

Object-Oriented Programming

Laboratory 7



Exercise 1: Frame setup

To exit a program, we can use `System.exit(0)`. To close a window, we call the `dispose()` method.

Question 1.1: Create a main program which displays two windows. Call the `dispose()` method when each window is closed.

Question 1.2: By using the method `setLocation(int x, int y)`, ensure that all windows are displayed next to each other when you run your program.

Question 1.3: Have a button on the first window change a label on the second window. Deal with the case where the second window is closed before the first.

Exercise 2: Elements setup within a Frame

Question 2.1: Create a window of size 800x600, not resizable (using method `setResizable`) with a `FlowLayout`.

Question 2.2: Create a new class called `Chess`, inheriting from `Panel` and a new class called `Text`, inheriting from `Button`. Call the `pack()` method. Does the result look nice to you?

Question 2.3: Modify the method `getPreferredSize()` of both `Chess` and `Text` classes, so that the former returns 600x600, and the latter returns 200x600. Does the result look nicer?

Exercise 3: Menu bar

A menu bar is an object of type `java.awt.MenuBar`. It is composed of several objects of type `java.awt.Menu`, each containing `java.awt.MenuItem` objects.

Question 3.1: Add a menu bar to a window using the `setMenuBar(MenuBar)` method. Your menu bar will contain two menus: `File` and `Edit`. The `File` menu contains the command `Quit`. The `Edit` menu contains commands `Copy` and `Paste`. Up to now, these commands have no actions.

Question 3.2: Add an `ActionListener` to the `Quit` command by using `addActionListener` to the corresponding `MenuItem`. The click on `Quit` will quit the program.

Question 3.3: Create a shortcut for the `Copy` and `Paste` commands. The shortcut is added using `setShortcut(MenuShortcut)` on each `MenuItem`. The shortcut for `Copy` is 'c' and the shortcut for `Paste` is 'v'.

Exercise 4: Dialog box

Dialog boxes are used to ask questions to the user.

Question 4.1: When the user clicks on a button, create a new `java.awt.Dialog` object and display it using `setVisible(boolean)`. Add a new button in this dialog.

Question 4.2: Add a button creating a FileDialog, and add a label with the name of the file which is selected (using method getFile()).