

# Worksheet 4: GUI for Predictive Text Entry

Seyyed Shah and Uday Reddy

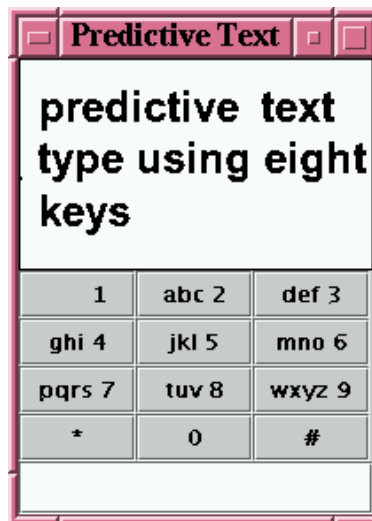
Assigned: Tuesday, 16th February

Deadline : Tuesday, 23rd February, 11:59pm

As usual, include in your submission:

1. appropriate comments and JavaDoc.
2. **thorough** testing. (You may use JUnit wherever applicable.)

In this exercise you will develop a Graphical user interface to the predictive text dictionary implemented in Worksheet 3. The minimal user interface should look like this:



In this GUI, there is one main control mechanism, the buttons in the numeric keypad. The message appears in the text window at the top. By pressing the button “2”, you will get one of “a”, “b”, or “c”. To cycle through the words (or prefixes of words) that match, the user clicks the button labelled “\*”. Pressing “0” completes the entry of the current word and creates a space, ready for the entry of the next word.

You can add other features if you wish, such as a “C” button for clearing the last typed character, a drop-down menu for the list of words that match, up/down buttons to scroll through the menu etc. Feel free to practice working with the Swing API.

## Model-View separation

To interface the GUI with the dictionary, you should use the *Observer* pattern to achieve a model-view separation. (This is also called the “Model View Controller” architecture, even though we don’t necessarily use a separate “controller”.) The dictionary, the text message etc. should be encapsulated in a *Model* class. The GUI with the user interface components should be built in a separate *View* class. The View class should be an Observer (**implements Observer**) and the Model class should be an Observable (**extends Observable**).

## Hints

- Your GUI can be developed using any implementation of the **Dictionary** interface from Worksheet 3, even though the **TreeDictionary** class would work the best. It should not be necessary to modify any of your existing classes to complete this part.
- If you are not satisfied with your Dictionary classes, you can also use the **SampleDictionary** class that we provide on the module web site.<sup>1</sup>
- You can use the demo application provided on the module web page to get a feel for how the user interface should work.
- You may want to implement more features such as number entry by keyboard, punctuation, editing already typed words and adding words to the dictionary, but make sure the basic functionality is working and committed to your Subversion repository first.
- You can find documentation for Swing on-line<sup>2</sup>

---

<sup>1</sup><https://www.cs.bham.ac.uk/internal/courses/java/msc/>

<sup>2</sup>A good place to start: <http://java.sun.com/docs/books/tutorial/uiswing/>