

Programming Language Assignment #1

External Document

소프트웨어대학 소프트웨어학부

20204043 문벼리



#코드실행방법

1. 문서 안에 들어있는 codefile 안의 main.py 함수를 실행합니다.
2. 1 번 줄의 open 함수에서

```
code = open('../code1.txt')
```

경로를 바꾸어줍니다.

Code1 부분을 code2, code3, code4.....로 바꾸면 다른 텍스트파일이 실행이 됩니다.

#과제 예제 코드 1

code1.txt 파일에 있습니다.

코드:

operand1 := 3 ;

operand2 := operand1 + 2 ;

target := operand1 + operand2 * 3

컴파일 결과:

```
/usr/local/bin/python3.7 /Users/PC/Desktop/PLhomework/codeFile/main.py
operand2 := + 2 ;
{'ID': 2, 'CONST': 1, 'OP': 1}
(Error) 정의되지 않은 변수(operand1)가 참조됨
target := operand1 + operand2 * 3
{'ID': 3, 'CONST': 1, 'OP': 2}
(OK)
Result==> {'operand2': 'Unknown', 'operand1': 'Unknown', 'target': 'Unknown'}

Process finished with exit code 0
```

#과제 예제 코드 2

code2.txt 파일에 있습니다.

코드:

operand2 := operand1 + 2 ;

target := operand1 + operand2 * 3

컴파일 결과:

```
operand1 := 3 ;  
{'ID': 1, 'CONST': 1, 'OP': 0}  
(OK)  
operand2 := operand1 + 2 ;  
{'ID': 2, 'CONST': 1, 'OP': 1}  
(OK)  
target := operand1 + operand2 * 3  
{'ID': 3, 'CONST': 1, 'OP': 2}  
(OK)  
Result==> {'operand1': 3, 'operand2': 5, 'target': 18}
```

#과제 예제 코드 3

code3.txt 파일에 있습니다.

코드:

operand1 := 1;

operand2 := (operand1 * 3) + 2 ;

target := operand1 + operand2 * 3

컴파일 결과:

```
operand1 := 1 ;
{'ID': 1, 'CONST': 1, 'OP': 0}
(OK)
operand2 := ( operand1 * 3 ) + 2 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(OK)
target := operand1 + operand2 * 3
{'ID': 3, 'CONST': 1, 'OP': 2}
(OK)
Result==> {'operand1': 1, 'operand2': 5, 'target': 16}
```

#과제 예제 코드 4

code4.txt 파일에 있습니다.

코드:

operand1 := 3 ;

operand2 := operand1 + + 2 ;

target := operand1 + operand2 * 3

컴파일 결과:

```
operand1 := 3 ;
{'ID': 1, 'CONST': 1, 'OP': 0}
(OK)
operand2 := operand1 + 2 ;
{'ID': 2, 'CONST': 1, 'OP': 1}
(Warning) 중복 연산자 제거: +
target := operand1 + operand2 * 3
{'ID': 3, 'CONST': 1, 'OP': 2}
(OK)
Result==> {'operand1': 3, 'operand2': 5, 'target': 18}
```

처리한 에러: 중복된 연산자를 제거하고 파싱을 완료했다.

#코드 5

*추가적으로 작성한 에러 예외처리를 설명하고자 code5 파일을 추가했습니다.

code5.txt 파일에 있습니다.

코드:

operand1 := 5

operand2 := (operand1 + 3 *4 ;

operand3 := (operand1 + 3 (* 3 ;

operand4 := operand1 * (operand2 + 1 ;

operand5 := () operand1 + 2) * 3 ; ;

operand6 := (operand1 + 2) * + 3 ;

operand7 := (operand1 + 2) * * 3 ;

operand8 := (operand1 + 2) * 3

operand9 := (operand1 + 2) * 3 ; ;

operand10 = (operand1 + 2) * 3 ;

operand11 : (operand1 + 2) * 3 ;

operand12 w:= operand1 + 2 ;

operand13 operand1 + 2 * 3

컴파일 결과:

```
operand1 := 5 ;
{'ID': 1, 'CONST': 1, 'OP': 0}
(Warning) statement와 statement 사이에는 semi_colon이 필요합니다.
operand2 := ( operand1 + 3 * 4 ) ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) 닫는 괄호 추가
operand3 := ( operand1 + 3 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) 괄호 방향 변경
operand4 := operand1 * ( operand2 + 1 ) ;
{'ID': 3, 'CONST': 1, 'OP': 2}
(Warning) 닫는 괄호 추가
operand5 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) 괄호 제거: )
(Warning) 중복 semicolon 제거
operand6 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) 중복 연산자 제거: +
operand7 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) 중복 연산자 제거: *
operand8 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) statement와 statement 사이에는 semi_colon이 필요합니다.
```

```

operand9 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) 중복 semicolon 제거
operand10 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) =을 :=으로 변경
operand11 := ( operand1 + 2 ) * 3 ;
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) :을 :=으로 변경
operand12 := operand1 + 2 ;
{'ID': 2, 'CONST': 1, 'OP': 1}
(Warning) 인식할 수 없는 토큰 삭제: w
operand13 := operand1 + 2 * 3
{'ID': 2, 'CONST': 2, 'OP': 2}
(Warning) assign_op가 필요합니다.

```

Result==> {'operand1': 5, 'operand2': 17, 'operand3': 24, 'operand4': 90, 'operand5': 21, 'operand6': 21, 'operand7': 21, 'operand8': 21, 'operand9': 21, 'operand10': 21, 'operand11': 21, 'operand12': 7, 'operand13': 11}

처리한 에러:

Warning 뒤의 문장에 설명되어있습니다.