

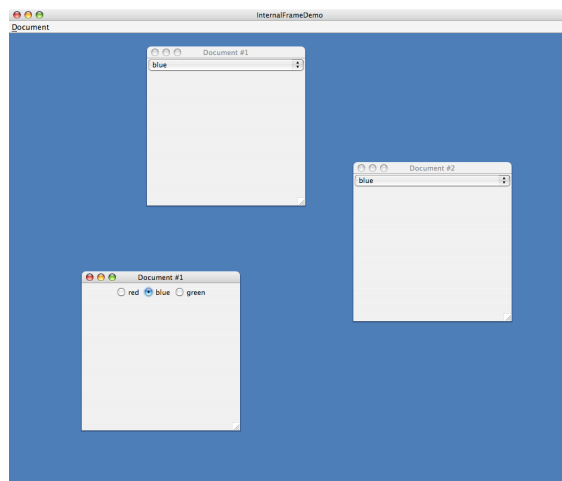
Homework 9

Guidelines

While we acknowledge that beauty is in the eye of the beholder, you should nonetheless strive for elegance in your code. Not every program which runs deserves full credit. Make sure to state invariants in comments which are sometimes implicit in the informal presentation of an exercise. If auxiliary functions are required, describe concisely what they implement. Do not reinvent wheels, and try to make your functions small and easy to understand. Use tasteful layout and avoid long winded and contorted code. None of the problems require more than a few lines of Java code. Your solution must be your own work.

Graphical User Interfaces

You are asked to implement a graphical user interface that looks a little bit like that.



Some code that gives you the blue background has been provided as `TheThreeColors.java` and is available from the webpage.

Problem 1

Define a model for a color object that can be either red, blue, or green.

Problem 2

Define three different windows `JInternalFrame` that pose different views of this rather trivial model. The first kind of frame should give you a drop down menu with the three color option. The second kind of frame gives you three radio buttons again for the three colors, and the third kind of frame simply displays the current color stored in the model (not displayed in the image). In the provided file, you see the three methods `createFrame` for the first kind, `createBauttonFrame` for the second, and `createColorFrame` for the third. Please fill them in. Don't worry yet about updating the model yet, we do this next.

Problem 3

For each frame window in Problem 2, we notice that there are two directions of listening. A frame must listen to the model, and update itself if the model changes. In the first two cases, the model must listen to the frame, and update itself when the content of the frame changes (i.e. when some user input occurred).

Problem 4

Follow the methodology outlined in class, and implement an action listener for the model. Use the standard `ActionListner` class for user events and the listener for the model from Problem 3. Do not use `Observer` and `Observable` classes. It is important that your code for each graphical element is local.