Programming Applications (PRA), Small Group Tutorial 2, "Hello, can you hear me?"

Please review the document marked 'What is a Code Dojo?' on KEATS if you were unable to attend your first Small Group Tutorial session, or if you require clarification on the structure of Small Group Tutorials in PRA.

If you have any feedback about the format of your small group tutorial sessions, please email Martin and Steffen. If there are reasons noted in your KIP (if you are unaware of what a KIP is, then this instruction does not apply to you), or hitherto undisclosed reasons, that would make participating in the session in the proposed way difficult or impossible, please also let us know.

1 Action Listeners

During this Code Dojo (as you probably expected) you will be bringing life to the static smartphone keypad you built during the first Dojo.

Your TA may have retained a copy of the code that you built as a group during the last session. If not, there is a model solution on KEATS available called 'Keypad.java', which you will build on during this session. During your Code Dojo, if you notice any lines of code that do not make sense to you from this solution, discuss what these lines might achieve with the other students in your session.

The basic functionality I want you to add to the keypad is as follows:

- When each button is pressed, the number associated with that button should appear in the number output field.

- When the delete button is pressed (<), the last number in the number output field should be removed.
- Make the clock at the top of the frame real, so that it advances every minute in accordance with the actual time (thanks to Ali Azam for this idea). *Hint: The classes* javax.swing.Timer *and* java.util.Calendar *should help you with this task.*
- Using the same idea, make the battery (which should start at 100%) drain by 1% every minute (to accurately replicate the average battery life of an iPhone). Make sure the battery level does not fall below zero. If the battery is at 0% all features of the dial pad (e.g. entering numbers, deleting numbers etc.) should become inoperable.

If you have time, explore the following advanced functionality:

- When the store button is pressed (+) any numbers currently in the number output field should be stored to a specific file. Only sets of numbers that do not already exist in this file should be stored.
- In the era before smartphones, we had to press a button multiple times in order to show a letter, typically when writing a text message. For example, pressing the key '2' once would show a '2' in the associated output area, but if the key '2' was pressed again, quickly, then the '2' in the output area would transform into an 'A'. Pressing the '2' button again, quickly, would transform this 'A' into a 'B', and so on for the remaining characters on the '2' key. Thus, pressing the same key multiple times in short succession would allow you to display each potential piece of output from that key, on a loop (i.e. when pressing '2', once 'C' is displayed, pressing '2' again would transform this 'C' back into a '2', because there are no further characters associated with the key '2', so we start again from the beginning). Pressing a button multiple times with a sufficient pause in between each press would output the numeric value associated with that button multiple times as normal.

  Let's forget that Apple use this dial pad to only enter phone numbers, and try and replicate the functionality described above in the GUI: pressing a dial pad button once, or with a sufficient pause between multiple presses, should show the associated number(s) as normal in the number output field. However, pressing this button again, quickly, should transform the output number into the first character it represents, and then into the second when the number is pressed again, and so on. To give an additional example to the one above, pressing the key '8' once should show '8' in the number output, then pressing '8' quickly again should transform this '8' into a 'T'. If '8' is pressed, then the user pauses before pressing '8' again, another '8' should be added to the number output as normal.

> You should avoid printing spaces or parenthesis to the number output.
>
> Even if you cannot implement this functionality directly, discuss with the rest of your group about how you might do so, and which datastructures and method calls would support your solution.

In this Dojo I want you to think carefully about efficiency, and how you can avoid code duplication.