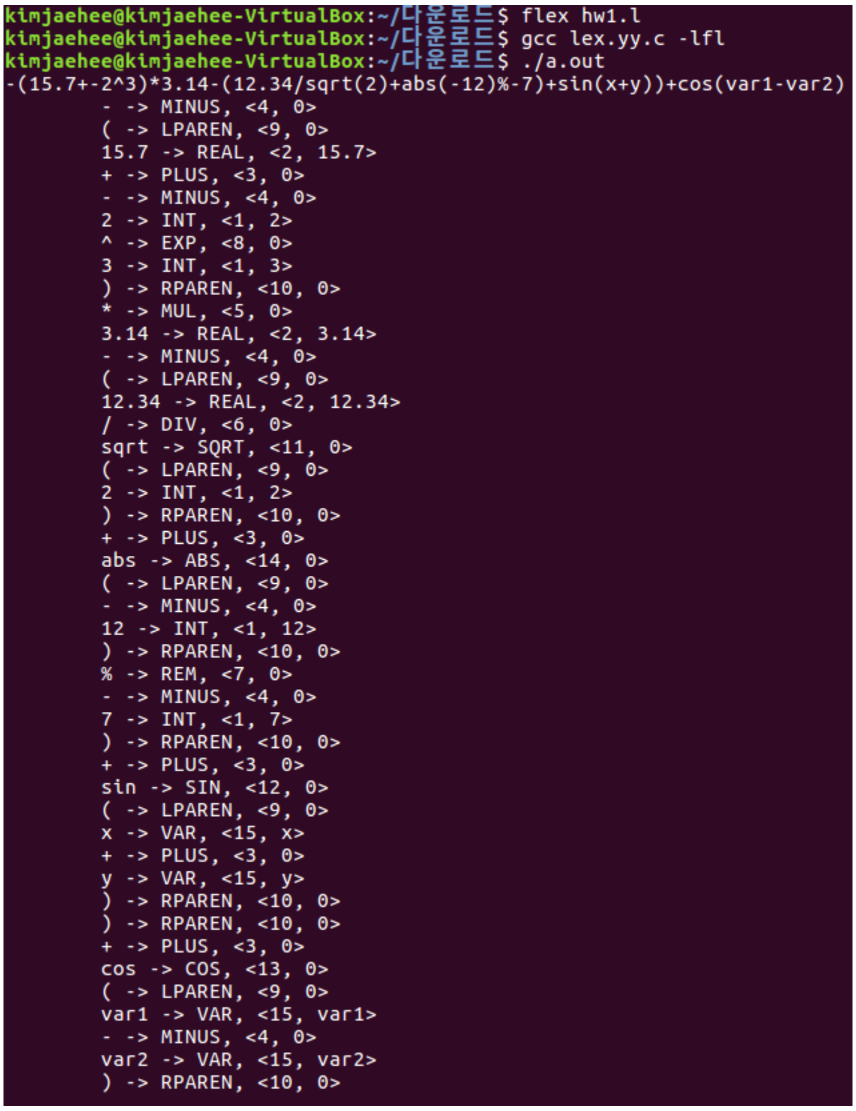
컴파일러 – Project3 (lex)

국민대학교 컴퓨터공학과

20143046 김재희

1. 아래 수식에 대해 각 토큰을 인식하는 Lex 입력 파일을 작성하시오.

−(15.7 + −2^3) * 3.14 − (12.34 / sqrt(2) + abs(−12) % −7) + sin(x+y)) +cos(var1−var2)



```
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ flex hw1.l
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ gcc lex.yy.c -lfl
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ ./a.out
-(15.7+-2^3)*3.14-(12.34/sqrt(2)+abs(-12)%-7)+sin(x+y))+cos(var1-var2)
        - -> MINUS, <4, 0>
        ( -> LPAREN, <9, 0>
        15.7 -> REAL, <2, 15.7>
        + -> PLUS, <3, 0>
        - -> MINUS, <4, 0>
        2 -> INT, <1, 2>
        ^ -> EXP, <8, 0>
        3 -> INT, <1, 3>
        ) -> RPAREN, <10, 0>
        * -> MUL, <5, 0>
        3.14 -> REAL, <2, 3.14>
        - -> MINUS, <4, 0>
        ( -> LPAREN, <9, 0>
        12.34 -> REAL, <2, 12.34>
        / -> DIV, <6, 0>
        sqrt -> SQRT, <11, 0>
        ( -> LPAREN, <9, 0>
        2 -> INT, <1, 2>
        ) -> RPAREN, <10, 0>
        + -> PLUS, <3, 0>
        abs -> ABS, <14, 0>
        ( -> LPAREN, <9, 0>
        - -> MINUS, <4, 0>
        12 -> INT, <1, 12>
        ) -> RPAREN, <10, 0>
        % -> REM, <7, 0>
        - -> MINUS, <4, 0>
        7 -> INT, <1, 7>
        ) -> RPAREN, <10, 0>
        + -> PLUS, <3, 0>
        sin -> SIN, <12, 0>
        ( -> LPAREN, <9, 0>
        x -> VAR, <15, x>
        + -> PLUS, <3, 0>
        y -> VAR, <15, y>
        ) -> RPAREN, <10, 0>
        ) -> RPAREN, <10, 0>
        + -> PLUS, <3, 0>
        cos -> COS, <13, 0>
        ( -> LPAREN, <9, 0>
        var1 -> VAR, <15, var1>
        - -> MINUS, <4, 0>
        var2 -> VAR, <15, var2>
        ) -> RPAREN, <10, 0>
```

정수와 실수와 명칭을 인식할 때를 제외하고는 token value가 0이 되도록 출력했다.

각각의 토큰들을 정의하여 출력하였다.

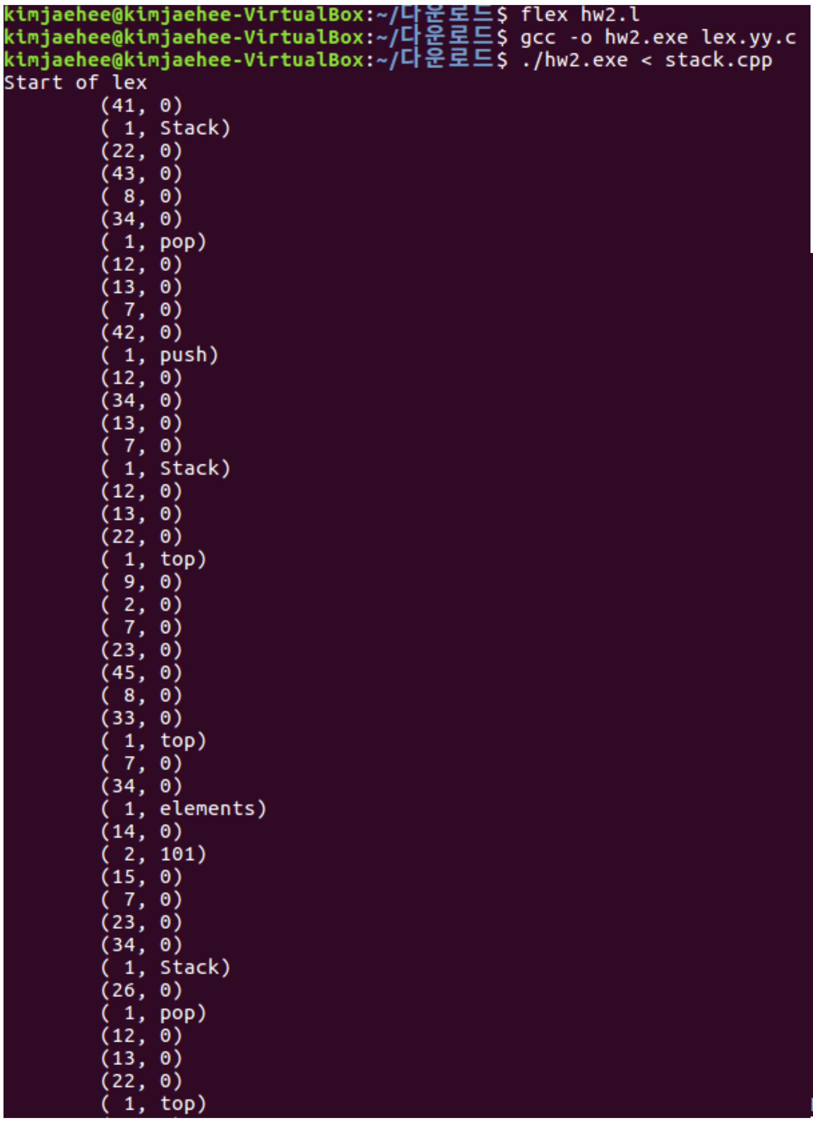
정수라면 [0-9]+ 하여 0에서 9까지의 숫자가 1번 이상 반복하면 인식하도록 하였다.

실수라면 [0-9]*₩.[0-9]+ 하여 0에서 9까지의 숫자가 0번 이상 반복한 뒤 .(dot)을 써주고 0에서 9까지의 숫자가 1번 이상 반복하면 인식하도록 하였다.

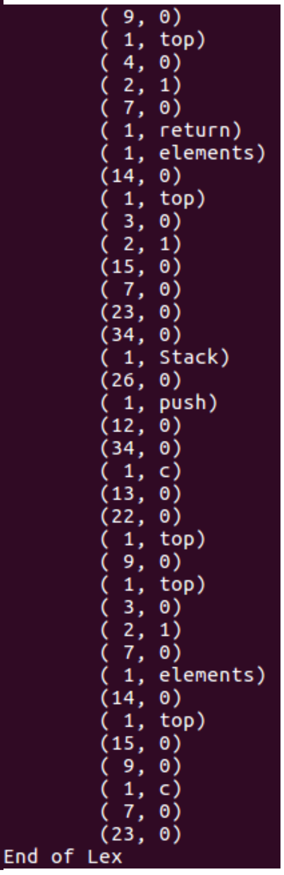
Enum을 사용하여 ERROR를 0으로 시작하여 1,2,3 차례대로 token number를 부여했다.

2. mini-python과 C++/Java의 class에 대한 어휘분석기를 작성하시오.

1) Stack.cpp 파일에 대하여

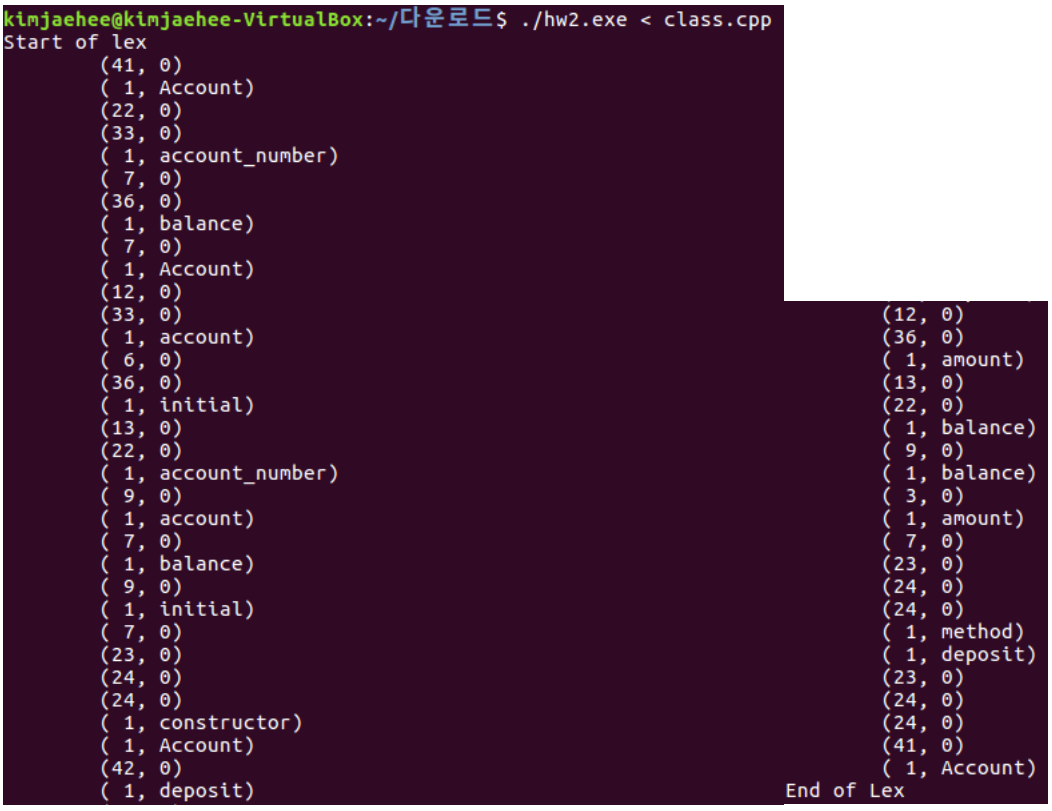


```
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ flex hw2.l
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ gcc -o hw2.exe lex.yy.c
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ ./hw2.exe < stack.cpp
Start of lex
        (41, 0)
        ( 1, Stack)
        (22, 0)
        (43, 0)
        ( 8, 0)
        (34, 0)
        ( 1, pop)
        (12, 0)
        (13, 0)
        ( 7, 0)
        (42, 0)
        ( 1, push)
        (12, 0)
        (34, 0)
        (13, 0)
        ( 7, 0)
        ( 1, Stack)
        (12, 0)
        (13, 0)
        (22, 0)
        ( 1, top)
        ( 9, 0)
        ( 2, 0)
        ( 7, 0)
        (23, 0)
        (45, 0)
        ( 8, 0)
        (33, 0)
        ( 1, top)
        ( 7, 0)
        (34, 0)
        ( 1, elements)
        (14, 0)
        ( 2, 101)
        (15, 0)
        ( 7, 0)
        (23, 0)
        (34, 0)
        ( 1, Stack)
        (26, 0)
        ( 1, pop)
        (12, 0)
        (13, 0)
        (22, 0)
        ( 1, top)
```



```
        ( 9, 0)
        ( 1, top)
        ( 4, 0)
        ( 2, 1)
        ( 7, 0)
        ( 1, return)
        ( 1, elements)
        (14, 0)
        ( 1, top)
        ( 3, 0)
        ( 2, 1)
        (15, 0)
        ( 7, 0)
        (23, 0)
        (34, 0)
        ( 1, Stack)
        (26, 0)
        ( 1, push)
        (12, 0)
        (34, 0)
        ( 1, c)
        (13, 0)
        (22, 0)
        ( 1, top)
        ( 9, 0)
        ( 1, top)
        ( 3, 0)
        ( 2, 1)
        ( 7, 0)
        ( 1, elements)
        (14, 0)
        ( 1, top)
        (15, 0)
        ( 9, 0)
        ( 1, c)
        ( 7, 0)
        (23, 0)
End of Lex
```

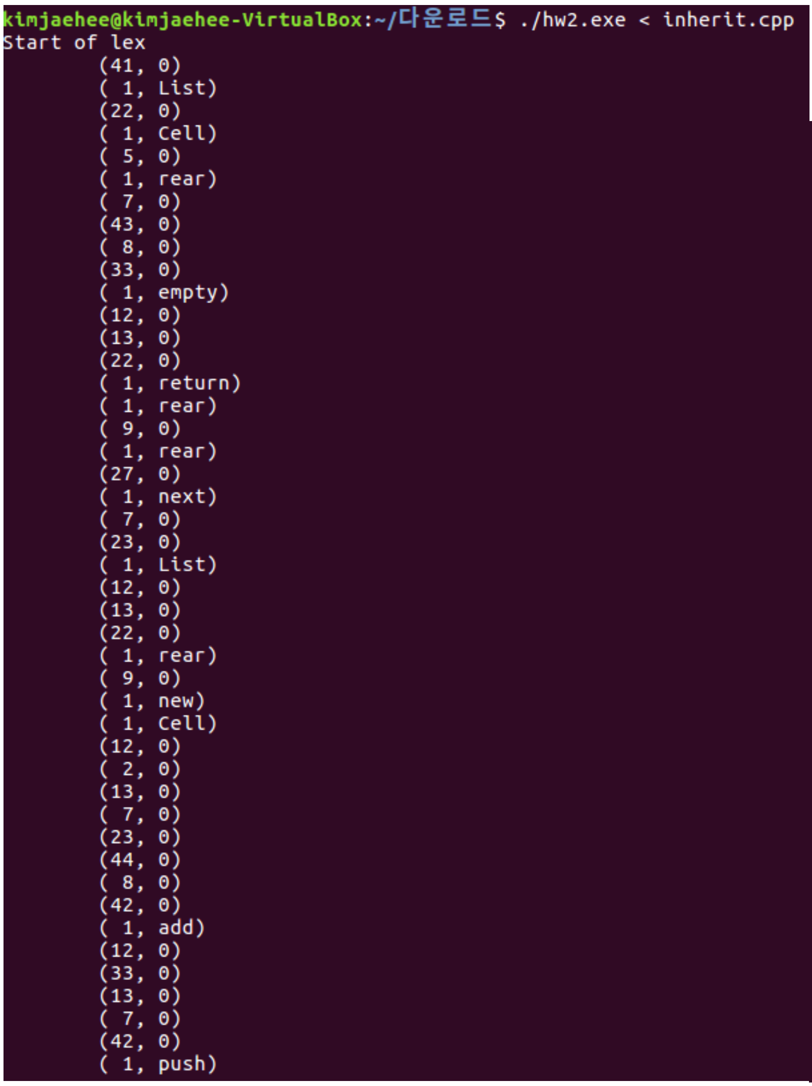
2) Class.cpp 파일에 대하여



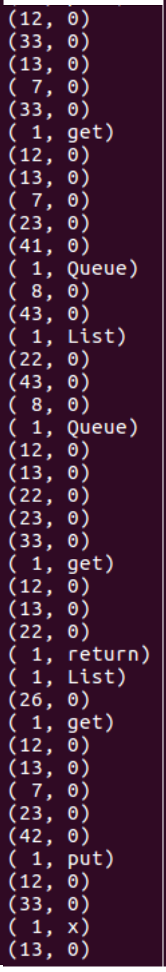
```
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ ./hw2.exe < class.cpp
Start of lex
        (41, 0)
        ( 1, Account)
        (22, 0)
        (33, 0)
        ( 1, account_number)
        ( 7, 0)
        (36, 0)
        ( 1, balance)
        ( 7, 0)
        ( 1, Account)
        (12, 0)
        (33, 0)
        ( 1, account)
        ( 6, 0)
        (36, 0)
        ( 1, initial)
        (13, 0)
        (22, 0)
        ( 1, account_number)
        ( 9, 0)
        ( 1, account)
        ( 7, 0)
        ( 1, balance)
        ( 9, 0)
        ( 1, initial)
        ( 7, 0)
        (23, 0)
        (24, 0)
        (24, 0)
        ( 1, constructor)
        ( 1, Account)
        (42, 0)
        ( 1, deposit)
```

```
        (12, 0)
        (36, 0)
        ( 1, amount)
        (13, 0)
        (22, 0)
        ( 1, balance)
        ( 9, 0)
        ( 1, balance)
        ( 3, 0)
        ( 1, amount)
        ( 7, 0)
        (23, 0)
        (24, 0)
        (24, 0)
        ( 1, method)
        ( 1, deposit)
        (23, 0)
        (24, 0)
        (24, 0)
        (41, 0)
        ( 1, Account)
End of Lex
```

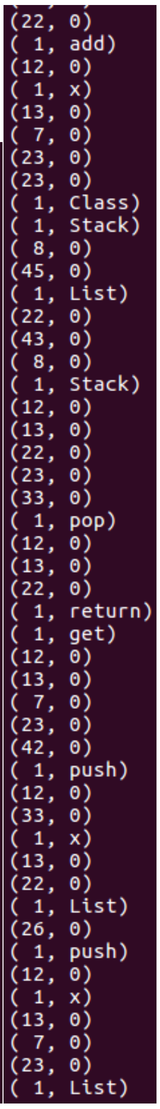
3) Inherit.cpp 파일에 대하여



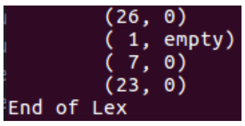
```
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ ./hw2.exe < inherit.cpp
Start of lex
        (41, 0)
        ( 1, List)
        (22, 0)
        ( 1, Cell)
        ( 5, 0)
        ( 1, rear)
        ( 7, 0)
        (43, 0)
        ( 8, 0)
        (33, 0)
        ( 1, empty)
        (12, 0)
        (13, 0)
        (22, 0)
        ( 1, return)
        ( 1, rear)
        ( 9, 0)
        ( 1, rear)
        (27, 0)
        ( 1, next)
        ( 7, 0)
        (23, 0)
        ( 1, List)
        (12, 0)
        (13, 0)
        (22, 0)
        ( 1, rear)
        ( 9, 0)
        ( 1, new)
        ( 1, Cell)
        (12, 0)
        ( 2, 0)
        (13, 0)
        ( 7, 0)
        (23, 0)
        (44, 0)
        ( 8, 0)
        (42, 0)
        ( 1, add)
        (12, 0)
        (33, 0)
        (13, 0)
        ( 7, 0)
        (42, 0)
        ( 1, push)
```



```
(12, 0)
(33, 0)
(13, 0)
( 7, 0)
(33, 0)
( 1, get)
(12, 0)
(13, 0)
( 7, 0)
(23, 0)
(41, 0)
( 1, Queue)
( 8, 0)
(43, 0)
( 1, List)
(22, 0)
(43, 0)
( 8, 0)
( 1, Queue)
(12, 0)
(13, 0)
(22, 0)
(23, 0)
(33, 0)
( 1, get)
(12, 0)
(13, 0)
(22, 0)
( 1, return)
( 1, List)
(26, 0)
( 1, get)
(12, 0)
(13, 0)
( 7, 0)
(23, 0)
(42, 0)
( 1, put)
(12, 0)
(33, 0)
( 1, x)
(13, 0)
```



```
(22, 0)
( 1, add)
(12, 0)
( 1, x)
(13, 0)
( 7, 0)
(23, 0)
(23, 0)
( 1, Class)
( 1, Stack)
( 8, 0)
(45, 0)
( 1, List)
(22, 0)
(43, 0)
( 8, 0)
( 1, Stack)
(12, 0)
(13, 0)
(22, 0)
(23, 0)
(33, 0)
( 1, pop)
(12, 0)
(13, 0)
(22, 0)
( 1, return)
( 1, get)
(12, 0)
(13, 0)
( 7, 0)
(23, 0)
(42, 0)
( 1, push)
(12, 0)
(33, 0)
( 1, x)
(13, 0)
(22, 0)
( 1, List)
(26, 0)
( 1, push)
(12, 0)
( 1, x)
(13, 0)
( 7, 0)
(23, 0)
( 1, List)
```



```
        (26, 0)
        ( 1, empty)
        ( 7, 0)
        (23, 0)
End of Lex
```

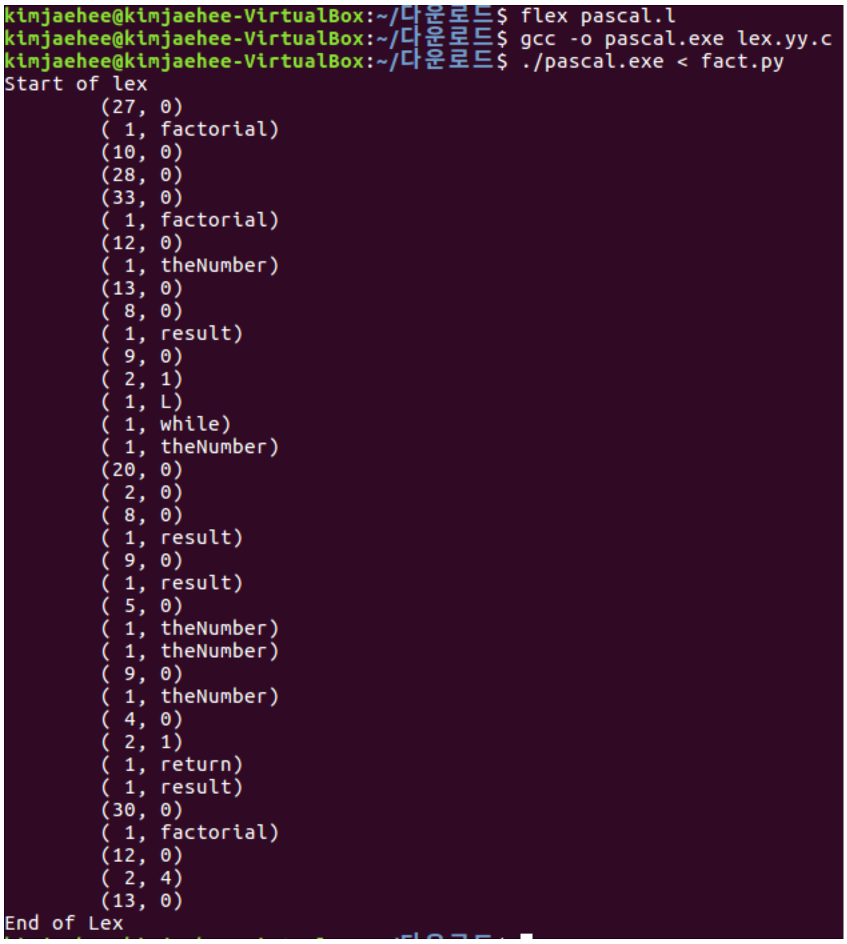
기존에 나와있던 mini-c.l 파일을 참고하여 c++ class 문법과 관련된 규칙을 추가해주었다.

우선 class 에서 사용되는 "::"를 TSEMICOLON이라는 이름을 붙여 추가하였다.

"->" TCREATE이라는 이름을 붙여 추가하였다.

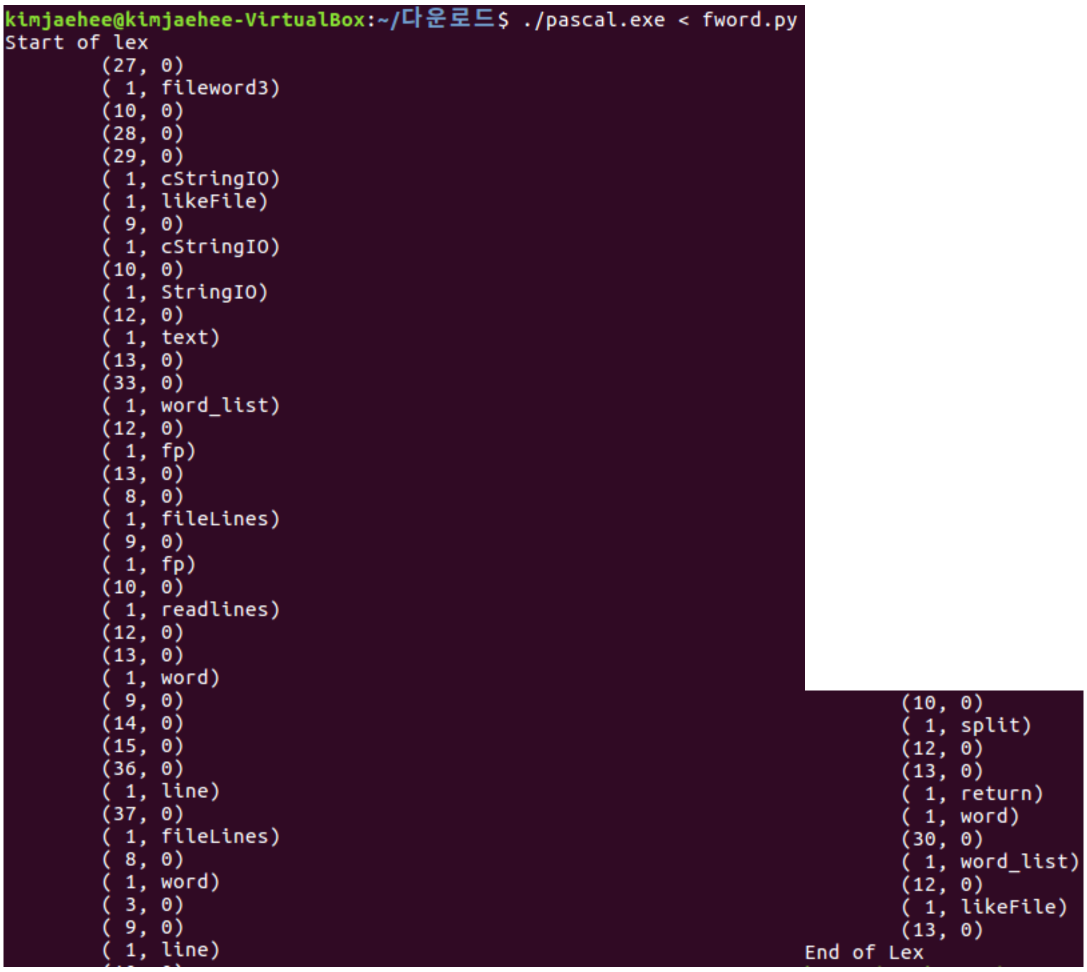
Class, public, protected, private도 추가하였다.

4) Fact.py파일에 대하여(python에 대하여 작성)



```
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ flex pascal.l
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ gcc -o pascal.exe lex.yy.c
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ ./pascal.exe < fact.py
Start of lex
        (27, 0)
        ( 1, factorial)
        (10, 0)
        (28, 0)
        (33, 0)
        ( 1, factorial)
        (12, 0)
        ( 1, theNumber)
        (13, 0)
        ( 8, 0)
        ( 1, result)
        ( 9, 0)
        ( 2, 1)
        ( 1, L)
        ( 1, while)
        ( 1, theNumber)
        (20, 0)
        ( 2, 0)
        ( 8, 0)
        ( 1, result)
        ( 9, 0)
        ( 1, result)
        ( 5, 0)
        ( 1, theNumber)
        ( 1, theNumber)
        ( 9, 0)
        ( 1, theNumber)
        ( 4, 0)
        ( 2, 1)
        ( 1, return)
        ( 1, result)
        (30, 0)
        ( 1, factorial)
        (12, 0)
        ( 2, 4)
        (13, 0)
End of Lex
```

5) Fword.py 파일에 대하여



```
kimjaehee@kimjaehee-VirtualBox:~/다운로드$ ./pascal.exe < fword.py
Start of lex
        (27, 0)
        ( 1, fileword3)
        (10, 0)
        (28, 0)
        (29, 0)
        ( 1, cStringIO)
        ( 1, likeFile)
        ( 9, 0)
        ( 1, cStringIO)
        (10, 0)
        ( 1, StringIO)
        (12, 0)
        ( 1, text)
        (13, 0)
        (33, 0)
        ( 1, word_list)
        (12, 0)
        ( 1, fp)
        (13, 0)
        ( 8, 0)
        ( 1, fileLines)
        ( 9, 0)
        ( 1, fp)
        (10, 0)
        ( 1, readlines)
        (12, 0)
        (13, 0)
        ( 1, word)
        ( 9, 0)
        (14, 0)
        (15, 0)
        (36, 0)
        ( 1, line)
        (37, 0)
        ( 1, fileLines)
        ( 8, 0)
        ( 1, word)
        ( 3, 0)
        ( 9, 0)
        ( 1, line)
```

```
        (10, 0)
        ( 1, split)
        (12, 0)
        (13, 0)
        ( 1, return)
        ( 1, word)
        (30, 0)
        ( 1, word_list)
        (12, 0)
        ( 1, likeFile)
        (13, 0)
End of Lex
```

pyton에서 변수형은 사용하지 않기 때문에 지워주었다.

대신 print, import, if, else, elif, try, except, for, in, def에 관한 문법을 추가해주었다.