Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине: «Проектирование лексического анализатора с использованием регулярных выражений»

# Тема: «Проектирование лексического анализатора с использованием регулярных выражений»

Выполнил:

Студент 4 курса

Группы ИИ-21

Литвинюк Т. В.

Проверил:

Монтик Н.С.

Брест 2024

**Цель работы:** изучение основных понятий теории регулярных грамматик, ознакомление с назначением и принципами работы лексических анализаторов (сканеров), получение практических навыков построения сканера на примере заданного простейшего входного языка.

11. Входной язык содержит последовательность описаний записей (record) в соответствии со спецификацией языка Паскаль, разделенных символом ;(точка с запятой). Считать, что записи могут содержать только поля скалярных типов integer, real, byte, word, char и строки string с возможным указанием длины строки в квадратных скобках.

import re

# Регулярные выражения для токенов

TOKEN\_SPECIFICATION = [

('COMMENT', r'{[^}]\*}'),

('STRING', r'"[^"]{0,32}"'),

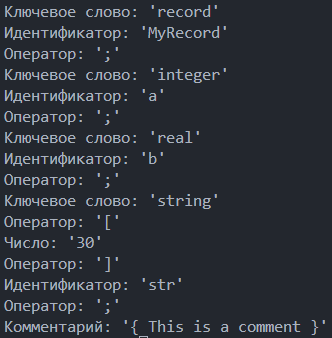
('NUMBER', r'\d+(\.\d+)?'),

('KEYWORD', r'\b(record|integer|real|byte|word|char|string)\b'),

('IDENTIFIER', r'[A-Za-z\_][A-Za-z0-9\_]{0,31}'),

('OPERATOR', r'[=;:,.\[\]]'),

('WHITESPACE', r'\s+'),

**** ('MISMATCH', r'.'),

]

TOKENS\_REGEX = '|'.join(

f'(?P<{pair[0]}>{pair[1]})' for pair in TOKEN\_SPECIFICATION)

TOKEN\_TYPES = {

'KEYWORD': 'Ключевое слово',

'IDENTIFIER': 'Идентификатор',

'NUMBER': 'Число',

'STRING': 'Строка',

'OPERATOR': 'Оператор',

'COMMENT': 'Комментарий',

'MISMATCH': 'Ошибка'

}

class Lexer:

def \_\_init\_\_(self, input\_text):

self.input\_text = input\_text

def tokenize(self):

tokens = []

for match in re.finditer(TOKENS\_REGEX, self.input\_text):

token\_type = match.lastgroup

token\_value = match.group(str(token\_type))

if token\_type == 'WHITESPACE':

continue

elif token\_type == 'MISMATCH':

tokens.append((token\_type, token\_value, "Некорректный символ"))

else:

tokens.append((token\_type, token\_value))

return tokens

if \_\_name\_\_ == "\_\_main\_\_":

input\_text = """

record MyRecord;

integer a;

real b;

string[30] str;

{ This is a comment }

"""

lexer = Lexer(input\_text)

tokens = lexer.tokenize()

for token\_type, token\_value, \*error in tokens:

if error:

print(f"{TOKEN\_TYPES[token\_type]}: '{token\_value}' -> {error[0]}")

else:

print(f"{TOKEN\_TYPES[token\_type]}: '{token\_value}'")

**Вывод:** в ходе выполнения лабораторной работы изучил основные понятия теории регулярных грамматик, ознакомился с назначением и принципами работы лексических анализаторов (сканеров), получил практические навыки построения сканера на примере заданного простейшего входного языка.