SYMBOL TABLE for test.lusd

Total Tokenized Lexemes :267
Elapsed Time :0.0029916763305664062

LINE	LEXEME	TOKEN
1	import	KEYWORD_IMPORT
1	TestLibrary	IDENTIFIER
1	;	STMT_TERMINATOR
3	public	KEYWORD_PUBLIC
3	class	KEYWORD_CLASS
3	testClass	IDENTIFIER
3	:	CODEBLK_INDICATOR
4	{	OPEN_CURLY_BRACKET
5	init	KEYWORD_INIT
5	(	OPEN_PARENTHESIS
5	this	KEYWORD_THIS
5	,	SEPARATOR
5	int	KEYWORD_INT
5	num	IDENTIFIER
5	)	CLOSE_PARENTHESIS
5	:	CODEBLK_INDICATOR
6	{	OPEN_CURLY_BRACKET
7	this	KEYWORD_THIS
7	.	OBJECT_DELIMITER
7	property_2_1	IDENTIFIER
7	=	OP_ASSIGNMENT
7	num	IDENTIFIER
7	;	STMT_TERMINATOR
8	display	KEYWORD_DISPLAY
8	(	OPEN_PARENTHESIS
8	num	IDENTIFIER
8	)	CLOSE_PARENTHESIS

LINE	LEXEME	TOKEN
8	;	STMT_TERMINATOR
9	}	CLOSE_CURLY_BRACKET
11	public	KEYWORD_PUBLIC
11	func	KEYWORD_FUNC
11	TestMethod	IDENTIFIER
11	(	OPEN_PARENTHESIS
11	)	CLOSE_PARENTHESIS
11	:	CODEBLK_INDICATOR
12	{	OPEN_CURLY_BRACKET
13	display	KEYWORD_DISPLAY
13	(	OPEN_PARENTHESIS
13	"Hi Word"	STRING_LITERAL
13	)	CLOSE_PARENTHESIS
13	;	STMT_TERMINATOR
14	}	CLOSE_CURLY_BRACKET
16	public	KEYWORD_PUBLIC
16	func	KEYWORD_FUNC
16	TestMethod3	IDENTIFIER
16	(	OPEN_PARENTHESIS
16	str	KEYWORD_STR
16	val1	IDENTIFIER
16	)	CLOSE_PARENTHESIS
16	:	CODEBLK_INDICATOR
17	{	OPEN_CURLY_BRACKET
18	display	KEYWORD_DISPLAY
18	(	OPEN_PARENTHESIS
18	val1	IDENTIFIER
18	)	CLOSE_PARENTHESIS
18	;	STMT_TERMINATOR
18	display	KEYWORD_DISPLAY

LINE	LEXEME	TOKEN
18	(	OPEN_PARENTHESIS
18	val1	IDENTIFIER
18	)	CLOSE_PARENTHESIS
18	;	STMT_TERMINATOR
19	}	CLOSE_CURLY_BRACKET
20	}	CLOSE_CURLY_BRACKET
22	private	KEYWORD_PRIVATE
22	class	KEYWORD_CLASS
22	testClass2	IDENTIFIER
22	:	CODEBLK_INDICATOR
23	{	OPEN_CURLY_BRACKET
24	initialize	KEYWORD_INIT
24	(	OPEN_PARENTHESIS
24	this	KEYWORD_THIS
24	,	SEPARATOR
24	int	KEYWORD_INT
24	num	IDENTIFIER
24	)	CLOSE_PARENTHESIS
24	:	CODEBLK_INDICATOR
25	{	OPEN_CURLY_BRACKET
26	display	KEYWORD_DISPLAY
26	(	OPEN_PARENTHESIS
26	num	IDENTIFIER
26	/	OP_DIVISION
26	2	INT_LITERAL
26	)	CLOSE_PARENTHESIS
26	;	STMT_TERMINATOR
27	}	CLOSE_CURLY_BRACKET
28	}	CLOSE_CURLY_BRACKET
30	main	KEYWORD_MAIN

LINE	LEXEME	TOKEN
30	(	OPEN_PARENTHESIS
30	)	CLOSE_PARENTHESIS
30	{	OPEN_CURLY_BRACKET
31	info	KEYWORD_INFO
31	(	OPEN_PARENTHESIS
31	)	CLOSE_PARENTHESIS
31	;	STMT_TERMINATOR
32	information	KEYWORD_INFO
32	(	OPEN_PARENTHESIS
32	)	CLOSE_PARENTHESIS
32	;	STMT_TERMINATOR
34	int	KEYWORD_INT
34	intvar_fiRst	IDENTIFIER
34	=	OP_ASSIGNMENT
34	3	INT_LITERAL
34	;	STMT_TERMINATOR
35	integer	KEYWORD_INT
35	intvar	IDENTIFIER
35	=	OP_ASSIGNMENT
35	3	INT_LITERAL
35	+	OP_ADDITION
35	4	INT_LITERAL
35	;	STMT_TERMINATOR
37	private	KEYWORD_PRIVATE
37	obj	KEYWORD_OBJ
37	SecretClass	IDENTIFIER
37	=	OP_ASSIGNMENT
37	testClass	IDENTIFIER
37	(	OPEN_PARENTHESIS
37	3	INT_LITERAL

LINE	LEXEME	TOKEN
37	)	CLOSE_PARENTHESIS
37	;	STMT_TERMINATOR
38	public	KEYWORD_PUBLIC
38	object	KEYWORD_OBJ
38	object2	IDENTIFIER
38	=	OP_ASSIGNMENT
38	testClass2	IDENTIFIER
38	(	OPEN_PARENTHESIS
38	5	INT_LITERAL
38	)	CLOSE_PARENTHESIS
38	;	STMT_TERMINATOR
40	display	KEYWORD_DISPLAY
40	(	OPEN_PARENTHESIS
40	"Hello Word!\b"	STRING_LITERAL
40	)	CLOSE_PARENTHESIS
40	;	STMT_TERMINATOR
42	float	KEYWORD_FLOAT
42	FloatVar	IDENTIFIER
42	=	OP_ASSIGNMENT
42	420.0f	FLOAT_LITERAL
42	;	STMT_TERMINATOR
43	double	KEYWORD_DOUBLE
43	double_var	IDENTIFIER
43	=	OP_ASSIGNMENT
43	0.69	DOUBLE_LITERAL
43	;	STMT_TERMINATOR
45	char	KEYWORD_CHAR
45	char_Va_r	IDENTIFIER
45	=	OP_ASSIGNMENT
45	'd'	CHAR_LITERAL

LINE	LEXEME	TOKEN
45	;	STMT_TERMINATOR
46	str	KEYWORD_STR
46	stringVariable	IDENTIFIER
46	=	OP_ASSIGNMENT
46	"Nico Nico niiiiii\n"	STRING_LITERAL
46	;	STMT_TERMINATOR
47	list	KEYWORD_LIST
47	Ex_Taylor	IDENTIFIER
47	[	OPEN_SQUARE_BRACKET
47	3	INT_LITERAL
47	]	CLOSE_SQUARE_BRACKET
47	=	OP_ASSIGNMENT
47	[	OPEN_SQUARE_BRACKET
47	"Loki"	STRING_LITERAL
47	,	SEPARATOR
47	"What happen bella"	STRING_LITERAL
47	,	SEPARATOR
47	"Red Scarf"	STRING_LITERAL
47	]	CLOSE_SQUARE_BRACKET
47	;	STMT_TERMINATOR
49	str	KEYWORD_STR
49	9d0fvm_3d	UNKNOWN_TOKEN
49	=	OP_ASSIGNMENT
49	"Test"	STRING_LITERAL
49	;	STMT_TERMINATOR
50	char	KEYWORD_CHAR
50	__Dienamayt	UNKNOWN_TOKEN
50	=	OP_ASSIGNMENT
50	'dc'	UNKNOWN_TOKEN
50	;	STMT_TERMINATOR

LINE	LEXEME	TOKEN
51	str	KEYWORD_STR
51	Inabikari_sunawachi_eien_nari_	UNKNOWN_TOKEN
51	;	STMT_TERMINATOR
53	str	KEYWORD_STR
53	order_var	IDENTIFIER
53	=	OP_ASSIGNMENT
53	"Oratrice Mecanique d'Analyse Cardinale;	UNKNOWN_TOKEN
55	if	KEYWORD_IF
55	(	OPEN_PARENTHESIS
55	intvar	IDENTIFIER
55	<=	LESS_OR_EQUAL_OP
55	3	INT_LITERAL
55	and	KEYWORD_AND
55	not	KEYWORD_NOT
55	5	INT_LITERAL
55	)	CLOSE_PARENTHESIS
55	then	KEYWORD_THEN
55	:	CODEBLK_INDICATOR
56	{	OPEN_CURLY_BRACKET
57	intvar	IDENTIFIER
57	=	OP_ASSIGNMENT
57	5	INT_LITERAL
57	+	OP_ADDITION
57	3	INT_LITERAL
57	+	OP_ADDITION
57	9	INT_LITERAL
57	+	OP_ADDITION
57	7	INT_LITERAL
57	/	OP_DIVISION

LINE	LEXEME	TOKEN
57	2	INT_LITERAL
57	;	STMT_TERMINATOR
58	}	CLOSE_CURLY_BRACKET
59	elif	KEYWORD_ELIF
59	(	OPEN_PARENTHESIS
59	intvar	IDENTIFIER
59	!=	INEQUALITY_OP
59	3	INT_LITERAL
59	)	CLOSE_PARENTHESIS
59	:	CODEBLK_INDICATOR
60	{	OPEN_CURLY_BRACKET
61	intvar	IDENTIFIER
61	=	OP_ASSIGNMENT
61	-	OP_SUBTRACTION
61	10	INT_LITERAL
61	;	STMT_TERMINATOR
62	}	CLOSE_CURLY_BRACKET
63	else	KEYWORD_ELSE
63	:	CODEBLK_INDICATOR
64	{	OPEN_CURLY_BRACKET
65	pass	KEYWORD_PASS
65	;	STMT_TERMINATOR
66	}	CLOSE_CURLY_BRACKET
68	for	KEYWORD_FOR
68	(	OPEN_PARENTHESIS
68	int	KEYWORD_INT
68	i	IDENTIFIER
68	=	OP_ASSIGNMENT
68	0	INT_LITERAL
68	;	STMT_TERMINATOR

LINE	LEXEME	TOKEN
68	i	IDENTIFIER
68	<	LESS_THAN_OP
68	5	INT_LITERAL
68	;	STMT_TERMINATOR
68	i	IDENTIFIER
68	++	OP_INCREMENT
68	)	CLOSE_PARENTHESIS
68	:	CODEBLK_INDICATOR
69	{	OPEN_CURLY_BRACKET
70	display	KEYWORD_DISPLAY
70	(	OPEN_PARENTHESIS
70	testClass	IDENTIFIER
70	.	OBJECT_DELIMITER
70	property_2_1	IDENTIFIER
70	)	CLOSE_PARENTHESIS
70	;	STMT_TERMINATOR
71	}	CLOSE_CURLY_BRACKET
73	while	KEYWORD_WHILE
73	(	OPEN_PARENTHESIS
73	int	KEYWORD_INT
73	j	IDENTIFIER
73	<	LESS_THAN_OP
73	5	INT_LITERAL
73	)	CLOSE_PARENTHESIS
73	:	CODEBLK_INDICATOR
74	{	OPEN_CURLY_BRACKET
75	j	IDENTIFIER
75	++	OP_INCREMENT
75	;	STMT_TERMINATOR
76	}	CLOSE_CURLY_BRACKET

LINE	LEXEME	TOKEN
77	}	CLOSE_CURLY_BRACKET