

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики

Лабораторна робота №5
З дисципліни “Системне програмування”

Виконав:
студент 3-го курсу групи ТТП-32
спеціальності "Інформатика"
Панасюк Богдан Ігорович

Київ – 2023

За умовою потрібно розробити синтаксичний аналізатор для арифметичних виразів.

Виконання програми

Програма зчитує ввід користувача і виконує прості арифметичні дії.

Реалізація

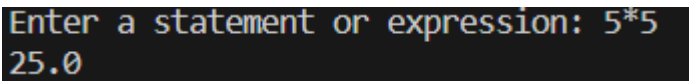
файл lex.py зберігає наші регулярні правила які розпізнають наші операції

```
tokens = (  
    'NUMBER',  
    'PLUS',  
    'MINUS',  
    'TIMES',  
    'DIVIDE',  
    'LPAREN',  
    'RPAREN',  
    'SQRT',  
    'EXPONENT',  
    'IDENTIFIER',  
    'EQUALS',  
)  
  
# Regular expression rules for simple tokens  
t_PLUS = r'\+'  
t_MINUS = r'\-'  
t_TIMES = r'\*'  
t_DIVIDE = r'\/'  
t_LPAREN = r'\('  
t_RPAREN = r'\)'  
t_EXPONENT = r'\^'  
t_EQUALS = r'='  
t_SQRT = r'sqrt'
```

На прикладі операції мінус, плюс, множення і ділення, продемонструю виконання програми

```
# Grammar rule for handling binary operations in arithmetic expressions.
def p_expression_binop(p):
    '''expression : expression PLUS expression
    | expression MINUS expression
    | expression TIMES expression
    | expression DIVIDE expression'''
    if p[2] == '+':
        p[0] = p[1] + p[3]
    elif p[2] == '-':
        p[0] = p[1] - p[3]
    elif p[2] == '*':
        p[0] = p[1] * p[3]
    elif p[2] == '/':
        if p[3] == 0:
            print("Error: dividing by zero")
        else:
            p[0] = p[1] / p[3]
```

На вхід приймаємо якісь символи які були розпізанні нашим аналізатор і після цього скануємо на наявність певних символів, які дають змогу зрозуміти як операція повинна виконуватись.

На прикладі множення: 

Спочатку передаємо input в main

```
while True:
    try:
        userInput = input('Enter a statement or expression: ')
```

Після чого input передається в лексер, який розбиває його на лексеми і передає їх в парсер, який виконує синтаксичний аналіз. Після цього ми отримуємо вивід.