

Лабораторная работа №5

```
from scanner import load_file, read_lex, next_lex, get_col, get_row
```

```
DEBUG = True
```

```
def is_type(name: str):  
    return name in ['SHORT', 'INT', 'LONG']
```

```
def program():  
    if DEBUG:  
        print('program')  
    if read_lex().name == 'EOF':  
        return  
    if read_lex().name == 'VOID':  
        function()  
    elif is_type(read_lex().name):  
        variable()  
    else:  
        err('Ожидался тип (short, int, long) или void')
```

```
def function():  
    if DEBUG:  
        print('function')  
    if next_lex().name != 'VOID':  
        err('Ожидался void')  
    if next_lex().name != 'ID':  
        err(f'Ожидался идентификатор')  
    if next_lex().name != 'ROUND_LEFT':  
        err(f'Ожидался (')  
    if next_lex().name != 'ROUND_RIGHT':  
        err(f'Ожидался ')  
    composite_operator()
```

```
def composite_operator():  
    if DEBUG:  
        print('composite_operator')  
    if next_lex().name != 'CURLY_LEFT':  
        err('Ожидался {')  
    while read_lex().name != 'EOF' and read_lex().name != 'CURLY_RIGHT':  
        if read_lex().name == 'ID':  
            call_function()
```

```
    if is_type(read_lex().name):
        variable()
    if read_lex().name == 'IF':
        call_if()
    if next_lex().name != 'CURLY_RIGHT':
        err('Ожидался }')
```

```
def call_function():
    if DEBUG:
        print('call_function')
    if next_lex().name != 'ID':
        err('Ожидался идентификатор')
    if next_lex().name != 'ROUND_LEFT':
        err('Ожидался (')
    expression()
    while read_lex().name == 'COMMA':
        next_lex()
        expression()
    if next_lex().name != 'ROUND_RIGHT':
        err('Ожидался ')
    if next_lex().name != 'SEMICOLON':
        err('Ожидался ;')
```

```
def expression():
    if DEBUG:
        print('expression')
    expression_1()
```

```
def expression_1():
    if DEBUG:
        print('expression_1')
    expression_2()
    while read_lex().name == 'EQ' or read_lex().name == 'NOT_EQ':
        next_lex()
        expression_2()
```

```
def expression_2():
    if DEBUG:
        print('expression_2')
    expression_3()
    while read_lex().name == 'LESS' or read_lex().name == 'GREATER' or read_lex().name
    == 'LESS_EQ' or read_lex().name == 'GREATER_EQ':
        next_lex()
        expression_3()
```

```

def expression_3():
    if DEBUG:
        print('expression_3')
    expression_4()
    while read_lex().name == 'R_SHIFT' or read_lex().name == 'L_SHIFT':
        next_lex()
        expression_4()

def expression_4():
    if DEBUG:
        print('expression_4')
    expression_5()
    while read_lex().name == 'PLUS' or read_lex().name == 'MINUS':
        next_lex()
        expression_5()

def expression_5():
    if DEBUG:
        print('expression_5')
    expression_6()
    while read_lex().name == 'STAR' or read_lex().name == 'SLASH' or read_lex().name ==
'PERCENT':
        next_lex()
        expression_6()

def expression_6():
    if DEBUG:
        print('expression_6')
    while read_lex().name == 'MINUS' or read_lex().name == 'PLUS':
        next_lex()
        expression_7()

def expression_7():
    if DEBUG:
        print('expression_7')
    if read_lex().name == 'ROUND_LEFT':
        next_lex()
        expression()
        if next_lex().name != 'ROUND_RIGHT':
            err('Ожидался ')
    else:
        next_lex()

```

```

def variable():
    if DEBUG:
        print('variable')
    if not is_type(next_lex().name):
        err('Ожидался тип (short, int, long)')
    f = True
    while f:
        if next_lex().name != 'ID':
            err('Ожидался идентификатор')
        if next_lex().name != 'ASSIGN':
            err('Ожидался =')
        expression()
        f = False
        if read_lex().name == 'COMMA':
            f = True
            next_lex()
    if next_lex().name != 'SEMICOLON':
        err('Ожидался ;')

```

```

def call_if():
    if DEBUG:
        print('call_if')
    if next_lex().name != 'IF':
        err('Ожидался if')
    if next_lex().name != 'ROUND_LEFT':
        err('Ожидался (')
    expression()
    if next_lex().name != 'ROUND_RIGHT':
        err('Ожидался )')
    composite_operator()
    if read_lex().name == 'ELSE':
        next_lex()
        composite_operator()

```

```

def err(text: str):
    print(text, 'Найден:', read_lex(), 'Строка:', get_row(), 'Символ:', get_col())
    exit(1)

```