Test Case (InfixToPostfixTranslator)

No.	Case	Steps	Test Data	Expected Result	Actual Result	Pass/Fail	Comments
1	Input Dialog CANCEL_OPTION	1. Run the program 2. Input Dialog shown 3. Click the cancel button	–	An error message thrown The program terminated normally	java.lang.NullPointerException thrown	NG	To be revised
2	Input Dialog CLOSED_OPTION	1. Run the program 2. Input Dialog shown 3. Click the close button	–	An error message thrown The program terminated normally	java.lang.NullPointerException thrown	NG	To be revised
3	Enter null character	1. Enter null in the text field 2. Click the ok button	–	The program is not goin in while loop to translate from infix to postfix.	The program is not goin in while loop to translate from infix to postfix.	OK	
4	Enter space	1. Enter space in the text field 2. Click the ok button	–	The program is not goin in while loop to translate from infix to postfix.	The program is not goin in while loop to translate from infix to postfix.	OK	
5	Enter an unverified character	1. Enter an unverified character	$9 * 10	An error message thrown.	An error message "$9 is not verified character." thrown.	OK	
6	Enter an unverified character	1. Enter an unverified character	¥2000 – 500	An error message thrown.	An error message "¥2000 is not verified character." thrown.	OK	
7	Enter an unverified character	1. Enter an unverified character	65 * 85%	An error message thrown.	An error message "85% is not verified character." thrown.	OK	
8	Enter without space	1. Enter infix expression without space between characters	3-2/5*8+1/8*3	The input expression is treated as an unverified character. An error message thrown.	An error message "3-2/5*8+1/8*3 is not verified character." thrown.	OK	
9	Enter a negative number	1. Enter a negative number 2. Click the ok button	2 * -3 / 4 – 3	Translate normally.	2 -3 * 4 / 3 –	OK	
10	Put a positive sign (+)	1. Put a positive sign (+) in front of a number 2. Click the ok button	2 * 3 / 4 – +3	Translate normally.	2 3 * 4 / +3 –	OK	
11	Enter postfix expression	1. Enter postfix expression 2. Click the ok button	2 5 2 * 3 / 1 – +	Output postfix expression without any change. Or, validate expression from user input.	2 5 2 3 * 1 / – +	NG	To be revised
12	Operator precedence	1. Enter infix expression different precedence of operators 2. Click the ok button	5 – 4 * 3 – 4 / 2	Multiplication sign (*) and division sign (/) are given higher precedence than addition sign (+) and subtraction sign (–).	5 4 3 * – 4 2 / –	OK	
13	Enter with parentheses	1. Enter infix expression with parentheses 2. Click the ok button	(3 + 4) * (5 – 2) / 4	Characters in parentheses are given the highest precedence.	3 4 + 5 2 – * 4 /	OK	
14	Double parentheses	1. Enter infix expression with double parentheses 2. Click the ok button	(3 + (5 – 2)) / 4	Characters in inner parentheses are given higher precedence than in outer parentheses.	java.util.EmptyStackException thrown.	NG	To be revised
15	Enter floating numbers	1. Enter infix expression with floating point numbers 2. Click the ok button	1.41 * 3.14 – 1.73	Translate normally.	1.41 3.14 * 1.73 –	OK	

16	Enter big data	1. Enter big data 2. Click the ok button	3 + 3 +	Translate normally.	3 3 + +	OK	