высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Энгельсский технологический институт

Кафедра Естественные и математические науки

Специальность 09.03.04 Программная инженерия

Отчёт по Лабораторной работе №2по дисциплине “Теория языков программирования и методы трансляции”

|  |  |
| --- | --- |
|  | Выполнил: студент 3 курса  учебной группы ПИНЖ-31  очной формы обучения  Сироткин Д.Ю. |

Энгельс 2023

**2.Вариант № 17**

**Условие:**

1.Получить вариант задания у преподавателя.

2. В соответствии с выданным вариантом выполнить следующее:

2.1. Составит техническое задание (ТЗ) на разработку программы сканера, производящей лексический анализ произвольных текстов в пределах установленного алфавита.

2.2. Согласовать ТЗ с преподавателем.

2.3. Разработать программу-сканер на языках Паскаль, С++ или в интегрированных средах по собственному усмотрению.

2.4. Провести тестирование и отладку программы (предусмотреть все случаи вывода сообщений об ошибках пользователю).

2.5. Составить отчёт по работе и приложить к нему ТЗ.

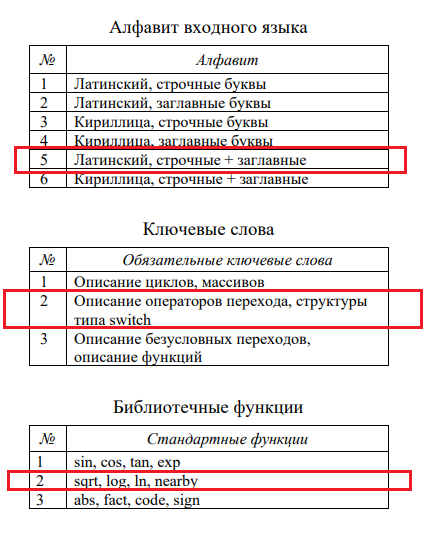
****

Рисунок 1.Изначальные данные для анализатора

**Выбранные ключевые слова:**

['BEGIN', 'PROGRAM', 'CASE', 'THEN', 'BREAK', 'VAR', 'IF', 'ELSE', 'END', 'SWITCH']

**Выбранные стандартные функции:**

sqrt, log, ln, nearby

**3.Внутренние таблицы сканера:**

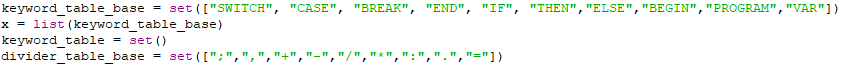
****

Рисунок 2.Таблица ключевых слов и разделителей

**4.Техническое задание**

**Основания для разработки**

Наименование документа: 2\_5280506778256160813

Организация утвердивший этот документ:СГТУ ЭТИ

Наименование разработки: Программы (сканера)

**Назначение разработки.**

**Требования к программе или программному изделию.**

**Требования к функциональным характеристикам:**

Программа должна производить лексический анализ входного текста для транслятора:

* создание таблиц выделенных лексем для конечных классов;
* кодирование идентификаторов, разделителей и констант;
* проверка правильности написания ключевых слов, операторов, стандартных функций и использование служебных символов;
* отображение дескрипторного текста.

**Требования к надежности:**

Программа должна стабильно обрабатывать текстовую информацию и производить лексический анализ.

**Условия эксплуатации:**

При использование программы пользователь должен облать минимальными навыками работы с языком программирования Python.

**Требования к информационно и программной совместимости:**

Программа должна быть запущенна на устройстве соответствующем минимальным требованиям языка Python:

* 64- или 32-разрядный процессор на архитектуре Intel или AMD;
* ГБ оперативной памяти, желательно 4 ГБ;
* 5 ГБ свободного места в памяти компьютера;
* Windows ,macOS ,Linux

Требования к программной документации

**Требования к программной документации.**

В комплекте с программой должен присутствовать документ со следующим содержимым:

* Код программы
* Вводимые текстовые данные
* Вывод программы

Стадии и этапы разработки

**Стадии и их этапы:**

Разработка технического задания:

1. Составление требования
2. Прописывание функций

Разработка программы:

1. Создание таблиц Ключевые слова и Разделители
2. Написание проверки на Идентификаторы и Константы
3. Вывод полученных таблиц (Ключевые слова, Разделители, Идентификаторы, Константы).
4. Написание алгоритма преобразование текста в дескрипторный текст
5. Вывод дескрипторного текста

Написание документа “Отчёт”:

1. Прописывание условий задачи
2. Вывод листинга программы
3. Вывод отладочного примера работа программы

**Сроки разработки:**

Дата начала разработки: 30.10.23

Дата завершения разработки:31.10.23

|  |  |
| --- | --- |
| Стадии: | Затраченное время |
| Разработка технического задания | 2 часа |
| Разработка программы | 2 часа |
| Написание документа “Отчёт”: | 1 час |

**Исполнители:**

Сироткин Денис Юрьевич

**Порядок контроля и приёмки**

**5.Листинг программы**

**Листинг программы:**

import re

code = """

VAR

a,b: Double;

N: Integer;

BEGIN

CASE N of

1: BEGIN

a:= sqrt(4);b:=1;

end;

2: BEGIN

a:=log(1);b:=1;

END;

ELSE WriteLn('ERROR N!');

END;

"""

print(code)

keyword\_table\_base = set(["SWITCH", "CASE", "BREAK", "END", "IF", "THEN","ELSE","BEGIN","PROGRAM","VAR"])

x = list(keyword\_table\_base)

keyword\_table = set()

divider\_table\_base = set([";",",","+","-","/","\*",":",".","="])

y = list(divider\_table\_base)

divider\_table = set()

identifier\_table = set()

const\_table = set()

tokens = re.findall(r'[a-zA-Za][a-zA-Zа\_d]\*|[;,:=]|[\d+\.\d+|\d+]', code)

consts = re.findall(r'\d+\.\d+|\d+', code)

print("Ключевые слова:")

print(x)

print("Разделители:")

print(y)

for token in tokens:

if token in keyword\_table\_base:

keyword\_table.add(token)

if token in divider\_table\_base:

divider\_table.add(token)

elif re.match(r'[a-zA-Za][a-zA-Zа\_d]\*$', token):

if not (token in keyword\_table):

identifier\_table.add(token)

print("Полученные Ключевые слова(K1):\n ", keyword\_table)

print("Полученнные Разделители(K2):\n ", divider\_table)

print("Полученные Идентификаторы(K3):\n ", identifier\_table)

z = list(identifier\_table)

g = set(consts);

print("Полученные Константы(K4):\n ", g)

g = list(consts)

g = set(consts);

g = list(consts)

print("Дескрипторный текст:")

for token in tokens:

if token in keyword\_table\_base:

print("(К1,", x.index(token), end = ')')

if token in divider\_table\_base:

print("(К2,", y.index(token),end = ')')

elif re.match(r'[a-zA-Za][a-zA-Zа\_d]\*$', token):

if not (token in keyword\_table):

print("(К3,", z.index(token),end = ')' )

elif re.match(r'[\d+\.\d+|\d+]\*$', token):

print("(К4,", g.index(token),end = ')' )

**6.Отладочный пример работы программы**

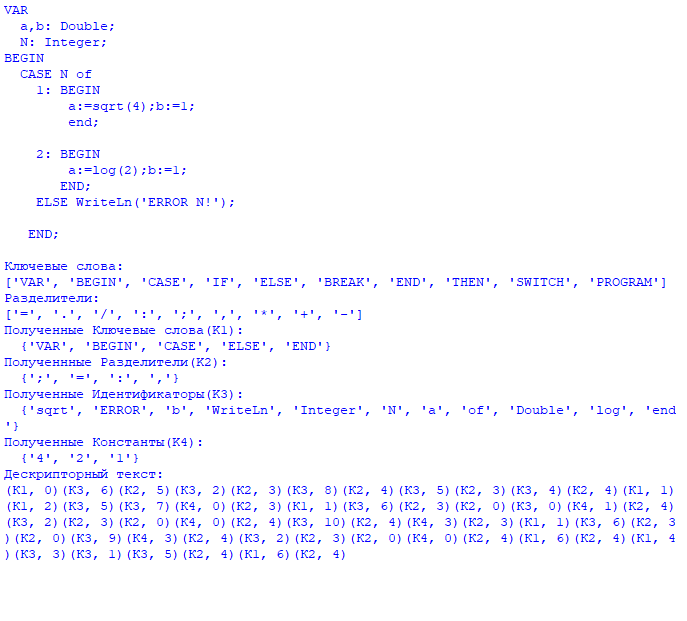
****

Рисунок 3.Пример работы программы