JC2002 Java Programming - Practical 7 (Day 9)

The goal of this practical is to get familiar with event-driven programming and implementing functionality in Swing GUIs. We will use the dummy GUI for a calculator implemented in the last practical as a starting point.

1. In the **GraphicalCalculator** class, first add instance integer variable **status** initialised to zero, and string variable **storedText** initialised to *null*. Then, implement action listeners to add functionality to the buttons, as follows:

   - For the button "Clear": set the content of the text field to an empty string using method **setText()** of class **JTextField**, set **status** to zero and **storedText** to "0", and return.
   - For the button "Add": read the string from the text field using method **getText()** of class **JTextField** and store it in **storedValue**., and store the newly parsed value to. Set content of the text field to an empty string and set **status** to 1.
   - For the button "Subtract": read the string from the text field, and convert it to a float and store the newly parsed value to **storedValue**. Set content of the text field to an empty string and set **status** to 2.
   - For the button "Multiply": read the string from the text field, and convert it to a float and store the newly parsed value to **storedValue**. Set content of the text field to an empty string and set **status** to 3.
   - For the button "Divide": read the string from the text field, and convert it to a float and store the newly parsed value to **storedValue**. Set content of the text field to an empty string and set **status** to 4.
   - For the button "Result": check the status. If **status** is zero, just return. Otherwise, convert the value of **storedText** to a float **firstValue** using method **Float.parseFloat()**, then read the string from the text field, and convert it to a float **secondValue**. Then, if **status** is 1, compute **firstValue** + **secondValue** and store the result in float **resultValue**. If status is 2, compute **firstValue** - **secondValue** and store the result in **resultValue**. If status is 3, compute **firstValue** * **secondValue** and store the result in **resultValue**. If **status** is 4, compute **firstValue** / **secondValue** and store the result in **resultValue**. Finally, convert **resultValue** to a string using **String.valueOf()** method, display the resulting value in the text field, and set **status** to zero.

2. You may notice that **Float.parseFloat()** method throws **NumberFormatException** exception if the input string written in the text field is not valid to convert to float. Implement a **try…catch** exception handler to handle illegal input (you can implement a specific method where the text field value is parsed and the exception is handled), and then use **JOptionPane.showMessageDialog()** method to notify the user about illegal input. In case of illegal input, set **status** to zero, **storedText** to *null*, and text field content to an empty string.

3. Think about different ways to improve the calculator. For example, how could you indicate the status to the user?