Threads and Java Swing

Swing is multi-threaded. Your project 3 program, and all Swing programs, use 2 different threads by default.

There is the main thread. This is the thread that runs the main method, the constructor for the class, etc.

There is the event dispatch thread. This is the thread responsible for displaying the Swing gadgets on the screen and it runs the event listener methods.

Problems: Because the constructor is not run by the event dispatch thread, you can get some errors. When you call setVisible(true), the event dispatch thread

is told to display the JFrame. However, if your program is still putting gadgets on the JFrame, those gadgets might be placed on -after- the event dispatch thread

displays the JFrame, and so they will not appear.

Because the event dispatch thread runs the event listeners, if you have code in the actionPerformed (or some other listener) method, while that code is running, the

event dispatch thread will be unable to update your JFrame. So, you will see your program's window "freeze" while that code is running.

The BadGUI example is an example that demonstrates one of these problems. The different buttons cause different actions, with the sort button taking a long time. While

the sort is going on, the window is frozen.

The solution:

1. The SwingUtilities class includes a method called invokeLater that takes a Thread (or any Runnable) and sends it to the event dispatch thread to run.

This should be used by the constructor (or any other method called when launching your program), and you should pass to the event dispatch thread all code that involves

displaying swing gadgets to the screen.

2. The SwingWorker class has methods to have the event dispatch thread create a separate thread (doInBackground) and to communicate between the

event dispatch thread and the new thread (publish, process, and done).

This should be used in the actionPerformed method to do all time consuming tasks. Only the tasks that change the display should be sent back to the event dispatch thread

so they can be displayed properly.

The GoodGUI example demonstrates the use of both of these classes.