Lab Exercises 7

Dr. Sarvar Abdullaev
s.abdullaev@inha.uz

November 26, 2020

You must attempt all exercises given in this lab. After completing it, you must push it to your corresponding GitHub Classroom repository. Note, you do not have to upload compiled Java bytecode or screenshot of your program's output.

1 Calculator GUI

Create the following GUI given in Figure 1. You do not have to provide any functionality. You can create it either programmatically or using IntelliJ's GUI Design tool.

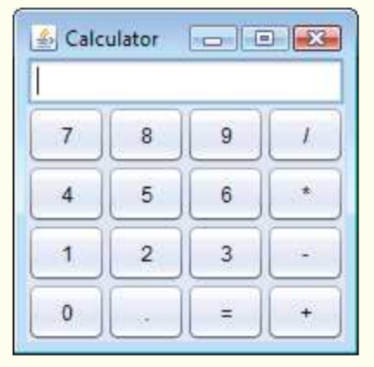
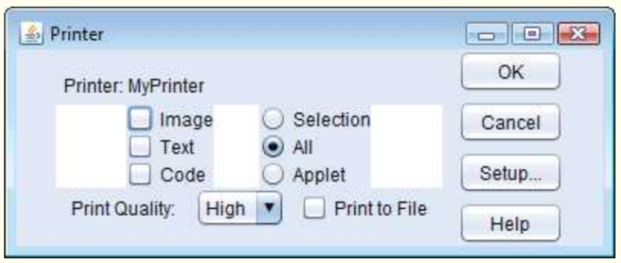


Figure 1: Calculator GUI

2 Checkboxes and Radio buttons GUI

Create the following GUI given in Figure 2. You do not have to provide any functionality. You can create it either programmatically or using IntelliJ's GUI Design tool.


Printer: MyPrinter

□ Image ○ Selection
□ Text ● All
□ Code ○ Applet

Print Quality: High ▾ □ Print to File

Figure 2: Checkbox and Radio buttons GUI

3 Temperature Conversion

Write a temperature-conversion application that converts from Fahrenheit to Celsius. The Fahrenheit temperature should be entered from the keyboard (via a `JTextField`). A `JLabel` should be used to display the converted temperature. Use the following formula for the conversion:

$$Celsius = \frac{5}{9} \times (Fahrenheit - 32)$$

4 Guess-the-Number Game

Write an application that plays “guess the number” as follows: Your application chooses the number to be guessed by selecting an integer at random in the range 1–1000. The application then displays the following in a label:

```
I have a number between 1 and 1000. Can you guess my number?
Please enter your first guess.
```

A `JTextField` should be used to input the guess. As each guess is input, the background color should change to either red or blue. Red indicates that the user is getting ”warmer,” and blue, ”colder.” A `JLabel` should display either ”Too High” or ”Too Low” to help the user zero in. When the user gets the correct answer, ”Correct!” should be displayed, and the `JTextField` used for input should be changed to be uneditable. A `JButton` should be provided to allow the user to play the game again. When the `JButton` is clicked, a new random number should be generated and the input `JTextField` changed to be editable.

5 Ecofont

Ecofont (`https://www.ecofont.com/download`)—developed by SPRANQ (a Netherlands-based company)—is a free, open-source computer font designed to reduce by as much as 20% the amount of ink used for printing, thus reducing also the number of ink cartridges used and the environmental impact of the manufacturing and shipping processes (using less energy, less fuel for shipping, and so on). The font, based on sans-serif Verdana, has small circular “holes” in the letters that are not visible in smaller sizes—such as the 9- or 10-point type frequently used. Download Ecofont, then install the font file `Spranq_eco_sans_regular.ttf` using the instructions from the Ecofont website. Next, develop a GUI-based program that allows you to type in a text string to be displayed in the Ecofont. Create *Increase Font Size* and *Decrease Font Size* buttons that allow you to scale up or down by one point at a time. Start with a default font size of 9 points. As you scale up, you’ll be able to see the holes in the letters more clearly. As you scale down, the holes will be less apparent. What is the smallest font size at which you begin to notice the holes?