Computer Science A, Spring 2007

NATBAS, Roskilde Universitety

Miniproject 3

March 29, 2007

This miniproject asks you to implement graphical application for displaying bar charts based on numerical values read from one or more files. The project is open, and it is up to you to decide which features you want to incorporate, how to structure the program into classes, and how to implement the details.

The answer to this exercise should consist of a Java program that you develop yourself. You must hand in a small report that presents your program, that shows at least a few example runs, and that explains the limitations of your solution, if any. (You are welcome to extend the program with additional features.)

The program should be designed and implemented by you alone. This implies that **you may not** work together in groups. It also implies that **you may not** ask someone else to implement your solution **nor may you** download parts of your solution from the internet.

You answer should be handed in before Tuesday, April 17, at 13:00.

Morten Rhiger
`mir@ruc.dk`

1 A bar chart drawing program

In this exercise you are asked to implement a program that reads numerical data from one or more files and displays the data as *bar charts* in a windows.

1.1 Minimum requirements

You program should at least be able to read a sequence of numbers (positive or negative) from a file, and display these values as bar charts in a window (a `JFrame`).

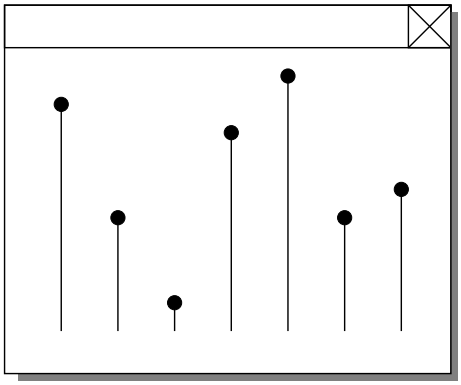

Figure 1a. Bar charts

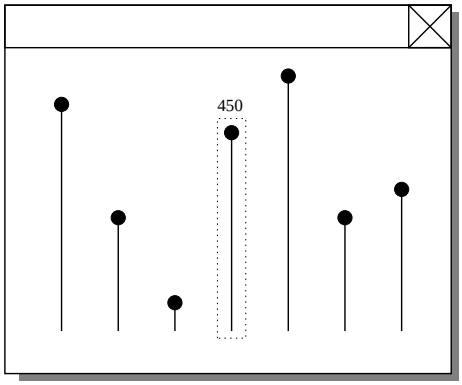

Figure 1b. Highlighting

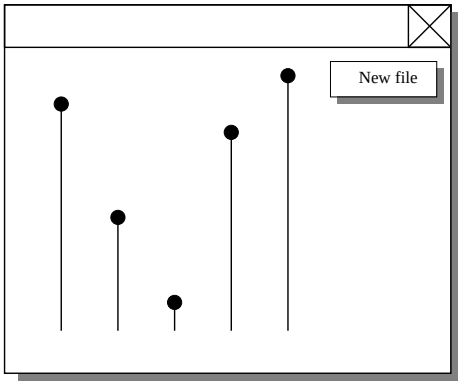

Figure 1c. A load button

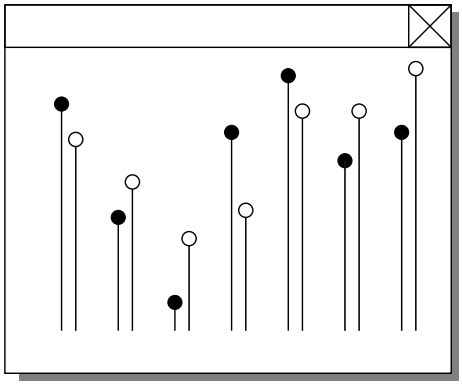

Figure 1d. Multiple charts

Your program is required to scale ("stretch out") the bars so that they fit within the window. It is also required to adjust the space between the bars so that all bars fit within the window. If the window is resized, the bar chart should be stretched to fit within the new dimensions. An example window is displayed in Figure 1a.

1.2 Extensions

Furthermore, you should extend your program with one (or more, if you wish) of the following features:

1. Highlighting

   When the user moves the mouse over a bar, the bar becomes highlighted and the value of the bar is displayed (somewhere inside the window). The result might look as displayed in Figure 1b.

2. A load button

   When the user clicks on a button (a `JButton`), a dialog (a `JFileChooser`, see Advanced Topic 16.1 in *Java Concepts*) will open and ask for another file. If the user enters the name of a file, the old bar char will be discarded, and the content

of the new file is displayed instead. The main window might look as displayed in Figure 1c.

3. Multiple charts

   Rather than displaying the content of one file, your program should display the content of several files. To distinguish them from each other, bars from different files should be given different colors. The result might look as displayed in Figure 1d. It is up to you to decide how the individual files are specified.

1.3 Approach

It is important that you structure your solution into smaller parts that can be implemented and tested independently.

You may find it necessary to use the class `java.util.Scanner` to read input from a file typed in by the player. To read input from a file, construct an object of the class `java.io.FileReader` as follows.

```
FileReader reader = new FileReader("myfile.dat");
Scanner in = new Scanner(reader);
```

See section 16.1 in *Java Concepts* for further explanations.

If you find it difficult to get started, try to implement a program that reads data from a file and stores is as an object of a suitable class. Ask yourself what operations such a class should provide.

Try also to implement another program that draws a fixed bar chart in a window. (That is, a bars whose values are fixed in the program.) Then experiment with stretching.