

HOMEWORK PROBLEMS 17

- 1) Where does the `setTitle` method called on line 11 in Fig. 17.1 come from?
- 2) What is a container?
- 3) How does a listener object determine which component triggers an event?
- 4) Are radio buttons individually added to a container or are they added all together in one group?
- 5) Why does an applet not have a `main` method?
- 6) Run the GUI1 program in Fig. 17.1. Resize the window by placing the cursor on the lower right corner of the window and dragging (hold the left button down). What happens to the placement of the components on the window?
- 7) Comment out line 19 (the `setLayout`) in `C17h7.java` (a copy of GUI1 in Fig. 17.1). Compile and run. What is the effect of the modification?
- 8) Place a `println` statement that displays `hello` in the `performedAction` method in `C17h8.java` (a copy of GUI3 in Fig. 17.5). Compile and run. What happens every time you click the count button?
- 9) Add a second button to `C17h9.java` (a copy of GUI3) program that terminates the window when clicked. To terminate the window, execute

```
System.exit(0);
```

- 10) Compile and run `C17h10.java` (a copy of GUI5 in Fig. 17.7). What happens if you enter a non-integer into the text field and then click on the count button? Modify the program so that it can handle invalid data in the text field. To do this, use a `try/catch` block. Test your program.
- 11) Add the statements below to the `performAction` method in the `C17h11.java` program. Compile and run. Click on the count button.

```
System.out.println("RED:    " + Color.RED);
System.out.println("GREEN:  " + Color.GREEN);
System.out.println("BLUE:   " + Color.BLUE);
System.out.println("BLACK:  " + Color.BLACK);
System.out.println("WHITE:  " + Color.WHITE);
System.out.println("GRAY:   " + Color.GREY);
System.out.println("ORANGE: " + Color.ORANGE);
System.out.println("MAGENTA:" + Color.MAGENTA);
System.out.println("CYAN:   " + Color.CYAN);
```

The computer screen can produce red, green, and blue light. However, if these colors are mixed, our eyes will perceive other colors. For example, mixing red and green gives orange. Every color is represented with three numbers, each in the range 0-255, that specify the intensity of its red, green, and blue components. For example, pure red is [255, 0, 0] and orange is [255, 200, 0].

- 12) Set line 20 in the `C17h12.java` (a copy of GUI1 in Fig. 17.1) to

```
contentPane.setBackground(new Color(255, 200, 0));
```

Compile and run. What is the background color? Now change this statement to

```
contentPane.setBackground(new Color(255, 255, 255));
```

Compile and run. What is the background color now? What color do you get when you mix red, green and blue?

- 13) Modify `C17h13.java` (a copy of `GUI1` in Fig. 17.1) so that it adds five buttons to the frame and no labels. Compile and run. Resize the window by dragging the window's right corner. How does the size of the window affect the placement of the buttons?
- 14) Same as homework problem 13 except modify `C17h14.java` (a copy of `GUI1` in Fig. 17.1), and use `BorderLayout`.
- 15) Same as homework problem 13 except modify `C17h15.java` (a copy of `GUI1` in Fig. 17.1), use `GridLayout` with 3 rows and 2 columns.
- 16) Modify the `C17h16.java` (a copy of `GUI5` in Fig. 17.7) in the following way:
 - a) Change line 4 to


```
class C17h16 extends JFrame implements ActionListener
```
 - b) Move the `performAction` method to the `C17h16` class.
 - c) Delete the `Listener` inner class
 - d) Change line 26 to


```
button.addActionListener(this);
```
 - e) Change line 43 to


```
text1.addActionListener(this);
```

Does the modified program still work? Why is `this` passed to the `addActionListener` method?

- 17) Open the `index.html` file in Fig. 17.8a with a web browser after you have compiled the `Applet1.java` applet in Fig. 17.8c. How are the three words "An", "applet", and "demonstration" laid out on the screen? Add extra spaces after each of these words in `index.html`. How does this change affect the layout? Put each item in `index.html` on a separate line. How does this change affect the layout?
- 18) Write a GUI application program with two text fields, one button, and one label. Use the text fields for input. When the button is clicked, the sum of the numbers in the two text fields should be added and displayed in the label. Hint: to set the text in a label, use the `setText` method (`setText` is in label objects as well as text objects).
- 19) Modify the `C17h19.java` (a copy of `GUI5` in Fig. 17.7) so that it does not use a count variable. Instead, it should use the text in the text field to keep track of the current count. Each time the count button is clicked, access the text in the text field, convert it to type `int` (with `Integer.parseInt`), add or subtract 1 depending on which radio button is selected, and store the result back in the text field as a string. Using this approach, you do not need a listener for the text field component.
- 20) Add to `C17h20.java` (a copy of `GUI5` in Fig. 17.7) a second group of five radio buttons that specify the size of the increment/decrement. The five radio buttons should correspond to the increment/decrement sizes 1, 2, 3, 4, and 5. For example, if the radio button 4 is selected, then the count should be incremented or decremented by 4 each time the count button is clicked.
- 21) Write a GUI program that has two panels. One panel should have three text fields and a button labeled `Show Color`. Label the text fields `Red`, `Green`, and `Blue`. When the `Show Color` button is clicked, the color specified by the three text fields should be displayed in the other panel. If any of the text fields do not hold an integer in the range 0 to 255 when the `Show Color` button is clicked, set the offending field or fields to 0. Then proceed to display the color.
- 22) Radio buttons are used to specify mutually exclusive options. Check boxes are similar to radio buttons, but are used to specify options that not mutually exclusive. Any combination of check boxes can be checked at one time.

A check box is created from the `JCheckBox` class. Add two check boxes to the `C17h22.java` (a copy of `GUI5` in Fig. 17.7) program, one on each panel, that control the background color of its panel. When a checkbox on a panel is checked, the background color on that panel should go white. When unchecked, the background color should revert to its original color (yellow for panel 1 or green for panel 2).

23) Convert the `GUI5` application in `C17h23.java` to an applet. Embed in HTML and view with a web browser.

24) `JOptionPane` is a “quick and easy” GUI. Run the following program. What does it do?

```
import javax.swing.JOptionPane;
class C17h24
{
    public static void main(String[] args)
    {
        String s = JOptionPane.showInputDialog("Enter non-neg number");
        double d = Double.parseDouble(s);
        JOptionPane.showMessageDialog(null, "Root = " + Math.sqrt(d));
        System.exit(0);
    }
}
```

25) What does the applet below do? Experiment with different arguments to see how they affect the display.

```
import javax.swing.*;
import java.awt.Graphics;
public class C17h25 extends JApplet
{
    public void paint(Graphics p)
    {
        p.drawOval(0, 0, 50, 50);
        p.fillOval(100, 0, 50, 50);
        p.drawArc(0, 200, 50, 50, 0, 90);
    }
}
```

26) Write an applet that displays a happy face with eyes, eyebrows, nose, ears, and mouth on the screen. See homework problem 25.