Test Cases Document - Test cases, including each test path, the corresponding test data to execute the test path, and the expected output.

By: Cesar Frayre

Date: November 25, 2024

For the entirety of the document the following format shall be followed:

#. “Name of the function in printtokens.java”

#	Test Path	Test Data/Input	Expected Output
1	…	…	…

- Each test case shall begin with the “Start” node (0^{th} node) and end with the “End” node from its corresponding CFG diagram as given in the file

1. String get_token(BufferedReader br)

#	Test Path	Test Data/Input	Expected Output
1	[Start,1,2,End]	\0	Null
2	[Start,1,3,4,5,4,6,7,End]	\n\0	Null
3	[Start, 1, 3, 4, 6, 8, 9, End]	(	“(“
4	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 15, End]	\na	“a”
5	[Start, 1, 3, 4, 6, 8, 10, 12, 13, 14, 15, End]	;	“;”
6	[Start, 1, 3, 4, 6, 8, 10, 11, 12, 14, 15, End]	“	“””
7	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 18, 19, 21, 22, End]	AB	“AB”
8	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 18, 20, 17, 21, 22, End]	ABC\0	“ABC”
9	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 24, End]	ABC(	“ABC”
10	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 25, 26, End]	ABC”	“ABC”

11	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 25, 27, 28, End]	ABC;	“ABC
12	[Start, 1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 25, 27, 29, End]	“ABC”	““ABC””

2. Boolean is_token_end(int str_com_id, int res)

#	Test Path	Test Data/Input	Expected Output
1	[Start, 1, 2, End]	0, -1	True
2	[Start, 1, 3, 4, 5, End]	1, 34	True
3	[Start, 1, 3, 4, 6, End]	1, 67	false
4	[Start, 1, 3, 7, 8, 9, End]	2, 10	true
5	[Start, 1, 3, 7, 8, 10, End]	2, 68	False
6	[Start, 1, 3, 7, 11, 12, End]	0, 40	True
7	[Start, 1, 3, 7, 11, 13, 14, End]	0, 32	True
8	[Start, 1, 3, 7, 11, 13, 15, End]	0, 69	False

3. Boolean is_keyword(String str)

#	Test Path	Test Data/Input	Expected Output
1	[Start, 1, 2, End]	“and”	True
2	[Start, 1, 3, End]	“forgive”	False

4. Boolean is_num_constant(String str)

#	Test Path	Test Data/Input	Expected Output
1	[Start, 1, 7, End]	“A”	False
2	[Start, 1, 2, 3, 5, End]	“1A”	False
3	[Start, 1, 2, 3, 4, 2, 6, End]	“10”	True

5. Boolean is_str_constant(String str)

#	Test Path	Test Data/Input	Expected Output
1	[Start, 1, 7, End]	“1”	False
2	[Start, 1, 2, 3, 4, End]	“””””	True

3	[Start, 1, 2, 3, 5, 2, 6, End]	“A”	False

6. Boolean is_identifier(String str)

#	Test Path	Test Data/Input	Expected Output
1	[Start, 1, 7, End]	“1”	True
2	[Start, 1, 2, 3, 5, End]	“a!”	False
3	[Start, 1, 2, 3, 4, 2, 6, End]	“a1”	True

7. Void main(String[] args)

#	Test Path	Test Data/Input
1	[Start, 1, 3, 5, End]	Args.length > 1
2	[Start, 1, 3, 4, 6, 7, 8, 11, End]	<reading empty file>
3	[Start, 1, 2, 6, 7, get_token.2, 8, 9, print_token.1, 10, get_token.1, 8, 11, End]	\n\0
4	[Start, 1, 2, 6, 7, get_token.3, 8, 9, print_token.3, 10, get_token.1, 8, 11, End]	(
5	[Start, 1, 2, 6, 7, get_token.4, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	/na
6	[Start, 1, 2, 6, 7, get_token.5, 8, 9, print_token.8, 10, get_token.1, 8, 11, End]	;
7	[Start, 1, 2, 6, 7, get_token.6, 8, 9, print_token.1, 10, get_token.1, 8, 11, End]	“
8	[Start, 1, 2, 6, 7, get_token.7, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	AB
9	[Start, 1, 2, 6, 7, get_token.8, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	ABC\0
10	[Start, 1, 2, 6, 7, get_token.9, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	ABC(
11	[Start, 1, 2, 6, 7, get_token.10, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	ABC”

12	[Start, 1, 2, 6, 7, get_token.11, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	ABC;
13	[Start, 1, 2, 6, 7, get_token.12, 8, 9, print_token.4, 10, get_token.1, 8, 11, End]	ABC
14	[Start, 1, 2, 6, 7, get_token.12, 8, 9, print_token.2, 10, get_token.1, 8, 11, End]	And
15	[Start, 1, 2, 6, 7, get_token.12, 8, 9, print_token.5, 10, get_token.1, 8, 11, End]	5
16	[Start, 1, 2, 6, 7, get_token.12, 8, 9, print_token.6, 10, get_token.1, 8, 11, End]	#c
17	[Start, 1, 2, 6, 7, get_token.12, 8, 9, print_token.7, 10, get_token.1, 8, 11, End]	“hello”