

# Tutorial 4: Exceptions, GUI and Listeners

---

## 1. Tutorial Objectives

There are three levels of difficulty of the tutorial exercises marked with asterisks (\*) as follows:

\* Easy      \*\* Medium      \*\*\* Hard

It is important to do the tutorial exercises in order as they often rely on a previous question to be completed first. Hints will be marked in italics.

You are expected to be able to finish the easy and medium exercises without any help. However, never hesitate to ask the demonstrator if you have any questions.

## 2. Exercises

**Note:** When you need to build a java GUI, you are free to build it using code or using Eclipse WindowBuilder.

See: <http://www.vogella.com/tutorials/EclipseWindowBuilder/article.html>

### \* Exercise 1      Javadoc

Create documentation for your final Chessboard game classes using the javadoc tool.

### \* Exercise 2      Currency Converter

Build a full GUI version of a currency conversion program e.g. for currency exchange rate conversion (GBP, USD, YEN, EU)

- Use jTextField to enter in data for the amount to convert.
- Use a JComboBox in the design view, to pull down from a list of currencies (so in the case of Currency Converter, a name and rate must be defined in an Object class called Currency, and use an ArrayList called Currencies to hold each of the Currency objects).
- Write out the conversion values into a file the first time it is run and they enter them, check if the file exists each time it runs.

### **\*\* Exercise 3      Whack a Mole Game**

Write a graphics program to simulate a game of Whack a mole. A 4 x 4 grid of black holes is places on the screen. As a timer counts down, different images appear over the 4 x 4 grid. A player should be able to click on a mole and if it is caught before it vanishes, a score increments.

You are welcome to use JLabels with ImageIcon for the images or any other technique for drawing clickable images. Use nested loops for the creation of the grid.

As time progresses, the speed of the moles appearing gets faster. Create parameters for Easy, Medium and Hard speeds. Store the top 10 scores with a name, in a file.

### **\* Exercise 4      File Handling**

Write a java program that will count the number of lines in each file that is specified on the command line. Assume that the files are text files. Note that multiple files can be specified, as in "java CountLines file1.txt file2.txt file3.txt". Write each file name, along with the number of lines in that file, to standard output.

Exceptions handling should be implemented as well. If an error occurs while trying to read from one of the files, you should print an error message for that file, but you should still process all the remaining files.

### **\*\* Exercise 5      Practice on JFrame and Listener**

For each of below bullet points, have a different GUI JFrame, set with a different layout.

- Write a keylistener that as you enter text in a Jtextfield it counts and updates a JLabel. A sneakier task would be to get it to append to a file as you type (Keystroke loggers do this).
- Write an itemlistener for a JComboBox with several items. You can simply output the text to a JOptionPane messagebox to say it was selected. Note that JComboBoxes have indices, as they are array-like in holding data together (so first value is at myjComboInstance.getSelectedIndex(0); )
- Write a focuslistener that as you enter any jComponent (jbutton or jlabel etc) it will update a JLabel with a message.
- Write a mouselistener (and mousemotionlistener) for making parts of a GUI clickable and then draggable, e.g. with Jbuttons, JLabels (with ImageIcon), JRadioButtons, JComboBoxes. Output coordinates of the movement and name of each selected item on a JLabel at the bottom of the JFrame.

~~~ 000 ~~~