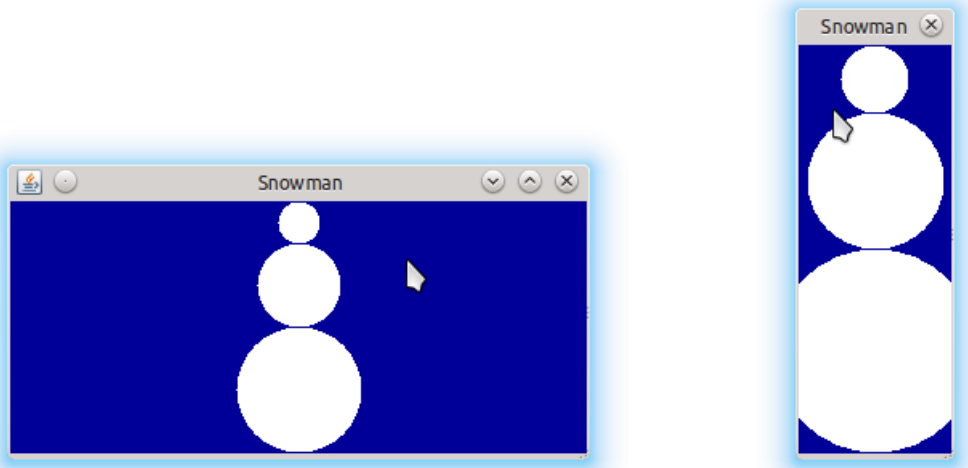
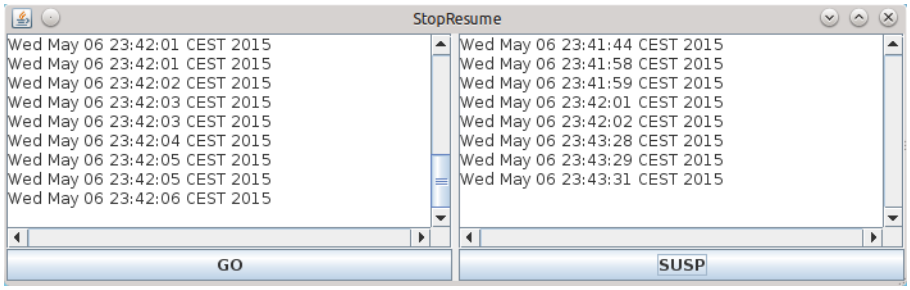
Problem 1

Write a program drawing a snowman consisting of three balls the diameters of which are in 1:2:3 proportion. These proportions should be kept when we resize the window. The snowman should always fill the full height of the window:



Problem 2

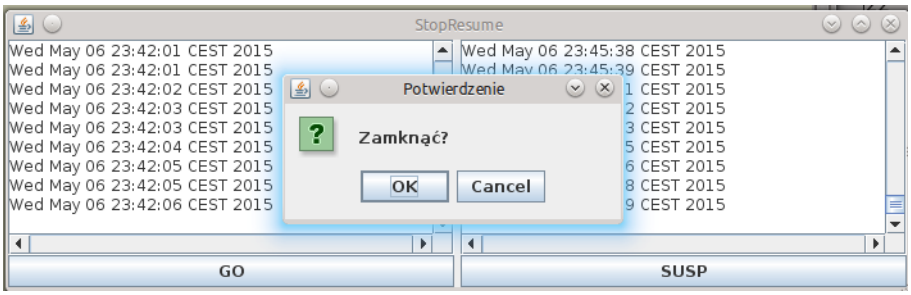
Write a program displaying an interface similar to the one below:



In both text areas (with sliders), current time is shown, updated every randomly chosen period of time (e.g., from the range $[500, 1500]$ ms. Each area is controlled by a separate thread. Below the text areas there are buttons; clicking on one of them causes:

- suspending (but *not* 'killing'!) the corresponding thread if it is running (the text on the button is then changed from 'SUSP' to 'GO');
- resuming the corresponding thread (but *not* launching another one!) if it is stopped (the text on the button is then changed from 'GO' to 'SUSP').

An attempt to terminate the program by closing the widow should provoke a dialog to be displayed in the middle of the interface and asking the user for confirmation (after pressing 'CANCEL' the program should continue):



(this effect can be obtained by using the method **setDefaultCloseOperation** on the main frame and installing a listener of events of type **WindowEvent** implementing interface **WindowListener** or, more conveniently, extending abstract class **WindowAdapter**).

Notes:
Do *not* use deprecated and erroneous methods **stop**, **resume** and **suspend** from class **Thread**! Use the *wait-notify* mechanism instead.
You can force the text area to scroll to the bottom after a call to append like this:

```
textArea.setCaretPosition(textArea.getDocument().getLength());
```
