CS106A
Summer 2013-2014

Handout #08S
July 15, 2014

Solutions for Section #3

Portions of this handout by Eric Roberts and Mehran Sahami

Problem 1: Colorful Circles

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

public class Circles extends GraphicsProgram {

        /* Determines the size of the color selector boxes */
        private static final int BOX_SIZE = 50;


        // instance variables for drawing a circle
        private GOval oval;
        private int centerX, centerY;


        // instance variables for color selection
        private Color currentColor;
        private GRect blackBox, redBox, yellowBox;


        public void run() {
                addMouseListeners();

                // create the color boxes
                blackBox = createBox(Color.black, BOX_SIZE);
                redBox = createBox(Color.red, BOX_SIZE);
                yellowBox = createBox(Color.yellow, BOX_SIZE);

                // add them to the top-left corner
                add(blackBox, 0, 0);
                add(redBox, 0, BOX_SIZE);
                add(yellowBox, 0, 2 * BOX_SIZE);

                // set the initial color
                currentColor = Color.black;
        }


        /* Creates a box with the given color and size. */
        private GRect createBox(Color color, double size) {
                GRect rect = new GRect(size, size);
                rect.setFilled(true);
                rect.setFillColor(color);
                return rect;
        }
        public void mousePressed(MouseEvent e) {
                // get the element at the mouse position
                GObject obj = getElementAt(e.getX(), e.getY());

                if (obj == blackBox) {
                        currentColor = Color.black;
                } else if (obj == redBox) {
                        currentColor = Color.red;
                } else if (obj == yellowBox) {
                        currentColor = Color.yellow;
                } else {
                        addCircle(e.getX(), e.getY());
                }
        }

        /* Adds a circle centered at the given x,y coordinate. */
        private void addCircle(int x, int y) {
                // mark the center position for future reference
                centerX = x;
                centerY = y;

                // create and add the circle
                oval = new GOval(0, 0);
                oval.setFilled(true);
                oval.setColor(currentColor);
                oval.setFillColor(currentColor);
                add(oval, centerX, centerY);
        }

        /* update the size of the circle, keeping center fixed */
        public void mouseDragged(MouseEvent e) {
                double radius = computeRadius(e.getX(), e.getY());
                oval.setLocation(centerX - radius, centerY - radius);
                oval.setSize(2 * radius, 2 * radius);
        }

        private double computeRadius(int mouseX, int mouseY) {
                double dx = mouseX - centerX;
                double dy = mouseY - centerY;
                return Math.sqrt(dx*dx + dy*dy);
        }
}
```

Problem 2: Adding Commas to Numeric Strings

```
        private String addCommasToNumericString(String digits) {
                String result = "";
                int len = digits.length();
                int nDigits = 0;
                for (int i = len - 1; i >= 0; i--) {
                        result = digits.charAt(i) + result;
                        nDigits++;
                        // we also need i > 0 so we don’t have a comma before the
                        if (((nDigits % 3) == 0) && (i > 0)) {
                                result = "," + result;
                        }
                }
                return result;
        }
```

Problem 3: Deleting characters from a string

```
private String removeAllOccurrences(String str, char ch) {
        String result = "";
        for (int i = 0; i < str.length(); i++) {
                if (str.charAt(i) != ch) {
                        result += str.charAt(i);
                }
        }
        return result;
}
```

Problem 4: Converting a string to alternating capital letters

```
private String altCaps(String str) {
        String result = "";
        int counter = 0;
        for (int i = 0; i < str.length(); i++) {
                if (Character.isLetter(str.charAt(i))) {
                        counter++;
                }
                if ((counter % 2) == 0) {
                        result += Character.toUpperCase(str.charAt(i));
                } else {
                        result += Character.toLowerCase(str.charAt(i));
                }
        }
        return result;
}
```