Duy Tran

1002190361

Unit tests

main			
#	**test paths**	**input**	**output**
1	[1, 2, 6, 7, 8, 9]	no aruments	no ouput
2	[1, 3, 5, 6, 7, 8, 9, 10, 11, 9]	main2.txt file contains:xor (a1 123 "asd" #a ;comment 123a	keyword,"xor". lparen. identifier,"a1". numeric,123. string,"asd". character,"a". comment,";comment ". error,"123a".
3	[1, 3, 4, 6, 7, 8, 9]	main2.txt, fakemain.txt	Error! Please give the token stream

is_spec_symbol			
#	**test paths**	**input**	**output**
1	[1, 2]	(	TRUE
2	[1, 3, 4]	)	TRUE
3	[1, 3, 5, 6]	[	TRUE
4	[1, 3, 5, 7, 8]	]	TRUE
5 **line 445: is true**	[1, 3, 5, 7, 9, 10]	/	FALSE
6	[1, 3, 5, 7, 9, 11, 12]	`	TRUE
7	[1, 3, 5, 7, 9, 11, 13, 14]	,	TRUE
8	[1, 3, 5, 7, 9, 11, 13, 15]	!	FALSE

print_spec_symbol			
#	**test paths**	**input**	**output**
1 **line 378:) absent**	[1, 2]	)	rparen.
2	[1, 3, 4]	)	rparen.
3	[1, 3, 5, 6]	[	lsquare.
4	[1, 3, 5, 7, 8]	]	rsquare.
5 **last func didn't have '**	[1, 3, 5, 7, 9, 10]	'	quote.
6	[1, 3, 5, 7, 9, 11, 12]	`	bquote.
7	[1, 3, 5, 7, 9, 11, 13, 14]	,	comma.

is_identifier			

#	test paths	input	output
1 **line 365: true when not letter?**	[1, 7]	#ac	FALSE
2 **line 362: false when is letter?**	[1, 2, 6]	a	TRUE
3	[1, 2, 3, 5]	a#	FALSE
4	[1, 2, 3, 4, 2, 6]	aa1	TRUE

is_str_constant			
#	**test paths**	**input**	**output**
1	[1, 7]	f	FALSE
2	[1, 2, 6]	"	FALSE
3	[1, 2, 3, 5]	"f	FALSE
4	[1, 2, 3, 4, 2, 6]	"f2"	TRUE

is_num_constant			
#	**test paths**	**input**	**output**
1	[1, 7]	f	FALSE
2	[1, 2, 6]	1	TRUE
3	[1, 2, 3, 5]	1a	FALSE
4	[1, 2, 3, 4, 2, 6]	32	TRUE

is_char_constant			
#	**test paths**	**input**	**output**
1	[1, 2]	#f	TRUE
2	[1, 3]	f	FALSE

is_keyword			
#	**test paths**	**input**	**output**
1	[1, 2]	=>	TRUE
2	[1, 3]	not	FALSE

is_comment			
#	**test paths**	**input**	**output**
1	[1, 2]	;is it?	TRUE
2	[1, 3]	is it?	FALSE

print_token			

#	test paths	input	output
1	[1, 3, 5, 7, 9, 11, 13, 15]	!	error,"!".
2	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]	and	keyword,"and".

token_type			
#	**test paths**	**input**	**output**
1	[1, 2]	and	1
2	[1, 3, 4]	[	2
3	[1, 3, 5, 6]	a2	3
4	[1, 3, 5, 7, 8]	22	41
5	[1, 3, 5, 7, 9, 10]	"r"	42
6	[1, 3, 5, 7, 9, 11, 12]	#e	43
7	[1, 3, 5, 7, 9, 11, 13, 14]	;e	5
8	[1, 3, 5, 7, 9, 11, 13, 15]	#	0

is_token_end			
#	**test paths**	**input**	**output**
1	[1, 2]	0, -1	TRUE
2	[1, 3, 4, 5]	1, 34	TRUE
3	[1, 3, 4, 6]	1, 97	FALSE
4	[1, 3, 7, 8, 9]	2, 10	TRUE
5	[1, 3, 7, 8, 10]	2, 98	FALSE
6	[1, 3, 7, 11, 12]	0, 40	TRUE
7 **line 191: 59 instead of ;**	[1, 3, 7, 11, 13, 14]	0, 32	TRUE
8	[1, 3, 7, 11, 13, 15]	0, 100	FALSE

get_token			
#	**test paths**	**input**	**output**
1	[1, 2]	empty file	null
2	[1, 3, 4, 5, 4, 6, 7]	\n	null
3	[1, 3, 4, 6, 8, 9]	(	(
4	[1, 3, 4, 6, 8, 10, 11]	"	"
5	[1, 3, 4, 6, 8, 10, 12, 13, 14, 15]	;	;
6	[1, 3, 4, 6, 8, 10, 12, 14, 15]	a	a
7	[1, 3, 4, 6, 8, 10, 12, 13, 14, 16, 17, 18, 20, 21]	;word	;word
8	[1, 3, 4, 6, 8, 10, 12, 13, 14, 16, 17, 19, 16, 20, 21]	;2 more words	;2 more words

9	[1, 3, 4, 6, 8, 10, 12, 14, 16, 20, 22, 23]	as]	as
10 **line 148: id 2 is for "**	[1, 3, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 25, 26, 27]	"word"	"word"
11	[1, 3, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 25, 27]	word!	word!
12 **line 155: cant have id == 0 and ch == 59 based on line 119**	[1, 3, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 28, 29]	;word	;word
13	[1, 3, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 28, 30]	;word!	;word!

open_token_stream			
#	**test paths**	**input**	**output**
1	[1, 2, 4]	empty	
2	[1, 3]	testfile.txt	

unget_char			
#	**test paths**	**input**	**output**
1	[1, 2]		[App warning]: this function is not available!

get_char			
#	**test paths**	**input**	**output**
1	[1, 2]	a	97

open_character_stream			
#	**test paths**	**input**	**output**
1	[1, 2, 4]	null	nothing
2	[1, 3, 4]	valid file path file contains: a	a

End-to-end testing

1)

main[1, 3, 5, 6, 7->open_token_stream[1, 2->open_character_stream[1, 3, 4], 4], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 9], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 3->is_spec_symbol[1, 2], 4], 4, 5, 6->print_spec_symbol[1, 2], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11->get_token[1->get_char[1, 2], 3, 4, 5, 4, 6, 7], 9]

input: (

expected output: lparen.

2)

main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 3, 5, 7, 9, 10] , 10, 12, 13, 14->get_char[1, 2], 16->is_token_end[1, 3, 7, 11->is_spec_symbol[1, 3, 5, 7, 9, 11, 13, 14], 13, 14], 17->get_char[1, 2] , 18, 20, 21->unget_char[1, 2]], 9, 10->print_token[1->token_type[1->is_keyword[1, 3], 3->is_spec_symbol[1, 3, 5, 7, 9, 11, 13, 15], 5->is_identifier[1, 2, 3, 4, 2, 6], 7->is_num_constant[1, 2, 3, 4, 2, 6], 9->is_str_constant[1, 2, 3, 4, 2, 6], 11->is_char_constant[1, 2], 13->is_comment[1, 2], 15], 3, 5, 7, 9, 11, 13, 15], 11->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 3, 5, 7, 9, 11, 12], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 3, 7, 11->is_spec_symbol[1, 3, 5, 7, 8], 13, 15], 20, 22->is_spec_symbol[1, 3, 5, 6], 24, 25, 26, 27], 9]

input: xor] a1 123 "asd" #a ;comment 123a

expected output: keyword,"xor".

rsquare.

identifier,"a1".

numeric,123.

string,"asd".

character,"a".

comment,";comment 123a".

3)

main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 3, 4], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 3, 7, 11->is_spec_symbol[1, 3, 5, 6], 12], 20, 22->is_spec_symbol[1, 3, 5, 6], 24, 28, 29->unget_char[1, 2]], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 3->is_spec_symbol[1, 3, 5, 6], 5->is_identifier[1, 2, 3, 5], 7->is_num_constant[1, 2, 3, 5], 9->is_str_constant[1, 2, 3, 5], 11->is_char_constant[1, 3], 13->is_comment[1, 3], 14], 2, 3, 4, 5, 6->print_spec_symbol[1, 3, 5, 7, 9, 11, 13, 14], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 3, 5, 6], 10, 12, 13, 14->get_char[1, 2], 16->is_token_end[1, 3, 7, 8, 10], 17->get_char[1, 2], 19, 16, 20, 21->unget_char[1, 2]], 9]

input: ` var 456 "test"

expected output: bquote.

identifier,"var".

numeric,456.

string,"test".

4)
main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 3, 5, 7, 9, 11, 12], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 3, 7, 8, 9], 20, 22->is_spec_symbol[1, 3, 5, 7, 9, 11, 12], 24, 28, 30], 9, 10->print_token[1->token_type[1->is_keyword[1, 3], 3->is_spec_symbol[1, 3, 5, 7, 9, 11, 12], 5->is_identifier[1, 2, 6], 7->is_num_constant[1, 2, 6], 9->is_str_constant[1, 2, 3, 5], 11->is_char_constant[1, 2], 12], 3, 5, 7, 9, 11, 13, 15], 11-> get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 3, 5, 7, 9, 11, 12], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 3, 4, 6], 20, 22->is_spec_symbol[1, 3, 5, 7, 9, 11, 12], 24, 25, 27], 9]

input: ' lambda 789 #t

expected output: quote.

keyword,"lambda".

numeric,789.

character,"t".

5)
main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 2], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 3->is_spec_symbol[1, 2], 5->is_identifier[1, 7], 7->is_num_constant[1, 7], 9->is_str_constant[1, 2, 6], 10], 2, 3, 4, 5, 6->print_spec_symbol[1, 3, 5, 7, 9, 11, 12], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 3, 4, 5], 20, 22->is_spec_symbol[1, 2], 23->unget_char[1, 2]], 9]

input: empty file

expected output: no ouput

6)
main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 10, 12, 14->get_char[1, 2], 15->unget_char[1, 2]], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 3->is_spec_symbol[1, 2], 5->is_identifier[1, 7], 7->is_num_constant[1, 7], 8], 2, 3, 4, 5, 6->print_spec_symbol[1, 3, 5, 7, 9, 10], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11-> get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 10, 12, 13, 14->get_char[1, 2], 15->unget_char[1, 2]], 9]

input: af

expected output: identifier,"af".

7)
main[1, 3, 5, 6, 7->open_token_stream[1, ->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 10, 11], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 3->is_spec_symbol[1, 2], 5->is_identifier[1, 2, 6], 6], 2, 3, 4, 5, 6->print_spec_symbol[1, 3, 5, 7, 8], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11-> get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 3, 4, 5], 20, 22->is_spec_symbol[1, 2], 23->unget_char[1, 2]], 9]

input: g4

expected output: identifier,"g4".

8)
main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 3, 4, 6, 8->is_spec_symbol[1, 2], 10, 12, 14->get_char[1, 2], 16->is_token_end[1, 2], 20, 22->is_spec_symbol[1, 2], 23->unget_char[1, 2]], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 3->is_spec_symbol[1, 2], 5->is_identifier[1, 2, 6], 7->is_num_constant[1, 2, 6], 9->is_str_constant[1,7], 10], 2, 3, 4, 5, 6->print_spec_symbol[1, 3, 5, 6], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11-> get_token[1->get_char[1, 2], 2], 9]

input: “hello”

expected output: string,"hello".

9)
main[1, 3, 5, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 2], 9, 10->print_token[1->token_type[1->is_keyword[1, 2], 2], 2, 3, 4, 5, 6->print_spec_symbol[1, 3, 4], 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], 11-> get_token[1->get_char[1, 2], 2], 9]

input: ,

expected output: comma

10)

main[1, 2, 6, 7->open_token_stream[1, 3->open_character_stream[1, 2, 4]], 8->get_token[1->get_char[1, 2], 2], 9]

input: no argument

expected output: no output

11)

main[1, 3, 4, 6, 7->open_token_stream[1, 3->open_character_stream[1, 3, 4]], 8->get_token[1->get_char[1, 2], 2], 9]

input: multiple arguments

expected output: Error! Please give the token stream