Faris Soepangat
Raziuddin Mohammad

Test Cases

This document consists of test cases, including each test path, the corresponding test data to execute the test path, and the expected output

Unit Testing: For each method other than the main method, select a set of test paths to achieve edge node coverage, i.e., cover all the edges in the CFG of the method

Program Testing: For the main method, select a set of test paths to achieve edge coverage for the entire program, i.e., cover all the edges in all the CFGs

1. open_character_stream
[1, 2, 3, 7]
    Input: 'null'
    Output: A 'BufferedReader' object that reads from 'System.in'

[1, 2, 4, 5, 6, 7]
    Input: '"test.txt"'
    Output: A 'BufferedReader' object that reads from 'test.txt'

2. get_char
[1, 2, 3, 4]
    Input: A 'BufferedReader' containing the string "hello"
    Output: The ASCII value of 'h' (104)

3. unget_char
[1, 2, 3]
    Input: any valid integer
    Output: 0

4. open_token_stream
[1, 2, 4]
    Input: ""
    Output: The method should call 'open_character_stream(null)'

[1, 3, 4]
    Input: test.txt
    Output: The method should call 'open_character_stream("test.txt")

5. get_token
[1, 2]

Input: \0
Output: null

[1, 3, 4, 5, 4, 6, 7]
Input: \n\0
Output: null

[1, 3, 4, 6, 8, 9]
Input: (
Output: “(“

[1, 3, 4, 6, 8, 10, 12, 14, 15]
Input: \na
Output: “a”

[1, 3, 4, 6, 8, 10, 12, 13, 14, 15]
Input: ;
Output: “;”

[1, 3, 4, 6, 8, 10, 11, 12, 14, 15]
Input: “
Output: “””

[1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 18, 19, 21, 22]
Input: AB
Output: “AB”

[1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 18, 20, 17, 21, 22]
Input: ABC\0
Output: “ABC”

[1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 24]
Input: ABC(
Output: “ABC”

[1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 25, 26]
Input: ABC”
Output: “ABC”

[1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 25, 27, 28]
Input: ABC;
Output: “ABC”

[1, 3, 4, 6, 8, 10, 12, 14, 16, 17, 21, 23, 25, 27, 29]

Input: “ABC”
Output: “”ABC””

6. is_token_end
[1, 2]
Input: 0, -1
Output: true

[1, 3, 4, 5]
Input: 1, 34
Output: true

[1, 3, 4, 6]
Input: 1, 67
Output: false

[1, 3, 7, 8, 9]
Input: 2, 10
Output: true

7. token_type
[1, 2]
Input: “and”
Output: ‘keyword’

[1, 3, 4]
Input: “(“
Output: ‘spec_symbol’

[1, 3, 5, 6]
Input: “Hi”
Output: ‘identifier’

[1, 3, 5, 7, 8]
Input: “5”
Output: ‘num_constant’

[1, 3, 5, 7, 9, 10]
Input: “\”Hello\””
Output: ‘str_constant’

[1, 3, 5, 7, 9, 10, 11, 12]
Input: “#sum”

Output: ‘char_constant’

[1, 3, 5, 7, 9, 10, 11, 13, 14]
Input: “;note”
Output: ‘comment’

[1, 3, 5, 7, 9, 10, 11, 13, 15]
Input: “#\”hello\””
Output: ‘error’

8. print_token
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
Input: print_token(“error_token”);
Output: error,”error_token”

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(“keyword_token”);
Output: keyword,”keyword_token”

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(“identifier_token”);
Output: identifier,”identifier_token”

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(“12345”);
Output: numeric,12345

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(“\”hello\””);
Output: string,”hello”

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(“\’c’”);
Output: character,”c”

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(“/* comment */”);
Output: comment,”/* comment */”

[1, 3, 5, 7, 9, 11, 13, 15]
Input: print_token(”;”);
Output: special sumbol,;,

9. is_comment

[1, 2]

Input: “;This is a comment”
Output: true

[1, 3]

Input: “a regular string”
Output: false

10. is_keyword

[1, 2]

Input: and
Output: true

[1, 3]

Input: test
Output: false

11. is_char_constant

[1, 2]

Input: “abc”
Output: true

[1, 3]

Input: “#a”
Output: true

12. is_num_constant

[1, 7]

Input: A
Output: false

[1, 2, 3, 5]

Input: 1A
Output: false

[1, 2, 3, 4, 2, 6]

Input: 10
Output: true

13. is_str_constant

[1, 7]

Input: 1
Output: false

[1, 2, 3, 4]
Input: “”
Output: true

[1, 2, 3, 5, 2, 6]
Input: “A
Output: false

14. is_identifier
[1, 7]
Input: 1
Output: false

[1, 2, 3, 5]
Input: a!
Output: false

[1, 2, 3, 4, 2, 6]
Input: a1
Output: true

15. print_spec_symbol
[1, 2]
Input: “(“
Output: lparen

[1, 3, 4]
Input: “)”
Output: rparen

[1, 3, 5, 6]
Input: “[“
Output: lsquare

[1, 3, 5, 7, 8]
Input: “]”
Output: rsquare

[1, 3, 5, 7, 9, 10]
Input: “”
Output: quote

[1, 3, 5, 7, 9, 11, 12]
Input: “’”

Output: bquote

[1, 3, 5, 7, 9, 11, 13, 14]
Input: “,”
Output: comma

16. is_spec_symbol
[1, 2]
Input: '('
Output: true

[1, 3, 4]
Input: ')'
Output: true

[1, 3, 5, 6]
Input: '['
Output: true

[1, 3, 5, 7, 8]
Input: ']'
Output: true

[1, 3, 5, 7, 9, 10]
Input: '/'
Output: true

[1, 3, 5, 7, 9, 11, 12]
Input: '` '
Output: true

[1, 3, 5, 7, 9, 11, 13, 14]
Input: ','
Output: true

[1, 3, 5, 7, 9, 11, 13, 15] (Default (c is not one of the specified characters))
Input: Any character that is not '(', ')', '[', ']', '/', '` ', or ','
Output: false

17. main
[1, 2, 6, 7, 8, 9, 10]
Input: String[] args = {};
Output: ‘fname’ is set to ‘””’

[1, 2, 6, 7, 8, 11]

Input: String[] args = {};

Output: ‘fname’ is set to ‘””’

[1, 3, 4, 6, 7, 8, 9, 10]

Input: String[] args = {“text.txt”};

Output: ‘fname’ is set to ‘”test.txt”’

[1, 3, 4, 10]

Input: String[] args = {“file1.txt”, “file2.txt”};

Output: Error

[1, 3, 4, 10, 8, 9, 10]

Input: invalid (System.exit(0) is already called)

Output: n/a

[1, 3, 4, 10, 8, 11]

Input: invalid

Output: n/a

[1, 3, 4, 6, 7, 8, 9, 10]

Input: same as path [3]

Output: same as path [3]

[1, 3, 4, 6, 7, 8, 11]

Input: same as path[4]

Output: same as path [4]