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 №2

по дисциплине “Теория языков программирования и методы трансляции”

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

Энгельс 2023

**2.Вариант № 15**

**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.

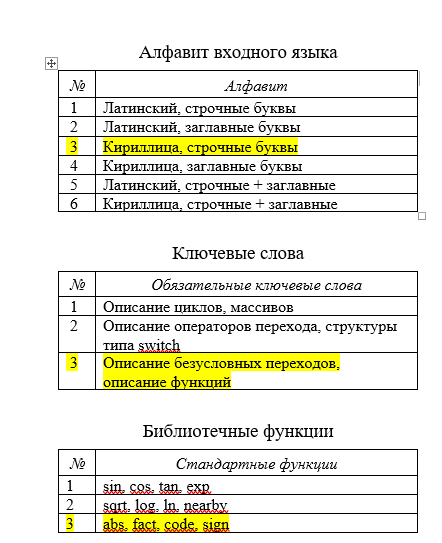


Рисунок 1.Изначальные данные для анализатора

**Выбранные ключевые слова:**

"программа", "функц", "вещ","нач", "кон", "возврат","ввод", "вывод", "переход","метка",программа"

**Выбранные стандартные функции:**

abs (модуль), fact (факт)

**3.Внутренние таблицы сканера:**

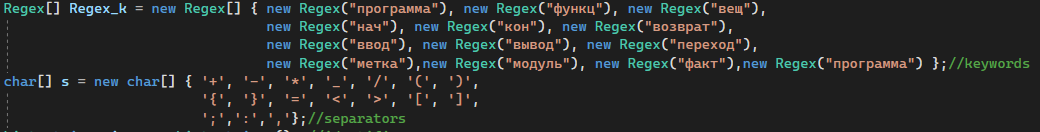


Рисунок 2.Таблица ключевых слов и разделителей

**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:**

Потешкин Александр Дмитриевич

**Порядок контроля и приёмки**

**5.Листинг программы**

**Листинг программы:**

string code =

"программа 1231\n" +

"функц Сложение(вещ x, вещ y)\n" +

"нач\n" +

" возврат x + y;\n" +

"кон\n" +

"\n" +

"нач вещ a, вещ b;\n" +

"метка 1;\n" +

" a = 0;\n" +

" ввод b;\n" +

" c = модуль(Сложение(a,b));\n" +

"переход 1;\n" +

" вывод c;\n" +

"кон\n";

Console.WriteLine(code);

Regex regex = new Regex(@"\([k,s,i,c],\d\*\)");

Regex[] Regex\_k = new Regex[] { new Regex("программа"), new Regex("функц"), new Regex("вещ"),

new Regex("нач"), new Regex("кон"), new Regex("возврат"),

new Regex("ввод"), new Regex("вывод"), new Regex("переход"),

new Regex("метка"),new Regex("модуль"),new Regex("программа"),new Regex("факт") };//keywords

char[] s = new char[] { '+', '-', '\*', '\_', '/', '(', ')',

'{', '}', '=', '<', '>', '[', ']',

';',':',','};//separators

List<string> i = new List<string>(); //identifiers

List<string> c = new List<string>(); //constants

string newcode = "";

foreach (var j in code)

{

if (s.Contains(j)) { newcode += $" (s,{Array.IndexOf(s,j)}) "; }

else { newcode += j; }

}

for (int j = 0; j < Regex\_k.Count(); j++)

{

if (Regex\_k[j].Matches(newcode).Count != 0) { newcode = Regex\_k[j].Replace(newcode, $" (k,{j}) "); }

}

List<string> sd = newcode.Split(' ').ToList();

newcode = string.Join(" ",sd);

for (var r = 0; r < sd.Count; r++)

{

if (sd[r].Trim(' ', '\n') != "")

{

if (sd[r].Trim(' ','\n')[0] != '(' && sd[r].Trim(' ', '\n')[0] != ')')

{

if (Regex.IsMatch(sd[r], @"('[^']\*'|\d+)"))

{

c.Add(sd[r].Trim(' ', '\n'));

sd[r] = $"(c,{c.Count - 1})";

}

else if (Regex.IsMatch(sd[r], @"\w+"))

{

i.Add(sd[r].Trim(' ', '\n'));

for (int l = r; l < sd.Count; l++)

{

if (sd[l] == i.Last())

{ sd[l] = Regex.Replace(sd[l], @"\b\w+\b", $"(i,{i.Count - 1})"); }

}

}

}

}

}

newcode = string.Join(" ",sd);

Console.WriteLine(newcode);

void WriteElem(string name, List<string> Elem)

{

Console.WriteLine($"{name}: ");

foreach (var j in Elem)

{

Console.Write($"{j} ");

}

Console.WriteLine("\n");

}

WriteElem("Идентификаторы", i);

WriteElem("Константы", c);

Console.WriteLine($"Ключевые слова: ");

foreach (var j in Regex\_k)

{

Console.Write($"{j} ");

}

Console.WriteLine("\n");

Console.WriteLine($"Разделители: ");

foreach (var j in s)

{

Console.Write($"{j} ");

}

Console.WriteLine("\n");

**6.Отладочный пример работы программы**

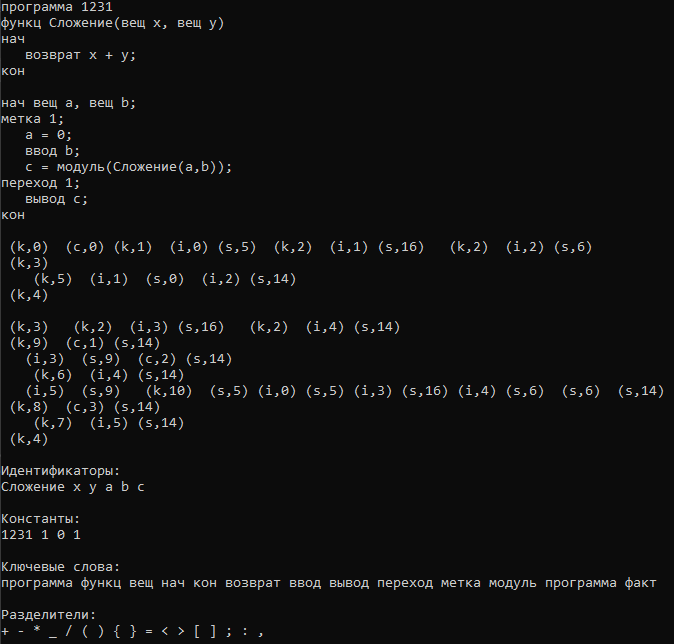


Рисунок 3.Пример работы программы