# Построение формального языка L

WHILE( <AB> ) [<FUNCTION([<PAR>, <PAR>, ...])>];

WHILE - входной терминальный символ

<AB > - условное выражение

<FUNCTION> - некоторая функция, которая может отсутствовать

<PAR> - параметры функции, которые тоже могут отсутствовать

Пример правильного синтаксиса:

WHILE(A>44<b=56+67) clrscr();

A>44<b=56+67 - условное выражение

clrscr() - функция, без параметров

# Подбор грамматики G[Z] по языку L

Любая грамматика, к примеру G[Z], содержит следующие базисные элементы Vt, Vn, Z, P, где:

Vt - словарь терминальных символов

Vn - словарь нетерминальных символов

Z - начальный нетерминальный символ

P - множество правил вывода

G[<оператор>]:

1. <Оператор> 🡪 while(<AB> ) [<Функция>];
2. <AB>🡪 T|<AB> < T|<AB> > T|<AB> <= T | <AB> >= T | <AB> != T
3. <T> 🡪 O | T\*O | T\*\*O | T+O | T-O | T/O
4. <O> 🡪 (<AB>) | <IDENT> | <ЦБЗ>
5. <IDENT> 🡪 Б{Б|Ц}
6. <ЦБЗ> 🡪 Ц{Ц}
7. <Функция> 🡪 <IDENT>([<PAR>{,<PAR>}])
8. <PAR> 🡪 <IDENT> | <ЦБЗ> |

# Классификация G[Z]

G[<оператор>]:

1. <Оператор> 🡪 while(<AB> ) [<Функция>];
2. <AB>🡪 T|<AB> < T|<AB> > T|<AB> <= T | <AB> >= T | <AB> != T
3. <T> 🡪 O | T\*O | T\*\*O | T+O | T-O | T/O
4. <O> 🡪 (<AB>) | <IDENT> | <ЦБЗ>
5. <IDENT> 🡪 Б{Б|Ц}
6. <ЦБЗ> 🡪 Ц{Ц}
7. <Функция> 🡪 <IDENT>([<PAR>{,<PAR>}])
8. <PAR> 🡪 <IDENT> | <ЦБЗ> |

Сделаем замену нетерминальных символов:

<Оператор> 🡪 Z

<AB> 🡪 A

<T> 🡪 B

<O> 🡪 C

<Функция > 🡪 D

<IDENT> 🡪 E

<ЦБЗ> 🡪 F

<PAR> 🡪 G

Сделаем замену терминальных символов:

WHILE 🡪 a

( 🡪 b

) 🡪 c

; 🡪 d

Ц 🡪 f

Б 🡪 g

, 🡪 h

G[Z]:

1. Z 🡪 abAc[D]d
2. A🡪 B|A<B|A > B|A <= B |A >= B |A != B
3. B 🡪 C | B\*C | B\*\*C | B+C | B-C | B/C
4. C 🡪 bAc | E | F
5. E 🡪 g{g|f}
6. F 🡪 f{f}
7. D 🡪 Eb[G{hG}]c
8. G 🡪 E | F |

Вывод : G[Z] - контекстно-свободная грамматика.

# Выбор метода анализа

Хотя однозначность в общем случае для контекстно-зависимых грамматик не доказана, ее использование возможно для грамматик в которых однозначность очевидна. Наиболее хорошо разработанным методом анализа, для данного типа грамматик является, метод рекурсивного спуска.

# Диагностика и нейтрализация ошибок

Разработанный алгоритм относится к общеизвестному методу синтаксического разбора, предложенный Айронсом.

Основная идея метода состоит в том, что по контексту без возврата отбрасываются те символы, которые привели в тупиковую ситуацию и разбор продолжается.

Приведем пример синтаксического разбора:

While (A > ) cls();

1. Z 🡪 abAc[D]d
2. A🡪 B|A<B|A > B|A <= B |A >= B |A != B
3. B 🡪 C | B\*C | B\*\*C | B+C | B-C | B/C
4. C 🡪 bAc | E | F
5. E 🡪 g{g|f}
6. F 🡪 f{f}
7. D 🡪 Eb[G{hG}]c
8. G 🡪 E | F |

**Z**

a b A c D

B A E

C B g{g} b G c

E C

g E

g

While ( A > ) cls ( ) ;

тупиковая ситуация

# Тестирование на цепочках

Протокол работы синтаксического распознавателя оператора цикла while языка С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - While(a>)cls();

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - a

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - )

Проверка на ЦБЗ, текущий символ - )

Проверка на FUNC, текущий символ - c

Проверка на IDENT, текущий символ - c

Найденные ошибки в строке While(a>)cls();

Предупреждение: Отсутствует условие ()

Не найден идентификатор или ЦБЗ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(1<(3\*(43+5\*(3-4-(4<454)))) ;

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 1

Проверка на ЦБЗ, текущий символ - 1

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 5

Проверка на ЦБЗ, текущий символ - 5

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на FUNC, текущий символ - ;

Проверка на IDENT, текущий символ - ;

Найденные ошибки в строке while(1<(3\*(43+5\*(3-4-(4<454)))) ;

Отсутствует )

Предупреждение: отсутствует имя функции

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - 435 4 whilei>>3t\*(ewqw++()\*we) rscr(456,345,rtgrt,345444rtr,,,)

Найден While проверка началась с символа - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - e

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - +

Проверка на ЦБЗ, текущий символ - +

Проверка на TERM

Проверка на O

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - w

Проверка на FUNC, текущий символ - r

Проверка на IDENT, текущий символ - r

Проверка на PAR, текущий символ - 4

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на PAR, текущий символ - 3

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на PAR, текущий символ - r

Проверка на IDENT, текущий символ - r

Проверка на PAR, текущий символ - 3

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на PAR, текущий символ - ,

Проверка на IDENT, текущий символ - ,

Проверка на ЦБЗ, текущий символ - ,

Найденные ошибки в строке 435 4 whilei>>3t\*(ewqw++()\*we) rscr(456,345,rtgrt,345444rtr,,,)

Отсутствует (

Отсутствует )

Предупреждение: Отсутствует условие ()

Отсутствует ; после функции

Параметр функции не может начинатся с цифры

Неизвестный идентификатор(ы) -

435, 4,

Не найден идентификатор или ЦБЗ

Идентификатор не может начинаться с цифры

Не найден или не верный параметр

Неизвестная знаковая конструкция

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - whiLE(43-(sss<233)fewfew) sd(we)

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - s

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - f

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ - s

Проверка на PAR, текущий символ - w

Проверка на IDENT, текущий символ - w

Найденные ошибки в строке whiLE(43-(sss<233)fewfew) sd(we)

Отсутствует ; после функции

Пропущено / или \* или \*\* или + или -

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(i>344>(5445<<2323\*343243+2343\*(w))>23\*232) clrscr(,)

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 5

Проверка на ЦБЗ, текущий символ - 5

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - w

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ - c

Проверка на PAR, текущий символ - ,

Проверка на IDENT, текущий символ - ,

Проверка на ЦБЗ, текущий символ - ,

Найденные ошибки в строке while(i>344>(5445<<2323\*343243+2343\*(w))>23\*232) clrscr(,)

Отсутствует ; после функции

Не найден или не верный параметр

Неизвестная знаковая конструкция

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(u>r+erhgeerg(e>=rwe+++r+788erwe++)) scr(eee,qee<)

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - u

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - r

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - e

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - e

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - r

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - r

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 7

Проверка на ЦБЗ, текущий символ - 7

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ - s

Проверка на PAR, текущий символ - e

Проверка на IDENT, текущий символ - e

Проверка на PAR, текущий символ - q

Проверка на IDENT, текущий символ - q

Проверка на PAR, текущий символ - <

Проверка на IDENT, текущий символ - <

Проверка на ЦБЗ, текущий символ - <

Найденные ошибки в строке while(u>r+erhgeerg(e>=rwe+++r+788erwe++)) scr(eee,qee<)

Отсутствует ; после функции

Пропущено / или \* или \*\* или + или -

Идентификатор не может начинаться с цифры

Не найден или не верный параметр

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(i>77777u777) clrscr(,...,)

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 7

Проверка на ЦБЗ, текущий символ - 7

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ - c

Проверка на PAR, текущий символ - ,

Проверка на IDENT, текущий символ - ,

Проверка на ЦБЗ, текущий символ - ,

Найденные ошибки в строке while(i>77777u777) clrscr(,...,)

Отсутствует ; после функции

Идентификатор не может начинаться с цифры

Не найден или не верный параметр

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Обрабатываем строчку - while(4545>>445--- ;

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 4

Проверка на ЦБЗ, текущий символ - 4

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ -

Проверка на ЦБЗ, текущий символ -

Проверка на FUNC, текущий символ - ;

Проверка на IDENT, текущий символ - ;

Найденные ошибки в строке while(4545>>445--- ;

Отсутствует )

Предупреждение: отсутствует имя функции

Не найден идентификатор или ЦБЗ

Неизвестная знаковая конструкция

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(i>=0);

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 0

Проверка на ЦБЗ, текущий символ - 0

Проверка на FUNC, текущий символ - ;

Проверка на IDENT, текущий символ - ;

Найденные ошибки в строке while(i>=0);

Предупреждение: отсутствует имя функции

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(i>=0) 544();

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 0

Проверка на ЦБЗ, текущий символ - 0

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ - 5

Найденные ошибки в строке while(i>=0) 544();

Отсутствует ; после функции

Предупреждение: отсутствует имя функции

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - whilei>=0) clrscr();13

Найден While проверка началась с символа - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - i

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 0

Проверка на ЦБЗ, текущий символ - 0

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ - c

Найденные ошибки в строке whilei>=0) clrscr();13

Отсутствует (

Предупреждение: Отсутствует условие ()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - whilertt<=243+++344-23!=345 wwqwq;

Найден While проверка началась с символа - r

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - r

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - =

Проверка на ЦБЗ, текущий символ - =

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на FUNC, текущий символ - w

Проверка на IDENT, текущий символ - w

Найденные ошибки в строке whilertt<=243+++344-23!=345 wwqwq;

Отсутствует (

Отсутствует )

Не найден идентификатор или ЦБЗ

Не найден или не верный параметр

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обрабатываем строчку - while(dd>>3432\*23432)

Найден While проверка началась с символа - (

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - d

Проверка на AB

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 3

Проверка на ЦБЗ, текущий символ - 3

Проверка на TERM

Проверка на O

Проверка на IDENT, текущий символ - 2

Проверка на ЦБЗ, текущий символ - 2

Проверка на FUNC, текущий символ -

Проверка на IDENT, текущий символ -

Найденные ошибки в строке while(dd>>3432\*23432)

Отсутствует ; после функции

Предупреждение: отсутствует имя функции

Неизвестная знаковая конструкция

# Листинг программы

ВАРИАHТ # 10

Синтаксический распознователь оператора цикла While

Файл программы - А513.cpp

с текстом для распознования - test.513

с протоколом работы - error.513

Кафедpа : АСУ

Гpуппа : А-513

Студент : Стариков Дмитрий Александрович

Пpеподаватели : кандидат технических наук, доцент

Шоpников Юpий Владимиpович,

ассистент Панова Веpа Боpисовна

Дата : 30 мая 1997г.

//----------------------------------------------------------------------------

// Заголовочные файлы.

//----------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <conio.h>

#include <dos.h>

#define UP 72 // стрелка вверх

#define DOWN 80 // стрелка вниз

#define ALTX 45 // выход

#define F1 59

#define F2 60

#define F3 61

#define ESC 27

#define FL\_NAME "test.513"

#define FL\_TEST "error.513"

void open\_fl(void); // откpыть файл

void work\_space(void); // упpавляет pабочей областью

void print\_page(void); // печатает текст на экpан

void help\_line(void); // текст с помощью нижняя стpочка

void help(int);

int scan(); // Делает разбор строки

int my\_while(); // опеpатоp

int AB(); // арифметическое выражение

int TERM(); // Терм

int O(); // Операнд

int IDENT(); // Идентификатор

int ZBZ(); // Целое без знака

int FUNC(); // Функция

int PAR(); // Параметр

char TEXT[22][80];

int position; // Номер текущей страницы

int cur\_y=3; // положение куpсоpа на экpане

int x,y; // Текущая позиция при компиляции в строчке

int f=0; // При f=1 ошибки для функции

char screen[4096]; // Сохpаняет полную копию экpана

char screen1[4096]; // Сохpаняет часть экpана

char \*mistake[]={"Ошибок нет!!!", //0

"Опеpатоp while не найден", //1

"Отсутствует (", //2

"Отсутствует )", //3

"Предупреждение: Отсутствует условие ()", //4

"Отсутствует ; после функции", //5

"Предупреждение: отсутствует имя функции", //6

"Пропущено / или \* или \*\* или + или - ", //7

"", //8

"Параметр функции не может начинатся с цифры", //9

"Неизвестный идентификатор(ы) - ", //10

"Не найден идентификатор или ЦБЗ", //11

"Идентификатор не может начинаться с цифры", //12

"Не найден или не верный параметр", //13

"Неизвестная знаковая конструкция", //14

""}; //15

FILE \*fl\_t;

int mistake\_number[15]; //массив хранящий коды ошибок (0 или 1)

//где 1-ошибка присутсвует, 0 в противном случае

//mistake\_number[i], где i номер ошибки из mistake

char strange[100]; //перечень неизвестных идентификаторов

//найденых в строке во время разбора

int s=0; //текущая позиция в strange

void main()

{

open\_fl();

help\_line();

print\_page();

work\_space();

}

//----------------------------------------------------------------------------

// Открытие файла

//----------------------------------------------------------------------------

void open\_fl(void)

{

FILE \*fl;

int i;

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

\_setcursortype(\_NOCURSOR);

if((fl\_t= fopen(FL\_TEST,"w"))== NULL) exit(1);

if ((fl = fopen(FL\_NAME,"r"))==NULL)

{

window(18,10,60,16);

textbackground(GREEN);

textcolor(BLACK);

clrscr();

gotoxy(5,2);cprintf(" Немогу найти файл test.513");

gotoxy(5,3);cprintf("Проверте его наличие на диске");

gotoxy(5,4);cprintf(" или создайте новый");

gotoxy(3,6);cprintf("Для продолжения нажмите любую клавишу");

getch();

exit(0);

}

for(i=0;i<25;i++)

{

window(40-i,12-i/5,40+i,12+i/5);

textbackground(GREEN);

textcolor(BLACK);

clrscr();

delay(7);

}

textcolor(WHITE);

gotoxy(12,2);cprintf("Чтение строк для разбора");

gotoxy(12,3);cprintf(" произведеное из");

gotoxy(12,4);cprintf(" файла ");

gotoxy(12,5);cprintf(" test.513 ");

gotoxy(7,8);cprintf(" Для продолжения нажмите любую клавишу");

getch();

for (i=0; i<20; i++)

{

if (fgets(TEXT[i],78,fl)==NULL) break; // читать строку файла

}

TEXT[i][0]=NULL;

position=0; //обнуление позиции сдвига стpаниц

fclose(fl);

}

//----------------------------------------------------------------------------

// Рабочая область

//----------------------------------------------------------------------------

void work\_space(void)

{

char c;

while(1)

{

gotoxy(1,cur\_y);

c=getch();

switch(c)

{

case UP:

if(cur\_y!=3)

{

cur\_y--;

print\_page();

}

else

{

if(position!=0)

{

position--;

print\_page();

cur\_y=3;

}

}

break;

case DOWN:

if(cur\_y!=7)

{

cur\_y++;

if(TEXT[cur\_y+position-3][0]==NULL) cur\_y--;

print\_page();

}

else

{

cur\_y=7;

if(TEXT[cur\_y+position-2][0]!=NULL)

{

position++;

print\_page();

}

}

break;

case F1:

help(1);

break;

case F2:

help(2);

break;

case F3:

help(3);

break;

case ALTX:

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

fclose(fl\_t);

clrscr();

exit(1);

case ESC:

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

fclose(fl\_t);

exit(1);

}

}

}

//----------------------------------------------------------------------------

// Распечатывает текст, выводит на экран ошибки, найденные в обрабатываемой

// (текущей) строке

//----------------------------------------------------------------------------

void print\_page()

{

int i,xx=10;

puttext(1,1,80,25,screen);

for(i=0;i<=15;i++) //очистка массива ошибок

mistake\_number[i] = 0;

strange[0]=NULL;

s=0;

x=cur\_y+position-3;

y=0;

f=0;

fprintf(fl\_t,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

fprintf(fl\_t,"Обрабатываем строчку - %s\n",TEXT[x]);

scan(); //вызывает сканер для разбора строки

fprintf(fl\_t,"\n Найденные ошибки в строке %s\n",TEXT[x]);

for(i=0;i<=15;i++)

if(mistake\_number[i] == 1)

{

gotoxy(4,xx++);

printf("%s ",mistake[i]);

fprintf(fl\_t," %s\n",mistake[i]);

if(i==10 && mistake\_number[i] == 1)

{

printf(" %s",strange);

fprintf(fl\_t," %s\n",strange);

}

}

if(xx == 10) //ошибок нет, т.к. координаты строки

{ //неизменились (xx)

gotoxy(4,xx++);

printf("%s ",mistake[0]);

}

textbackground(BLACK);

textcolor(WHITE);

if(TEXT[0][0]!=NULL)

{

for (i=1;i<6;i++)

{

if(TEXT[i-1+position][0]==NULL) break;

gotoxy(1,i+2);

puts(TEXT[i-1+position]);

}

}

gotoxy(1,cur\_y);

textbackground(RED);

clreol();

puts(TEXT[cur\_y+position-3]);

}

//----------------------------------------------------------------------------

// Выводит на экран текст с помощью

//----------------------------------------------------------------------------

void help\_line(void)

{

window(1,1,80,25);

textbackground(BLACK);

clrscr();

textbackground(GREEN);

textcolor(BLACK);

gotoxy(1,1);

clreol();

printf(" Borland C++ Веpсия только для опеpатоpа WHILE");

gotoxy(1,25);

textbackground(LIGHTCYAN);

clreol();

cprintf(" F1 - Помощь F2 - Грамматика F3 - Язык оператора ALT+X-Выход");

gotoxy(1,2);

textbackground(LIGHTCYAN);

clreol();

printf("------------------------------- CОДЕРЖИМОЕ ФАЙЛА -------------------------------");

gotoxy(1,8);

textbackground(LIGHTCYAN);

clreol();

printf("------------------------------ СООБЩЕНИЯ ОБ ОШИБКАХ ----------------------------");

gotoxy(1,9);

gettext(1,1,80,25,screen);

gettext(1,9,80,24,screen1);

}

//----------------------------------------------------------------------------

// Выводит на экран текст, в зависимости от значения n

//----------------------------------------------------------------------------

void help(int n)

{

char string[4096];

gettext(1,8,80,22,string);

window(1,8,80,22);

textbackground(CYAN);

textcolor(BLACK);

clrscr();

gotoxy(1,1);

if(n ==1)

{

cprintf("\n HОВОСИБИРСКИЙ ГОСУДАРСТВЕHHЫЙ ТЕХHИЧЕСКИЙ УHИВЕРСИТЕТ\n\r\r");

cprintf(" Куpсовая pабота по дисциплине СИСТЕМHОЕ ПРОГРАММИРОВАHИЕ\n\r\r");

cprintf(" Синтаксический pаспознаватель\n\n\r\r\r");

cprintf(" Используемые клавиши:\r\n");

cprintf(" F1 - данный HELP.\r\n");

cprintf(" F2 - гpамматика языка.\r\n");

cprintf(" F3 - язык оператора.\r\n");

cprintf(" Esc - выход из программы.\r\n");

cprintf(" Alt-X - выход из программы.\r\n\n\n");

cprintf(" (c) 1997 Стариков Дмитрий Александрович");

}

if(n == 2)

{

cprintf("\n ГРАММАТИКА ЯЗЫКА\n\n\r\r");

cprintf(" 1. <оператор> -> WHILE (<AB>)[<FUNCTION>];\n\r\r");

cprintf(" 2. <AB> -> T|<AB> >T |<AB> <T |<AB> <=T |<AB> >=T |<AB> !=T\n\r\r\r");

cprintf(" 3. T -> O | T+O | T-O | T\*O | T/O | T\*\*O\n\r\r\r");

cprintf(" 4. O ->(<AB>) | <IDENT> | <ЦБЗ>\n\r\r\r");

cprintf(" 5. <IDENT> -> Б{Б|Ц}\r\r\r\r\n");

cprintf(" 6. <ЦБЗ> -> Ц{Ц}\r\n");

cprintf(" 7. <FUNCTION> -> <IDENT>([<PAR>{,<PAR>}])\r\n");

cprintf(" 8. <PAR> -> <IDENT> | <ЦБЗ> \r\n");

}

if(n == 3)

{

cprintf("\n ЯЗЫК ОПЕРАТОРА\n\n\n\r\r");

cprintf(" WHILE(AB) [FUNCTION([PAR,PAR,...])];\r\n\n");

cprintf(" AB - Выражение \r\n");

cprintf(" FUNCTION - функция\r\n");

cprintf(" PAR - параметры функции, могут быть цифры или текст \r\n");

cprintf(" количество их не ограничено\r\n\n");

cprintf(" Пробелы между символами недопустимы\r\n\n\n");

}

getch();

puttext(1,8,80,22,string);

window(1,1,80,25);

}

//----------------------------------------------------------------------------

// Сканирует до появления While

//----------------------------------------------------------------------------

int scan()

{

int k,j,w; //счетчики

mistake\_number[1]=1;

for(j=0;;j++)

{

if(TEXT[x][j] == NULL) break;

if(TEXT[x][j] == 'W')

{

y=j; //если нашли W или w

if(my\_while() == 1) break; //то вызываем

} //my\_while

if(TEXT[x][j] == 'w')

{

y=j;

if(my\_while() == 1) break;

}

if(TEXT[x][j] != ' ')

{

strange[s++]=TEXT[x][j];

mistake\_number[10]=1;

}

if(TEXT[x][j] == ' ')

{

strange[s++]=',';

strange[s++]=' ';

}

}

strange[s]=NULL;

return(1);

}

//----------------------------------------------------------------------------

// Обрабатывает While и вызывает обработку функций

//----------------------------------------------------------------------------

int my\_while()

{

char str[10];

int k,j,w; //счетчики

for(w=0;w<5;w++)

{

if(TEXT[x][y] == NULL) break;

str[w]=toupper(TEXT[x][y++]); //Toupper - переводит

} //все буквы в заглавные

str[w]=NULL;

if(strcmp(str,"WHILE") == 0) //Если While найден то производим

{ // дальнейший разбор

fprintf(fl\_t,"Найден While проверка началась с символа - %c\n",TEXT[x][y]);

if(TEXT[x][y] != '(') mistake\_number[2]=1;

else y++;

AB();

if(TEXT[x][y] != ')') mistake\_number[3]=1;

y++;

mistake\_number[1]=0;

FUNC();

return(1);

}

else return(0);

}

//----------------------------------------------------------------------------

// Обработка выражения

//----------------------------------------------------------------------------

int AB()

{

char signal[4]="><!=";

int z;

fprintf(fl\_t,"Проверка на AB\n");

TERM();

if(TEXT[x][y] == '<')

{

y++;

if(TEXT[x][y] == '=') y++;

for(z=0;z<4;z++)

if(TEXT[x][y] == signal[z]) //Если слишком

{ //сложная знаковая

mistake\_number[14]=1; //конструкция

y++; //то ошибка

z=0;

}

AB();

return(1);

}

if(TEXT[x][y] == '>')

{

y++;

if(TEXT[x][y] == '=') y++;

for(z=0;z<4;z++)

if(TEXT[x][y] == signal[z])

{

mistake\_number[14]=1;

y++;

z=0;

}

AB();

return(1);

}

if(TEXT[x][y] == '=')

{

y++;

if(TEXT[x][y] == '=')

{

y++;

}

for(z=0;z<4;z++)

if(TEXT[x][y] == signal[z])

{

mistake\_number[14]=1;

y++;

z=0;

}

AB();

return(1);

}

if(TEXT[x][y] == '!')

{

y++;

for(z=0;z<3;z++)

if(TEXT[x][y] == signal[z])

{

mistake\_number[14]=1;

y++;

z=0;

}

if(TEXT[x][y+1] == '=')

{

y++;

}

AB();

return(1);

}

if(TEXT[x][y] == NULL) return(1);

return(1);

}

//----------------------------------------------------------------------------

// Обработка терма

//----------------------------------------------------------------------------

int TERM()

{

fprintf(fl\_t,"Проверка на TERM\n");

O();

if(TEXT[x][y] == '/')

{

y++;

TERM();

}

else if(TEXT[x][y] == '\*')

{

y++;

if(TEXT[x][y] == '\*') y++;

TERM();

}

else if(TEXT[x][y] =='+')

{

y++;

if(TEXT[x][y] == '+' && TEXT[x][y+1] == ')' )

{ //возможна обработка выражения

y++; //типа y++

return(0); //или y+++e

}

if(TEXT[x][y] == '+'&& TEXT[x][y+1] == '+') y+=2;

TERM();

}

else if(TEXT[x][y] =='-')

{

y++;

if(TEXT[x][y] == '-' && TEXT[x][y+1] == ')' )

{ //для y--

y++;

return(0);

}

if(TEXT[x][y] == '-'&& TEXT[x][y+1] == '-') y+=2;

TERM();

}

else if(TEXT[x][y] ==NULL ) return(1);

else if(TEXT[x][y] != '+' && TEXT[x][y]!='\*' && TEXT[x][y]!=')' && TEXT[x][y] != '/' && TEXT[x][y]!='-' && TEXT[x][y] != '<' && TEXT[x][y]!='>' && TEXT[x][y]!='!' && TEXT[x][y]!=' '&& TEXT[x][y]!='=')

{

mistake\_number[7]=1;

AB();

}

return(1);

}

//----------------------------------------------------------------------------

// Обработка операнда

//----------------------------------------------------------------------------

int O()

{

fprintf(fl\_t,"Проверка на O\n");

if(TEXT[x][y] == '(')

{

y++;

if(TEXT[x][y] == ')')

{

mistake\_number[4]=1;

y++;

}

else

{

AB();

if(TEXT[x][y] == '(')

{

AB();

}

if(TEXT[x][y] != ')')

{

mistake\_number[3]=1;

y++;

}

else y++;

}

}

else

{

if(IDENT() == 0)

if(ZBZ() == 0)

if(f==0) mistake\_number[11]=1;

return(0);

}

return(0);

}

//----------------------------------------------------------------------------

// Обработка идентификатора

//----------------------------------------------------------------------------

int IDENT()

{

fprintf(fl\_t,"Проверка на IDENT, текущий символ - %c\n",TEXT[x][y]);

if((isalpha(TEXT[x][y])) != NULL)

{

while(isalpha(TEXT[x][y]) || isdigit(TEXT[x][y]))

y++;

return(1);

}

else return(0);

}

//----------------------------------------------------------------------------

// Целое без знака

//----------------------------------------------------------------------------

int ZBZ()

{

fprintf(fl\_t,"Проверка на ЦБЗ, текущий символ - %c\n",TEXT[x][y]);

if((isdigit(TEXT[x][y])) != NULL)

{

y++;

while(1)

{

if((isalpha(TEXT[x][y])) != NULL)

{

/\*Если найден идентификатор,\*/ if(f==0) mistake\_number[12]=1;

/\*начинающийся с цифры, \*/ else mistake\_number[9]=1;

/\*то ошибка: \*/ while(isalpha(TEXT[x][y]) || isdigit(TEXT[x][y]))

/\*если f=0 то в идентификаторе\*/ y++;

/\*если f=1 то в параметре \*/ return(1);

/\* функции \*/ }

if((isdigit(TEXT[x][y])) == NULL) break;

y++;

}

return(1);

}

else return(0);

}

//----------------------------------------------------------------------------

// Обработка функции

//----------------------------------------------------------------------------

int FUNC()

{

f=1;

fprintf(fl\_t,"Проверка на FUNC, текущий символ - %c\n",TEXT[x][y]);

for(;;y++)

if(TEXT[x][y] != ' ') break;

if(IDENT()==0)

{

mistake\_number[6]=1;

while(isalpha(TEXT[x][y]) || isdigit(TEXT[x][y]))

y++;

}

if(mistake\_number[6] ==0)

if(TEXT[x][y] =='(')

{

y++;

if(TEXT[x][y] == ')')

{

mistake\_number[4]=1;

y++;

}

else while(6)

{

if(PAR() != 0)

{

if(TEXT[x][y] == ')')

{

y++;

break;

}

if(TEXT[x][y] == ';') break;

if(TEXT[x][y] == ',') y++;

}

else

{

mistake\_number[13]=1;

break;

}

}

}

else

{

mistake\_number[2]=1;

mistake\_number[13]=1;

}

if(mistake\_number[13]==1)

{

for(;;y++)

{

if(TEXT[x][y] == NULL)

{

mistake\_number[5]=1;

mistake\_number[3]=1;

break;

}

if(TEXT[x][y] == ';')

{

mistake\_number[3]=1;

break;

}

if(TEXT[x][y] == ')')

{

y++;

break;

}

}

}

if(TEXT[x][y] != ';') mistake\_number[5]=1;

return(1);

}

//----------------------------------------------------------------------------

// Обработка параметров функции

//----------------------------------------------------------------------------

int PAR()

{

fprintf(fl\_t,"Проверка на PAR, текущий символ - %c\n",TEXT[x][y]);

if(IDENT() == 0)

if(ZBZ() == 0)

{

mistake\_number[13]=1;

return(0);

}

return(1);

}

# Список литературы:

1. Д. Грис Конструирование компиляторов для цифровых вычислительных машин. М., Мир, 1975.
2. Лебедев В.Н. Введение в системы программирования. М., «Статистика», 1975.
3. Шорников Ю.В., Шегал Б.Р. Методическое пособие для выполнения курсовой работы по системному программированию. Новосибирск, 1992.