21, "+" is a "operator".
```

```
Line: 7, 1st char: 22, "5" is an "integer".
```

```
Line: 7, 1st char: 23, ";" is a "symbol".
```

```
Line: 8, 1st char: 9, "int" is a "reserved word".
```

```
Line: 8, 1st char: 13, "e" is an "ID".
```

```
Line: 8, 1st char: 15, "=" is an "operator".
```

```
Line: 8, 1st char: 17, "(" is a "symbol".
```

```
Line: 8, 1st char: 18, "+12" is an "integer".
```

```
Line: 8, 1st char: 21, "-" is a "operator".
```

```
Line: 8, 1st char: 22, "34" is an "integer".
```

```
Line: 8, 1st char: 24, ")" is a "symbol".
```

```
Line: 8, 1st char: 25, "-" is a "operator".
```

```
Line: 8, 1st char: 26, "56" is an "integer".
```

```
Line: 8, 1st char: 28, ";" is a "symbol".
```

```
Line: 9, 1st char: 9, "int" is a "reserved word".
```

```
Line: 9, 1st char: 13, "f" is an "ID".
```

```
Line: 9, 1st char: 15, "=" is an "operator".
```

```
Line: 9, 1st char: 17, "e" is an "ID".
```

```
Line: 9, 1st char: 18, "+" is a "operator".
```

```
Line: 9, 1st char: 19, "65" is an "integer".
```

```
Line: 9, 1st char: 21, ";" is a "symbol".
```

```
Line: 10, 1st char: 9, "float" is a "reserved word".
```

```
Line: 10, 1st char: 15, "g" is an "ID".
```

```
Line: 10, 1st char: 17, "=" is an "operator".
```

```
Line: 10, 1st char: 19, "-123.43e23" is a "float".
```

```
Line: 10, 1st char: 29, "+" is a "operator".
```

```
Line: 10, 1st char: 30, "12.34" is a "float".
```

```
Line: 10, 1st char: 35, ";" is a "symbol".
```

```
Line: 11, 1st char: 9, "float" is a "reserved word".
```

```
Line: 11, 1st char: 15, "h" is an "ID".
```

```
Line: 11, 1st char: 17, "=" is an "operator".
```

```
Line: 11, 1st char: 19, "(" is a "symbol".
```

```
Line: 11, 1st char: 20, "+12.2" is a "float".
```

```
Line: 11, 1st char: 25, "-" is a "operator".
```

```
Line: 11, 1st char: 26, "34.21f" is a "float".
```

```
Line: 11, 1st char: 32, ")" is a "symbol".
```

```
Line: 11, 1st char: 33, "-" is a "operator".
```

```
Line: 11, 1st char: 34, "56.21e-465" is a "float".
```

```
Line: 11, 1st char: 44, ";" is a "symbol".
```

```
Line: 12, 1st char: 5, "}" is a "symbol".
```

```
Line: 13, 1st char: 1, "}" is a "symbol".
```

Symbol table:

1. OwnTest4

2. a

3. b

4. c

5. d

6. e

7. f

8. g

9. h

```
1 // used to test '-' and '+'
2 public class OwnTest4 {
3     public static void main() {
4         int a = -1;
5         int b = 1-1;
6         int c = +123;
7         int d = -423+5;
8         int e = (+12-34)-56;
9         int f = e+65;
10        float g = -123.43e23+12.34;
11        float h = (+12.2-34.21f)-56.21e-465;
12    }
13 }
```