

Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального образования
«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Кафедра ТК

Курсовая работа

по дисциплине «Системное программное обеспечение»

на тему: «Разработка отдельных фаз компиляции для заданного входного языка»

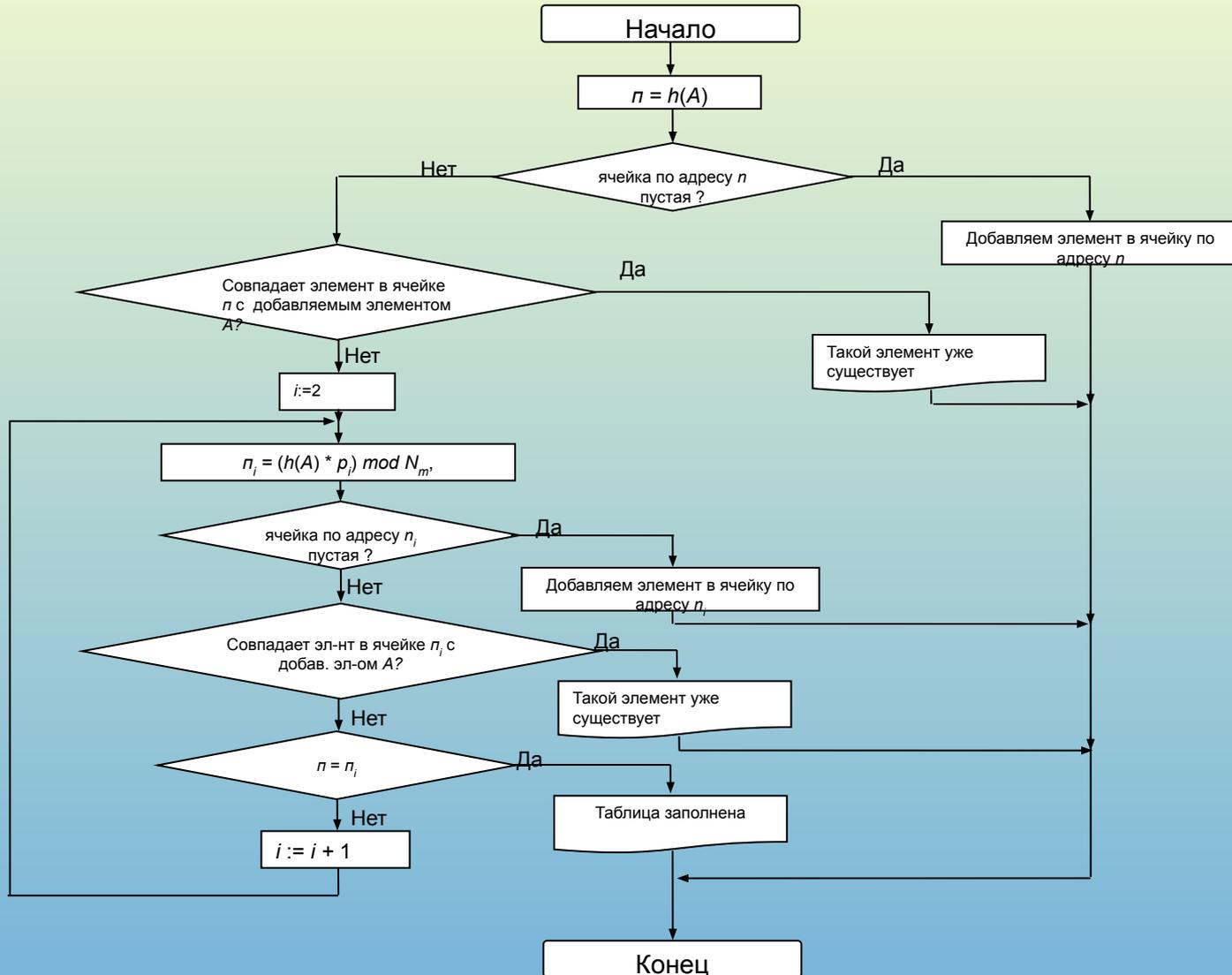
Выполнил: ст. гр. УТС-412

Ковалев Е. Н.

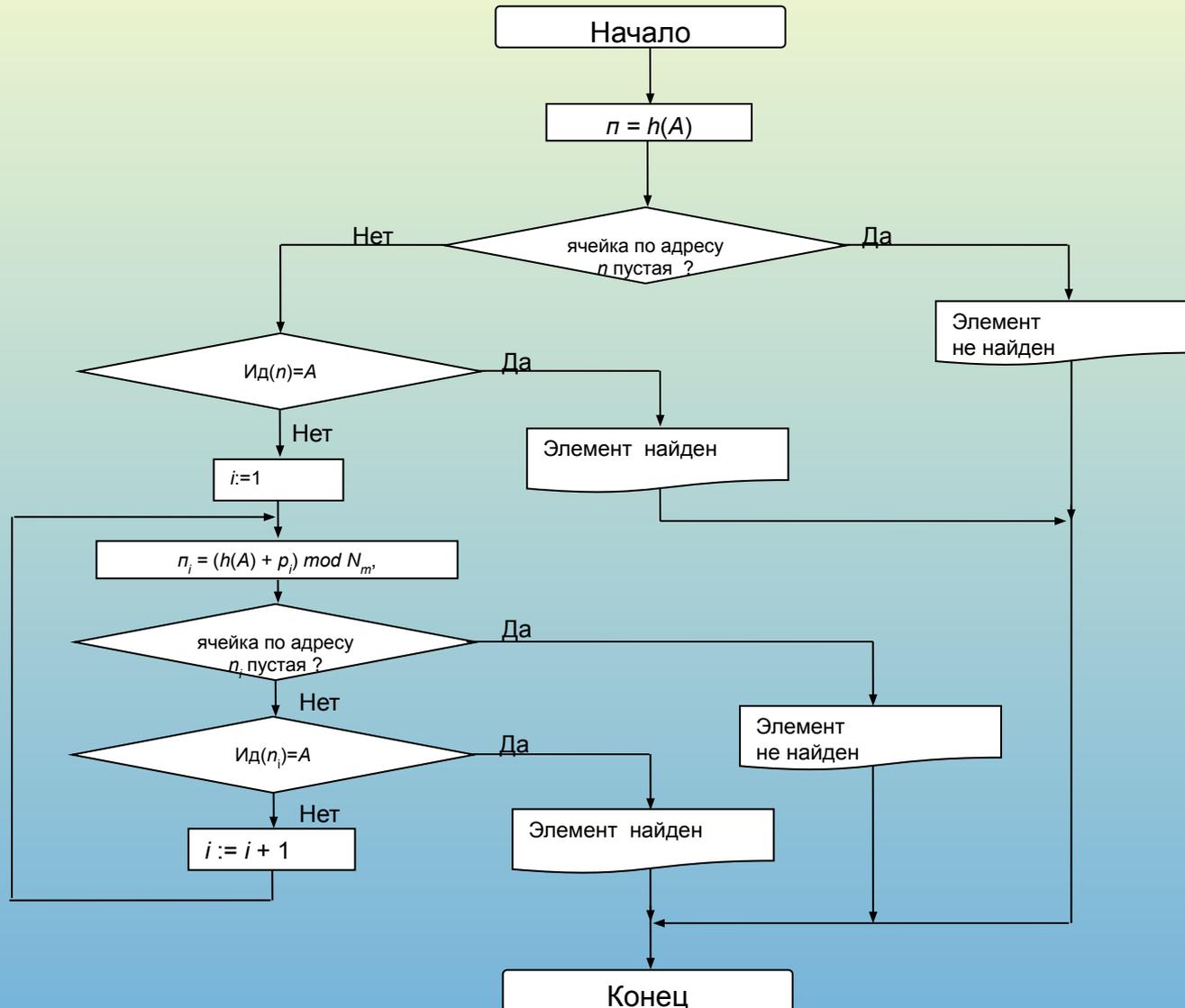
Консультант Карамзина А.Г.

Уфа 2007

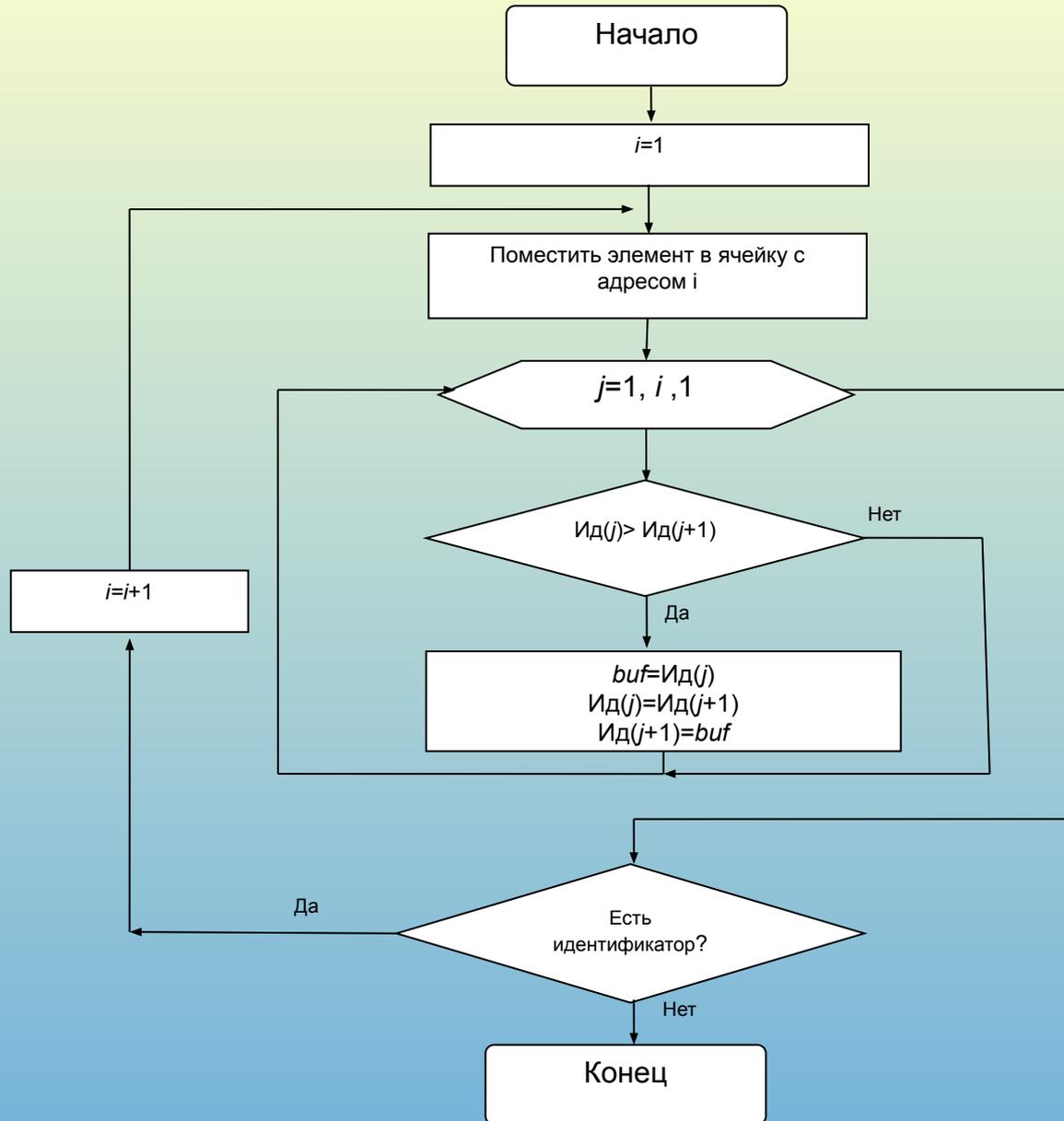
Блок-схема добавления элемента в таблицу идентификаторов методом хеш-адресации



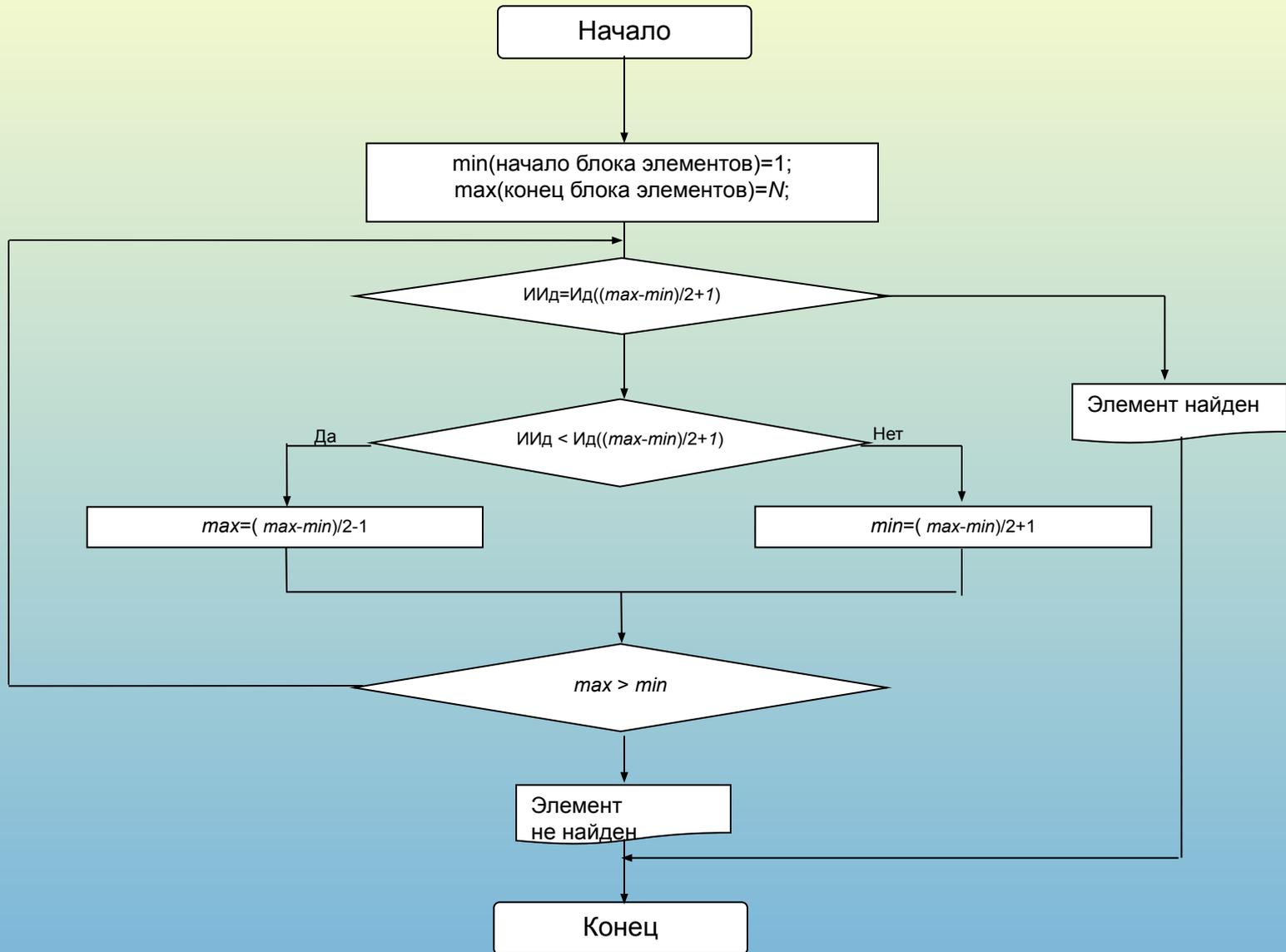
Блок-схема алгоритма поиска элемента в таблице идентификаторов методом хеш-адресации



Блок-схема алгоритма добавления элемента в упорядоченный список



Блок-схема алгоритма поиска элемента в упорядоченном списке



Результаты работы организации таблиц идентификаторов обоими методами

Form1

1 часть курсовой | 2 часть курсовой | 3 часть курсовой

Загрузить

Идентификаторы

sdf
wqer
sdh
xsn
уцен
вао
sghs
sghsgh
ety
счит
ко
п
роеовно
варавр
итит
уещркр
арфр
sgha
h
bvfnfn
fn
n
sfns
msrt
тыкт
snsnsmtu
k
гелегл
еще
т
чстм
ь
mbckuk

Рехэширование

ХФ	Иден-р
1	
2	
3	
4	
5	
6	
7	fgilFJgg
8	
9	
10	

ПУСК

Упорядоченный список

Номер	Иден-р
1	adhadjmah
2	agadgdgh
3	agrjgijxbkj
4	ahxth
5	bjzobhob
6	bvnfn
7	bzxvbh
8	dfgdjuhn
9	dfghiprig
10	dgh

Рехэширование	
Кол-во коллизий	28
Кол-во сравнений	3
Среднее кол-во сравнений	2,06944444444444

Упорядоченный список	
Кол-во сравнений	6
Среднее кол-во сравнений	7,18982455888002

sgha

НАЙТИ

Повторы

Удален повтор керкер

Результат поиска по рехэшированию

Иден-р sgha найден!
Его номер 145

Результат поиска по списку

Иден-р sgha найден!
Его номер 41

ОЧИСТИТЬ

Выход

- Лексический анализатор – это часть компилятора, которая читает литеры программы на исходном языке и строит из них слова (лексемы) исходного языка.
 - Лексема (лексическая единица языка) – это структурная единица языка, которая состоит из элементарных символов языка и не содержит в своем составе других структурных единиц языка.
- ✓ *Результатом работы лексического анализатора является перечень всех найденных в тексте исходной программы лексем. Этот перечень представляется в виде таблицы, называемой таблицей лексем*

Результат работы лексического анализатора

Form1

1 часть курсовой | 2 часть курсовой | 3 часть курсовой

Загрузить

```
prog
do
if ( vxb > 0 ) and ( a = b ) then begin
  c := nb / 0hAB165 + 4 ;
  d := 45 * fg - 54 ;
end
else c := 98 endif
while ( df > 0 ) ;
{dsgns}
end.
```

ТИ

ХФ	Иден-р
93	
94	
95	
96	
97	a
98	b
99	c
100	d
101	
102	

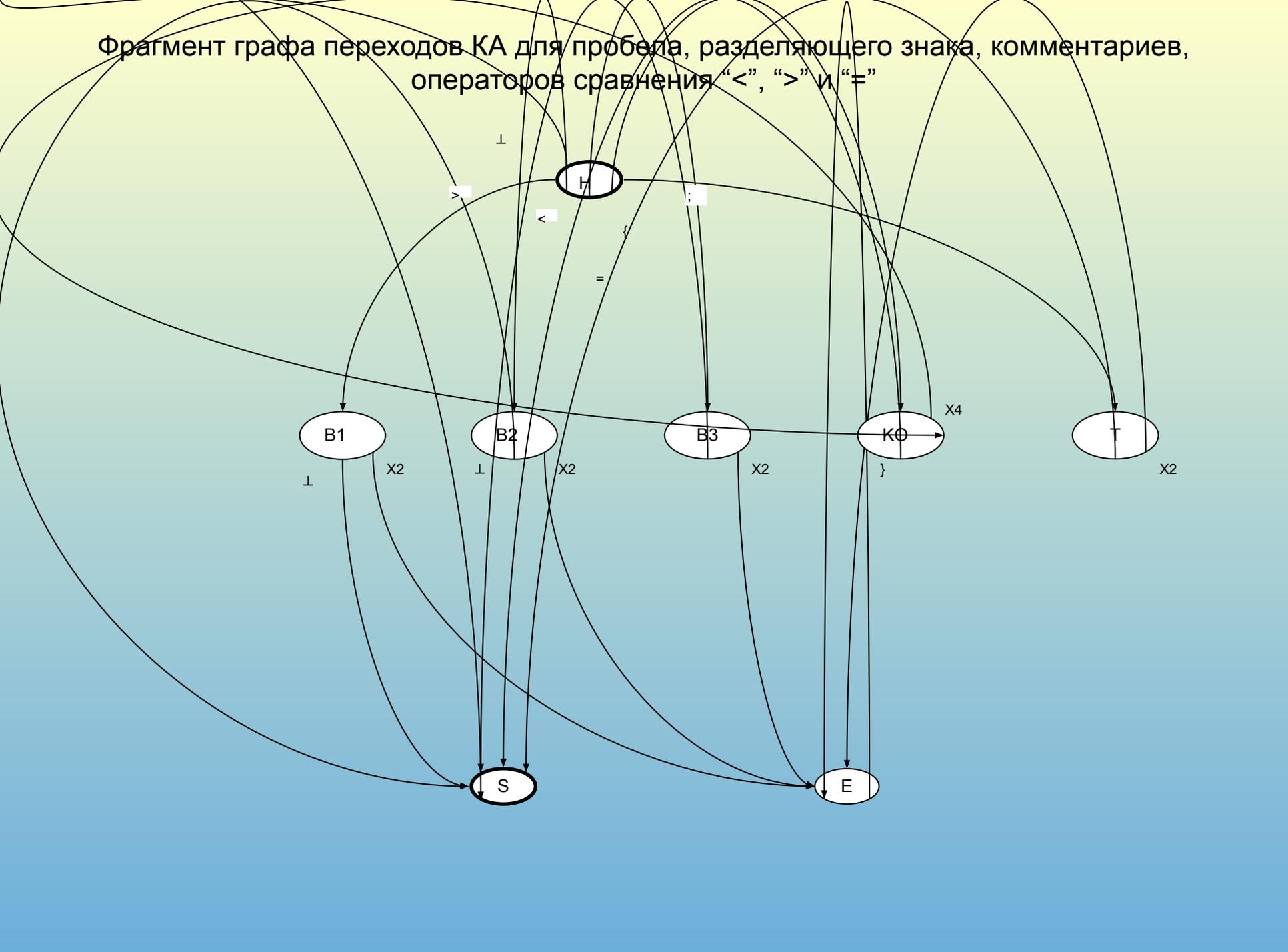
Таблица лексем

Номер	Значение	Лексема
1	prog	Ключевое слово
2	do	Ключевое слово
3	if	Ключевое слово
4	(Открывающаяся скобка
5	vxb	Идентификатор
6	>	Оператор сравнения "больше"
7	0	Целое число
8)	Закрывающаяся скобка
9	and	Логическое "и"
10	(Открывающаяся скобка
11	a	Идентификатор
12	=	Оператор сравнения "равно"
13	b	Идентификатор
14)	Закрывающаяся скобка
15	then	Ключевое слово
16	begin	Ключевое слово
17	c	Идентификатор
18	:=	Оператор присваивания
19	nb	Идентификатор
20	/	Арифметическая операция
21	0hAB165	Шестнадцатиричная константа
22	+	Арифметическая операция
23	4	Целое число

Ошибки

Выход

Фрагмент графа переходов КА для пробела, разделяющего знака, комментариев,
операторов сравнения "<", ">" и "="



Ошибка при не закрытом комментарии!!!

Form1

1 часть курсовой | 2 часть курсовой | 3 часть курсовой

Загрузить

```
prog
do
if ( vxb := 0 ) and ( a = b ) then begin
c := nb - 0hAB165 / 4 ;
d := 45 * fg - 54 ;
end
else c := 98 endif
while ( df > 0 ) ;
{dsgns
end.
```

ТИ

ХФ	Иден-р
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Таблица лексем

Номер	Значение	Лексема
1	prog	Ключевое слово
2	do	Ключевое слово
3	if	Ключевое слово
4	(Открывающаяся скобка
5	vxb	Идентификатор
6	:=	Оператор присваивания
7	0	Целое число
8)	Закрывающаяся скобка
9	and	Логическое "и"
10	(Открывающаяся скобка
11	a	Идентификатор
12	=	Оператор сравнения "равно"
13	b	Идентификатор
14)	Закрывающаяся скобка
15	then	Ключевое слово
16	begin	Ключевое слово
17	c	Идентификатор
18	:=	Оператор присваивания
19	nb	Идентификатор
20	-	Арифметическая операция
21	0hAB165	Шестнадцатиричная константа
22	/	Арифметическая операция
23	4	Целое число

Project1

Не закрыт комментарий!!!

OK

Ошибки

Выход

Входной язык задан с помощью следующей КС-грамматики:

$G(\{\text{prog, end., if, then, else, endif, begin, end, while, do, and, or, not, =, <, >, (,), -, +, a, ;, :=}\},$
 $\{S, L, O, B, C, D, H, E, T\}, P, S)$

с правилами P:

$S \rightarrow \text{prog } L \text{ end.}$

$L \rightarrow O \mid L ; O \mid L ;$

$O \rightarrow \text{if } B \text{ then } O \text{ else } O \text{ endif} \mid \text{if } B \text{ then } O \text{ endif} \mid \text{begin } L \text{ end} \mid$

$\text{do } O \text{ while } (B) \mid a := E$

$B \rightarrow B \text{ or } C \mid C$

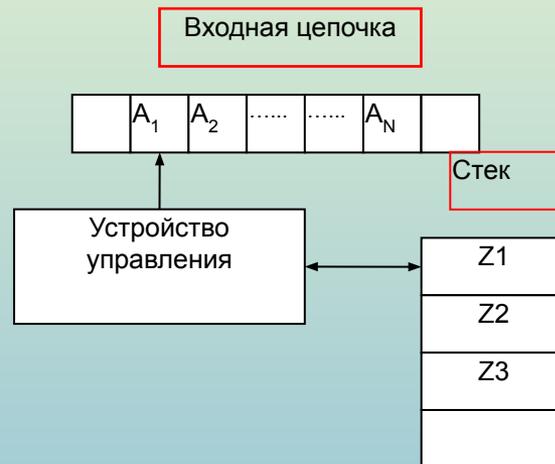
$C \rightarrow C \text{ and } D \mid D$

$D \rightarrow E < E \mid E > E \mid E = E \mid (B) \mid \text{not } (B)$

$E \rightarrow E - F \mid E + F \mid E * F \mid E / F \mid E$

$F \rightarrow (E) \mid a$

Класс КС-языков допускает распознавание с помощью недетерминированного конечного автомата со стековой (или магазинной) памятью – МП – автомата.



МП-автоматом выполняется алгоритм «сдвиг-свертка» для грамматики операторного предшествования. Для моделирования его работы необходима входная цепочка символов и стек, в котором автомат может обращаться не только к самому верхнему символу, но и к некоторой цепочке символов на вершине стека.

- В курсовом проекте КС-грамматика является грамматикой операторного предшествования. Для построения анализатора на основе этой грамматики, необходимо построить матрицу операторного предшествования. Для этого на первом шаге нужно получить множество крайних левых и крайних правых символов из правил грамматики G .

Множество крайних левых и крайних правых СИМВОЛОВ

Символы U	$L(U)$	$R(U)$
F	(, a), a
E	E, F, (, a	F,), a
D	(, not, E, F, a	E, F,), a
C	C, D, (, not, E, F, a	D, E, F,), a
B	B, C, D, (, not, E, F, a	C, D, E, F,), a
O	if, begin, do, a	endif, end, E,), F, a
L	O, L, if, begin, do, a	O, ;, endif, end, E,), F, a
S	prog	end.

Множество крайних левых и крайних правых терминальных символов.

Символы U	$L_t(U)$	$R_t(U)$
F	(, a), a
E	-, +, *, /, (, a	-, +, *, /,), a
D	<, >, =, (, not, -, +, *, /, a	<, >, =,), -, +, *, /, a
C	and, <, >, =, (, not, -, +, *, /, a	and, <, >, =,), -, +, *, /, a
B	or, and, <, >, =, (, not, -, +, *, /, a	or, and, <, >, =,), -, +, *, /, a
O	if, begin, do, a	endif, end,), :=, -, +, *, /, a
L	;; if, begin, do, a	;; endif, end,), :=, -, +, *, /, a
S	prog	end.
F	(, a), a

Результаты работы синтаксического анализатора

Form1

1 часть курсовой | 2 часть курсовой | 3 часть курсовой

Цепочка правил

1 4 8 17 26 26 5 9 26 7 4 3 9 20 26 22 26 26 9 23 26 20 26 26 12 25 18 26 26 25 14 25 17 26 26

Вывести дерево вывода

```
graph TD
    E1[E] --- prog
    E1 --- E2[E]
    E2 --- do
    do --- E3[E]
    E3 --- if
    if --- E4[E]
    E4 --- E5[E]
    E5 --- LP1["("]
    E5 --- E6[E]
    E5 --- RP1[")"]
    E6 --- not
    not --- E7[E]
    E7 --- LP2["("]
    E7 --- E8[E]
    E7 --- RP2[")"]
    E8 --- E9[E]
    E8 --- GT[">"]
    E8 --- E10[E]
    E9 --- a1[a]
    E10 --- a2[a]
    E4 --- RP3[")"]
    E4 --- and
    and --- E11[E]
    E11 --- LP3["("]
    E11 --- E12[E]
    E11 --- RP4[")"]
    E12 --- E13[E]
    E12 --- EQ["="]
    E12 --- E14[E]
    E13 --- a3[a]
    E14 --- a4[a]
    E4 --- then
```

Вывод

Цепочка правил

Вывести дерево вывода

Project1

Ошибка! Символы begin и (не могут следовать друг за другом

OK

Выход

Выводы

Проведено сравнение двух методов:

1. Хеш-адресация с рехешированием с помощью произведения;
2. Упорядоченный список.

Построен лексический анализатор, который формирует таблицу лексем и таблицу идентификаторов.

Построен синтаксический анализатор, который строит дерево разбора.