Java Software Development Homework 9

Scoring Criteria and Usage Scenario

2015/05/22

In the following usage scenarios, [EXPR] denotes the current calculation (e.g. [123 + 456]), and <BUTTON> denotes the specific button pressed (e.g. <+> means pressing the + button once).

[Part I] Basic Functions

✓ Numerical buttons (4%)

- Corresponding numbers can display correctly on the screen when the numerical buttons (0-9) are pressed.
- If the current number contains no dot (i.e. an integer), the first digit cannot be 0 except the number is indeed 0 (e.g. the number 012 is invalid). The leading digits can contain at most one zero (e.g. the numbers 000 and 00.12 are both invalid).

Below is an example scenario. [123 +] is the current calculation shown on the screen, and after the button 0 is pressed, the current calculation becomes [123 + 0], and so on.

[EXPR]	[123 +] ➔ [123 + 0] ➔ [123 + 0] ➔ [123 + 1] ➔ [123 + 12]
<BUTTON>	<0> ➔ <0> ➔ <1> ➔ <2>

✓ Dot button (4%)

- Dot can occur at most once in a number.
- Dot can occur neither before the first digit nor after the last digit (e.g. the numbers 123. and .123 are both invalid). For these cases, you should append a zero before and after the dot, respectively.

[EXPR]	[123] ➔ [123.] ➔ [123.] ➔ [123.4] ➔ [123.4] ➔ [123.40]
<BUTTON>	<.> ➔ <.> ➔ <4> ➔ <.> ➔ <0>
[EXPR]	[123 +] ➔ [123 + 0.] ➔ [123 + 0.0 +]
<BUTTON>	<.> ➔ <+>

✓ +/- button (4%)

- This button only applies to a non-zero number.

[EXPR]	[1 +] ➔ [1 +] ➔ [1 + 1] ➔ [1 + -1] ➔ [1 + 1]
<BUTTON>	<+/-> ➔ <1> ➔ <+/-> ➔ <+/->
[EXPR]	[0] ➔ [0] ➔ [0.] ➔ [0.] ➔ [0.0] ➔ [0.0] ➔ [0.01] ➔ [-0.01] ➔ [log(-0.01)] ➔ [log(-0.01)]
<BUTTON>	<+/-> ➔ <.> ➔ <+/-> ➔ <0> ➔ <+/-> ➔ <1> ➔ <+/-> ➔ <log> ➔ <+/->

✓ Unary operation buttons (8%)

- Use function-like expression for displaying unary operations (e.g. log(3) or sqrt(5.0)).
- Unary operations only apply to a number or unary operation.
- If the current number is a unary operation, when you press a numerical button, the unary operation should be replaced by the number.

[EXPR]	[1 +] ➔ [1 +] ➔ [1 + 2] ➔ [1 + log(2)] ➔ [1 + sqrt(log(2))] ➔ [1 + 3]
<BUTTON>	<log> ➔ <2> ➔ <log> ➔ <√> ➔ <3>

✓ Binary operation buttons (6%)

- Binary operators can only occur before or after a number or unary operation.
- If the last token is a binary operator, when you press a binary operation button again, the last binary operator will be overwritten.

[EXPR]	[log(10)] ➔ [log(10) +] ➔ [log(10) +] ➔ [log(10) *] ➔ [log(10) ^]
<BUTTON>	<+> ➔ <+> ➔ <*> ➔ <exp>

✓ Equals button (3%)

- The calculation result will show on the screen after pressing this button.

[EXPR]	[1 + 2 + 3] ➔ [6]
<BUTTON>	<=>
[EXPR]	[1 + 2 +] ➔ [3]
<BUTTON>	<=>
[EXPR]	[1 +] ➔ [1]
<BUTTON>	<=>
[EXPR]	[0] ➔ [0]
<BUTTON>	<=>

✓ Memory function buttons (6%)

- When you press M+ or M-, you should show a label "M" indicates that the stored value in the memory is non-empty. When you press MC, you should clear the stored value in the memory and also clear the label.

[EXPR]	[1 + 2] ➔ [1 + 2] ➔ [1 + 2] ➔ [1 + 4] ➔ [1 + 4] ➔ [1 + 0] ➔ [1 + 3] ➔ [1 + 3] ➔ [1 + 3] ➔ [1 + 6] ➔ [1 + 6] ➔ [1 + 0]
<BUTTON>	<M+> ➔ <M+> ➔ <MR> ➔ <M-> ➔ <MR> ➔ <3> ➔ <MS> ➔ <M+> ➔ <MR> ➔ <MC> ➔ <MR>

✓ Undo button (3%)

- This button deletes a digit from the current number.

[EXPR]	[0 + 1.23] ➔ [0 + 1.2] ➔ [0 + 1.] ➔ [0 + 1] ➔ [0 +] ➔ [0 +]
<BUTTON>	<←> ➔ <←> ➔ <←> ➔ <←> ➔ <←>

✓ C button (2%)

- This button reset the calculator to the initial state. (Note that this button will not clear the stored value in the memory.)

[EXPR]	[123 + 456 +] ➔ [0]
<BUTTON>	<C>

✓ CE button (3%)

- This button clears the current entry.

[EXPR]	[1 +] ➔ [1 +] ➔ [1 + 2] ➔ [1 + 2.] ➔ [1 +]
<BUTTON>	<CE> ➔ <2> ➔ <.> ➔ <CE>

- ✓ Show image / show history / clear history (15%)
  - There are three buttons which can show an image, show the calculation history, clear the calculation history, respectively.
  - The image and history list are both inside the same JScrollPane (i.e. it has a scroll bar) so they will not appear at the same time.
  - You should also record the calculation result to the history list. For example, if the current calculation is [1 + 2] and after pressing <=>, you should append "1 + 2 = 3" to the history list.

[Part 2] Calculation Correctness

- ✓ Binary operations are correct (18%, each operation for 3%)
  - All operators have the same precedence (i.e. evaluate the expression from left to right) except that the exponential operation should be evaluated first.
  - For example, the expression [1 + 2 * 3 + 2 ^ 5 / 10] should be evaluated to 4.1.

- ✓ Unary operations are correct (9%, each operation for 3%)
  - Note that by definition the factorial operation only applies to a non-negative integer.

[Part 3] Other Requirements

- ✓ Coding style, UI design (5%)
  - All required GUI components display correctly.
  - Functional buttons (in total 30) are arranged properly using GridLayout.

- ✓ If the calculation result is not a number (e.g. NaN or Infinity, etc.), you should disable all the functional buttons except the button C. You can press the button C to reset the calculator. (5%)

- ✓ The current calculation result will be adapted to the next calculation. In the next calculation, if you press a numerical button first, the former calculation result will be overwritten. (5%)

[EXPR]	[1 + 2] ➔ [3] ➔ [3 +] ➔ [3 + 2] ➔ [5] ➔ [log(5)] ➔ [0.69897] ➔ [1]
<BUTTON>	<=> ➔ <+> ➔ <2> ➔ <=> ➔ <log> ➔ <=> ➔ <1>