External Document

이름

이름	학번
20225779	김제신

Build And Test Environment

- OS : Windows 11
- JDK : Amazon Corretto 17
- Build System : Gradle 7.2

How To Run

```
>> java -jar .\Main.jar .\testcase\example1.txt
```

OR

```
>> java -jar .\Main.jar  -v .\testcase\example1.txt
```

How To Build

```
>> ./gradlew build
```

Exception Handling

Error

- 선언되지 않은 변수 사용

  - 예제 문장

    ```
    operand2 := operand1 + 2 ;
    target := operand1 + operand2 * 3
    ```

  - 실행 결과

    ```
    PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\example2.txt
    Statement : operand2 := operand1 + 2;
    ID : 2 CONST : 1 OP : 1
    (Error) - Undefined identifier (operand1)
    Statement : target := operand1 + operand2 * 3
    ID : 3 CONST : 1 OP : 2
    (Error) - Undefined identifier (operand1)
    (Error) - Undefined identifier (operand2)
    => Result : operand1 = Unknown operand2 = Unknown target = Unknown
    ```

- Statement가 변수로 시작되지 않는 경우

  - 예제 문장

    ```
    vim := 1000;
    emacs := 2000;
    + := vs;
    vscode := 500
    ```

  - 실행 결과

    ```
    PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\start_without_ident2.txt
    Statement : vim := 1000;
    ID : 1 CONST : 1 OP : 0
    (OK)
    Statement : emacs := 2000;
    ID : 1 CONST : 1 OP : 0
    (OK)
    Statement : + := vs;
    ID : 1 CONST : 0 OP : 1
    (Error) - Unexpected identifier
    (Error) - Undefined identifier (vs)
    Statement : vscode := 500
    ID : 1 CONST : 1 OP : 0
    (OK)
    => Result : emacs = 2000 vim = 1000 vs = Unknown vscode = 500
    ```

- $, % 같은 정의되지 않은 기호들이 나올 경우
  - 예제 문장

```
c := 1 + 1;
py := 2 % 2;
rs := 3 * 3;
rb := 4 $ 4
```

  - 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\unknown_op.txt
Statement : c := 1 + 1;
ID : 1 CONST : 2 OP : 1
(OK)
Statement : py := 2 % 2;
ID : 1 CONST : 2 OP : 0
(Error) - Unexpected operator
Statement : rs := 3 * 3;
ID : 1 CONST : 2 OP : 1
(OK)
Statement : rb := 4 $ 4
ID : 1 CONST : 2 OP : 0
(Error) - Unexpected operator
=> Result : c = 2 py = Unknown rb = Unknown rs = 9
```

- Operator가 없을 경우
  - 예제 문장

```
operand1 := 37;
operand2 := 5 operand1;
operand3 := operand2
```

  - 실행 결과

```
Statement : operand1 := 37;
ID : 1 CONST : 1 OP : 0
(OK)
Statement : operand2 := 5 operand1;
ID : 2 CONST : 1 OP : 0
(Error) - Unexpected token
(Error) - Unexpected token
Statement : operand3 := operand2
ID : 2 CONST : 0 OP : 0
(Error) - Undefined identifier (operand2)
=> Result : operand1 = 37 operand2 = Unknown operand3 = Unknown
```

- Operand가 없을 경우

- 예제 문장

```
seoul := 4 + 2;
wien := 4 + ;
berlin := seoul + wien
```

- 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\no_operand.txt
Statement : seoul := 4 + 2;
ID : 1 CONST : 2 OP : 1
(OK)
Statement : wien := 4 +;
ID : 1 CONST : 1 OP : 1
(Error) - Unexpected token
Statement : berlin := seoul + wien
ID : 3 CONST : 0 OP : 1
(Error) - Undefined identifier (wien)
=> Result : berlin = Unknown seoul = 6 wien = Unknown
```

- 중간에 세미콜론이 없는 경우
  - 예제 문장

```
num := 42;
doublenum := num * 2
quadnum := doublenum + doublenum
```

  - 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\no_semicolon1.txt
Statement : num := 42;
ID : 1 CONST : 1 OP : 0
(OK)
Statement : doublenum := num * 2 quadnum := doublenum + doublenum
ID : 5 CONST : 1 OP : 2
(Error) - Unexpected token
(Error) - Unexpected token
(Error) - Assignment Operator must be one
(Error) - Undefined identifier (doublenum)
(Error) - Undefined identifier (doublenum)
=> Result : doublenum = Unknown num = 42 quadnum = Unknown
```

- 파일이 공백일 경우
  - 예제 문장

  - 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\no_statement.txt
Statement :
ID : 0 CONST : 0 OP : 0
(Error) - There is no input.
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1>
```

Warning

- 연산자가 두 개 연속으로 나올 경우에 뒤의 연산자는 전부 무시

    - 예제 문장

    ```
    a := 1 + - * / / + - 4;
    b := a * / / - - a
    ```

    - 실행 결과

    ```
    PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\operators.txt
    Statement : a := 1 + 4;
    ID : 1 CONST : 2 OP : 1
    (Warning) - Consecutive operators were found - Remove Backward operator +
    (Warning) - Consecutive operators were found - Remove Backward Operator *
    (Warning) - Consecutive operators were found - Remove Backward Operator /
    (Warning) - Consecutive operators were found - Remove Backward Operator /
    (Warning) - Duplicate operators were found - Remove duplicate operator (+)
    (Warning) - Consecutive operators were found - Remove Backward operator +
    Statement : b := a * a
    ID : 3 CONST : 0 OP : 1
    (Warning) - Consecutive operators were found - Remove Backward operator *
    (Warning) - Consecutive operators were found - Remove Backward operator *
    (Warning) - Consecutive operators were found - Remove Backward operator-
    (Warning) - Consecutive operators were found - Remove Backward operator-
    => Result : a = 5 b = 25
    ```

- 맨 마지막 세미콜론이 있을 경우 삭제

    - 예제 문장

    ```
    open := 123;
    lab := 456;
    ```

    - 실행 결과

    ```
    PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\semicolon_at_the_end.tx
    Statement : open := 123;
    ID : 1 CONST : 1 OP : 0
    (OK)
    Statement : lab := 456
    ID : 1 CONST : 1 OP : 0
    (Warning) - Semicolon in the last statement is not required. - Delete semicolon
    => Result : lab = 456 open = 123
    ```

- Assignment Operator가 두 개 이상 나왔을 경우 하나만 빼고 무시.

    - 예제 문장

```
 a := := 3;
```

- 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\two_assignment_op.txt
Statement : a := 3
ID : 1 CONST : 1 OP : 0
(Warning) - Consecutive Assignment Operator - Remove Duplicate Assignment Operator
=> Result : a = 3
```

- Assignment Operator가 완성되지 않았을 때 (ex) `;` , `=`), 이것들을 `:=` 로 바꿔서 처리.

  - 예제 문장

```
 a : 2;
 b = 3;
```

  - 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\not_completed_assignment_op.txt
Statement : a := 2;
ID : 1 CONST : 1 OP : 0
(Warning) - Change : to :=
Statement : b := 3
ID : 1 CONST : 1 OP : 0
(Warning) - Semicolon in the last statement is not required. - Delete semicolon
(Warning) - Change = to :=
=> Result : a = 2 b = 3
```

- 괄호가 닫히지 않았을 때, 나머지 괄호들을 마지막에 넣음으로써 처리.

  - 예제 문장

```
 a := (1 + (2 + 3)
```

  - 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\not_closed_parenthesis.txt
Statement : a := (1 + (2 + 3))
ID : 1 CONST : 3 OP : 2
(Warning) - not found right parenthesis - add right parenthesis in the end
=> Result : a = 6
```


정상 실행 결과

- 예제 문장

```
operand1 := 3 ;
operand2 := operand1 + 2 ;
target := operand1 + operand2 * 3
```

- 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar .\testcase\example1.txt
Statement : operand1 := 3;
ID : 1 CONST : 1 OP : 0
(OK)
Statement : operand2 := operand1 + 2;
ID : 2 CONST : 1 OP : 1
(OK)
Statement : target := operand1 + operand2 * 3
ID : 3 CONST : 1 OP : 2
(OK)
=> Result : operand1 = 3 operand2 = 5 target = 18
```

- `-v` 옵션을 적용 시킨 실행 결과

```
PS C:\Users\Jeshin Kim\IntelliJProject\PL_Project1> java -jar .\Main.jar -v .\testcase\example1.txt
IDENT
ASSIGN_OP
INT_LIT
SEMICOLON
EOF
IDENT
ASSIGN_OP
IDENT
ADD_OP
INT_LIT
SEMICOLON
EOF
IDENT
ASSIGN_OP
IDENT
ADD_OP
IDENT
MULT_OP
INT_LIT
EOF
```

- `-v` 옵션을 적용시켰을 때에는 Token의 변화과정을 출력한다.