

Таблица лексем языка Go

Название	№	Запись в языке	Индекс
function	1	func	0
return	2	return	0
if	3	if	0
else	4	else	0
for	5	for	0
switch	6	switch	0
case	7	case	0
break	8	break	0
default	9	default	0
var	10	var	0
bool	11	bool	0
int	12	int	0
float	13	float	0
string	14	string	0
print	15	fmt.Print	0
scan	16	fmt.Scan	0
pow	17	fmt.Pow	0
sqrt	18	fmt.Sqrt	0
фигурные скобки	30	{,}	k({) = 0 k(}) = 1
знак отношения	31	==, <, >, <=, >=	k(==) = 0 k(<) = 1 k(>) = 2 k(<=) = 3 k(>=) = 4
±*/%	32	+, −, *, /, %	k(+) = 0 k(−) = 1 k(*) = 2 k(/) = 4 k(%) = 5
логические	33	!, &&,	k(!) = 0 k(&&) = 1 k() = 2
квадрат. скобки	34	[,]	k([) = 0 k(]) = 1
круг. скобки	35	(,)	k() = 0 k()) = 1
присвоение	36	=	0
запятая	38	,	0
идентификатор	40	length, a3, _45x	индекс в таблице идентификаторов
целая константа	50	132, 45	индекс в таблице целых констант

строковая константа	51	“Hello”, “Привет”	индекс в таблице строковых констант
---------------------	----	-------------------	-------------------------------------

Таблица идентификаторов

Индекс (указатель)	Имя
1	length
2	a3
3	_45x

Таблица целых констант

Индекс (указатель)	Значение
1	132
2	45

Таблица строковых констант

Индекс (указатель)	Значение
1	“Hello”
2	“Привет”

Грамматика языка Go

<программа>	::=	<блок функций><тело главной программы><блок функций>
<объявление переменных>	::=	var <список переменных>
< список переменных >	::=	<список имен><тип>
<список имен>	::=	ид. ид. , <список имен>
<тип>	::=	int uint float bool string
<тело программы>	::=	func main { <последовательность операторов> }
<посл. операторов>	::=	<оператор> <оператор> <посл. операторов>
<оператор>	::=	ид. := <выражение> if <условие> { < посл. операторов > } if <условие> { < посл. операторов > } else { < посл. операторов > } for <условие> { < посл. операторов > } for ид. = <выражение> ; <условие> ; <оператор>{ < посл. операторов > } fmt.Print (<выражение>) fmt.Read (<выражение>) math.Pow (<выражение>) math.Sqrt (<выражение>) switch ид. { case <выражение>: <последовательность операторов> break default: <последовательность операторов> } { <последовательность операторов> }

<блок функций>	::=	<функция> <функция> <блок функций>
<функция>	::=	function ид.(<входные параметры>) <тип>{ <последовательность операторов> } function ид.(<входные параметры>) <тип>{ <последовательность операторов> return <выражение> } }
<входные параметры>	::=	ид. <тип> ид. <тип>, <входные параметры>
<условие>	::=	!<условие> <сравнение> <сравнение> <логический оператор> <условие>
<сравнение> <лог. оператор>	::= ::=	<выражение> отн. <выражение> &&
<выражение>	::=	<численное выражение> <строковое выражение>
<числ. выражение>	::=	<терм> <терм> \pm */% <числ. выражение>
<терм>	::=	ид. (<числ. выражение>) цел. конст
<стр. выражение>	::=	<стр. терм> <стр. терм> + <стр. выражение>
<стр. терм>	::=	ид. с. конст

Таблица лексем Pascal

Название	№	Запись в языке	Индекс
procedure	1	procedure	0
function	2	function	0
begin	3	begin	0
end	4	end	0
if	5	if	0
then	6	then	0
else	7	else	0
while	8	while	0
do	9	do	0
for	10	for	0
to	11	to	0
read	12	read	0
write	13	write	0
var	14	var	0
and	15	and	0
or	16	or	0
integer	17	integer	0
real	18	real	0
boolean	19	boolean	0
string	20	string	0
pow	21	pow	0
sqrt	22	sqrt	0
точка с запятой	30	;	0
знак отношения	31	=, <, >, <=, >=	k(=) = 0 k(<) = 1 k(>) = 2 k(≤) = 3 k(≥) = 4
±	32	+, −, *, /, %	k(+) = 0 k(−) = 1 k(*) = 2 k(/) = 4 k(%) = 5
присваивание	34	:=	0
квадрат. скобки	35	[,]	k([) = 0 k(]) = 1
круг. скобки	36	(,)	k(() = 0 k()) = 1
двоеточие	37	:	0
точка	38	.	0
запятая	39	,	0
идентификатор	40	length, a3, _45x	индекс в таблице идентификаторов

целая константа	50	132, 45	индекс в таблице целых констант
строковая константа	51	'Hello', 'Привет'	индекс в таблице строковых констант

Таблица идентификаторов

Индекс (указатель)	Имя
1	length
2	a3
3	45x

Таблица целых констант

Индекс (указатель)	Значение
1	132
2	45

Таблица строковых констант

Индекс (указатель)	Значение
1	'Hello'
2	'Привет'

Грамматика языка Pascal

<программа>	::=	<блок функций><объявление переменных> <тело программы>
<объявление переменных> <список переменных> <блок переменных> <список имен> <тип>	::= ::= ::= ::= ::=	var <список переменных> <блок переменных>;<список переменных> <блок переменных>; <список имен> :<тип> ид. ид. , <список имен> real integer boolean string
<тело программы> <посл. операторов> <оператор>	::= ::= ::=	begin <последовательность операторов> end. <оператор> <оператор> ; <оператор> ; <посл. операторов> ид. := <выражение> if <условие> then <оператор> if <условие> then <оператор> else <оператор> while <условие> do <оператор> while <условие> do begin <посл. операторов> ид. := ид. +1; end write (<выражение>) read (<выражение>) begin <последовательность операторов> end
<блок функций> <функция>	::= ::=	<функция> <функция> <блок функций> procedure ид. (<входные параметры>) ; <объявление переменных> begin <последовательность операторов> end ; function ид. (<входные параметры>) :<тип>; <объявление переменных> begin <последовательность операторов> Ид. :=<выражение>; end ;

<входные параметры>	::=	ид.: <тип> ид.: <тип>, <входные параметры>
<условие>	::=	<сравнение> <сравнение> <логический оператор> <условие>
<сравнение>	::=	<выражение> отн. <выражение>
<лог. оператор>	::=	and or
<выражение>	::=	<численное выражение> <строковое выражение>
<числ. выражение>	::=	<терм> <терм> ±*/% <числ. выражение>
<терм>	::=	ид. (<числ. выражение>) цел. конст
<стр. выражение>	::=	<стр. терм> <стр. терм> + <стр. выражение>
<стр. терм>	::=	ид. с.конст