|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт Информационных Технологий | | |
| Кафедра Вычислительной техники | | |

|  |  |
| --- | --- |
| **ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ №3**  «Лексический анализатор на базе конечного автомата входного языка» | |
|  | |
| **по дисциплине** | |
| **«**Теория формальных языков**»** | |
|  | |
| Выполнил студент группы ИКБО-04-22 | *Кликушин В.И.* |
| Принял старший преподаватель | *Боронников А.С.* |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_04\_»\_\_\_09\_\_\_\_2023 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. |  |

СОДЕРЖАНИЕ

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc151150124)

[2 КОД ПРОГРАММЫ 4](#_Toc151150125)

[3 НАБОР ТЕСТОВ 7](#_Toc151150126)

[4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ 8](#_Toc151150127)

[5 ВЫВОДЫ 9](#_Toc151150128)

[6 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ 10](#_Toc151150129)

# 1 ПОСТАНОВКА ЗАДАЧИ

Написать на любом языке программирования (или доработать листинг 1) лексический анализатор на базе конечного автомата входного языка, описанного диаграммой состояний рис. 1.

Изображение выглядит как текст, диаграмма, круг, Шрифт

Автоматически созданное описание

# 2 КОД ПРОГРАММЫ

В качестве языка программирования был выбран Python.

with open('lex\_test.txt',encoding='utf-8') as file:

    data = file.read()

    data+='\n'

    status = "H"

    n = 0

    letter = data[n]

    key\_words = list()

    idents = list()

    numbers = list()

    operations = list()

    seps = list()

    mistakes = list()

    while (n!=len(data)):

            match status:

                case "H":

                    flag\_err = "OK"

                    while letter in ("\n"," ","\t"):

                        n+=1

                        try:

                            letter = data[n]

                        except:

                            print(f"Ключевые слова: {key\_words}")

                            print(f"Идентификаторы: {idents}")

                            print(f"Числа: {numbers}")

                            print(f"Операторы присваивания: {operations}")

                            print(f"Разделители: {seps}")

                            print(f"Исключения: {mistakes}")

                            exit()

                    if letter.isalpha() or letter=='\_':

                        status = "ID"

                    elif letter.isdigit() or letter in '+-.':

                        status = "NM"

                    elif letter == ":":

                        status = "ASGN"

                    else:

                        status = "DLM"

                case "ID":

                    flag\_err = "OK"

                    id = letter

                    n+=1

                    letter=data[n]

                    while letter.isalpha() or letter.isdigit() or letter == '\_':

                        id+=letter

                        n+=1

                        letter = data[n]

                    if id in ('for','do'):

                        key\_words.append(id)

                    else:

                        idents.append(id)

                    status = "H"

                case "NM":

                    flag\_err = "OK"

                    num = letter

                    n+=1

                    letter = data[n]

                    while letter.isdigit() or letter in '.+-eElLfF':

                        num+=letter

                        n+=1

                        letter = data[n]

                    c = num.count('f')+num.count('F')+num.count('l')+num.count('L')

                    if c == 1:

                        if num[-1] in 'fFlL':

                            try:

                                float(num[:-1])

                            except:

                                flag\_err = "NM"

                                status = "ERR"

                        else:

                            flag\_err = "NM"

                            status = "ERR"

                    elif c > 2:

                        flag\_err = "NM"

                        status = "ERR"

                    elif c==0:

                        try:

                            float(num)

                        except:

                            flag\_err = "NM"

                            status = "ERR"

                    if flag\_err == "OK":

                        numbers.append(num)

                        status = "H"

                case "ASGN":

                    flag\_err = "OK"

                    op = letter

                    n+=1

                    letter = data[n]

                    op+=letter

                    if op == ':=':

                        operations.append(op)

                        status = "H"

                    else:

                        flag\_err = "ASGN"

                        status = "ERR"

                    n+=1

                    letter=data[n]

                case "DLM":

                    flag\_err = "OK"

                    if letter in ('(',')',';','<','>','='):

                        seps.append(letter)

                        n+=1

                        letter = data[n]

                        status = "H"

                    else:

                        status = "ERR"

                case "ERR":

                    if flag\_err == "ASGN":

                        mistakes.append(op)

                    if flag\_err == "NM":

                        mistakes.append(num)

                    if flag\_err == "DLM":

                        mistakes.append(letter)

                    status = "H"

# 3 НАБОР ТЕСТОВ

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 – Содержание файла теста №1

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 3 – Содержание файла теста №2

Изображение выглядит как снимок экрана, текст, черный

Автоматически созданное описание

Рисунок 4 – Содержание файла теста №3

# 4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 5 – Результат теста №1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – Результат теста №2

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – Результат теста №3

# 5 ВЫВОДЫ

В ходе выполнения двух практических работ были получены теоретические и практические навыки принципа работы и написания лексического анализатора.

# 6 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Теория формальных языков. Лекция 4 – СДО (online-edu.mirea.ru)
2. Алгоритмы и методы – обратная польская запись: [Электронный ресурс]. URL: <https://www.interface.ru/home.asp?artid=1492> (дата обращения 04.09.2023)