

Section Solutions 4

Based on a handout by Eric Roberts, Patrick Young, and Jeremy Keeshin

Problem One: The Wizard of Java

The output of this program is shown here:

```
dorothy = Somewhere over the rainbow...
toto.getX() = 2.718
dorothy = Somewhere over the rainbow...
scarecrow = 137
dorothy = Somewhere over the rainbow...
scarecrow = 137
toto.getX() = 99.9
```

Problem Two: Rubber-Banding

```
import acm.program.*;
import acm.graphics.*;
import java.awt.event.*;

public class RubberBanding extends GraphicsProgram {
    public void run() {
        addMouseListeners();
    }

    /* The line that is currently being dragged around, or null if no
     * line is currently being drawn.
     */
    private GLine currentLine;

    public void mousePressed(MouseEvent e) {
        /* Set up a new rubber-banded line by setting the current line
         * to be a new line with both endpoints at the indicated point.
         */
        currentLine = new GLine(e.getX(), e.getY(), e.getX(), e.getY());
        add(currentLine);
    }

    public void mouseDragged(MouseEvent e) {
        /* There must be a line currently being drawn, since otherwise
         * the mouse couldn't be dragged. Update its endpoint to be
         * the current mouse position.
         */
        currentLine.setEndPoint(e.getX(), e.getY());
    }

    /* There is no need to have a mouseReleased method, since once the
     * mouse is released the dragging stops.
     */
}
```

Problem Three: Adding Commas to Numeric Strings

```
/**
 * Given a numeric string (a string of digits), inserts commas into
 * the string as appropriate and produces a new string.
 *
 * @param The string of digits
 * @return The updated string.
 */
private String addCommasToNumericString(String digits) {
    /* Build up the resulting string. */
    String result = "";

    /* Track how many digits we have seen so far. We'll use this to
     * determine whether or not we need a comma.
     */
    int nDigits = 0;

    /* Process the digits backwards so we know where the commas go. */
    for (int i = digits.length() - 1; i >= 0; i--) {
        /* Prepend the current digit to the result string. */
        result = digits.charAt(i) + result;

        /* Record that we've seen another digit. If this pushes us up
         * to the next group of three and there are digits before this
         * one prepend a comma to the result.
         */
        nDigits++;
        if ((nDigits % 3 == 0) && (i > 0)) {
            result = "," + result;
        }
    }
    return result;
}
```