Bill Bohling
Course: Java Programming 1
Date: 15 November 2010
Assignment: HW8

Part A
Objective: Incorporate a scrolling banner into the Calculator applet from HW6
Solution: Implemented a BannerThread class and added a banner section to the calculator GUI

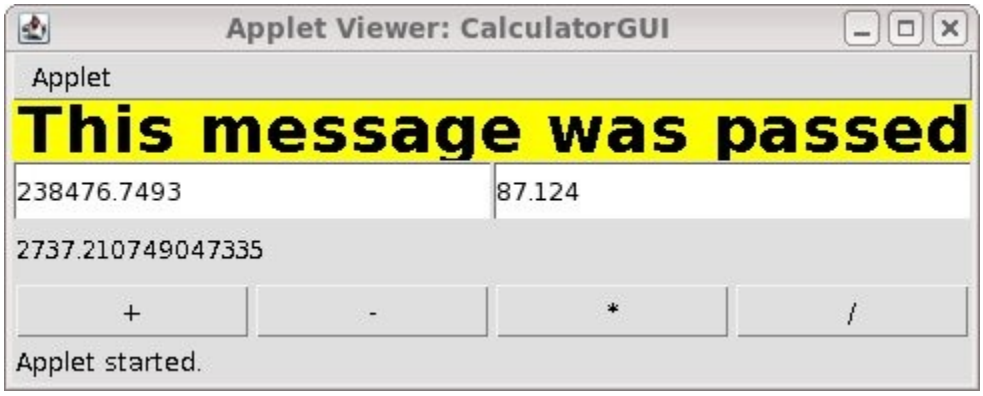
Code:
```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

/*
<applet code="CalculatorGUI" width=480 height=120>
</applet>
*/

public class CalculatorGUI extends Applet {

  // the GUI bits
  protected TextField t1 = new TextField(16);
  protected TextField t2 = new TextField(16);
  protected Label result = new Label("result here", Label.LEFT);
  protected Button addButton = new Button("+");
  protected Button subButton = new Button("-");
  protected Button multButton = new Button("*");
  protected Button divButton = new Button("/");
  protected Panel inputPanel = new Panel();
  protected Panel resultPanel = new Panel();
  protected Panel buttons = new Panel();

  // add a banner section to the GUI
  protected Label banner = new Label();
  protected Panel bannerPanel = new Panel();
  protected Font bannerFont = (new Font("SansSerif", Font.BOLD, 32));

  public void init() {
    setLayout(new GridLayout(4,1));
    // set up the banner
    banner.setText(getParameter("message"));
    bannerPanel.setLayout(new GridLayout(1,1));
    bannerPanel.setBackground(Color.YELLOW);
    bannerPanel.setFont(bannerFont);
    add(bannerPanel);
    bannerPanel.add(banner);

    // start banner thread
    new BannerThread(banner);
```

```
    // set up the calculator
    inputPanel.setLayout(new GridLayout(1,2));
    add(inputPanel);
    inputPanel.add(t1);
    inputPanel.add(t2);

    resultPanel.setLayout(new GridLayout(1,1));
    add(resultPanel);
    resultPanel.add(result);

    buttons.setLayout(new GridLayout(1,4));
    add(buttons);
    buttons.add(addButton);
    buttons.add(subButton);
    buttons.add(multButton);
    buttons.add(divButton);

    // now let the logic class do everything else
    CalculatorLogic cl = new CalculatorLogic(this);
  }


}

class CalculatorLogic implements ActionListener {

  private CalculatorGUI gui;

  CalculatorLogic(CalculatorGUI cg){
    gui = cg;
    gui.addButton.addActionListener(this);
    gui.subButton.addActionListener(this);
    gui.multButton.addActionListener(this);
    gui.divButton.addActionListener(this);
  }

  public void actionPerformed(ActionEvent e) {

    Object button = e.getSource();
    double num1 = Double.parseDouble(gui.t1.getText());
    double num2 = Double.parseDouble(gui.t2.getText());
    String opResult;

    if (button == gui.addButton) {
      opResult = Double.toString(num1 + num2);
    }
    else if (button == gui.subButton) {
      opResult = Double.toString(num1 - num2);
    }
    else if (button == gui.multButton) {
```

```
      opResult = Double.toString(num1 * num2);
    }
    else {
      if (num2 != 0) {
        opResult = Double.toString(num1 / num2);
      }
      else {
        opResult = "no, no, no division by 0";
      }
    }
    gui.result.setText(opResult);

  } // actionPerformed

} //CalculatorLogic

class BannerThread implements Runnable {
  String msg;
  Label banner;
  Thread t = null;
  boolean stopFlag;

  BannerThread(Label blabel){
    banner = blabel;
    msg = " " + banner.getText() + " ";
    t = new Thread(this);
    stopFlag = false;
    t.start();
  }

  public void run() {
    char ch;

    for ( ; ; ){
      try {
     banner.setText(msg);
        Thread.sleep(250);
     ch = msg.charAt(0);
     msg = msg.substring(1, msg.length());
     msg += ch;
        if(stopFlag){
      break;
        }
      } catch (InterruptedException e) {
        // nothing to do, really
      }
    }
  }

}
```

Test, passes the banner text in as an applet parameter:

```
<html>
<head>
<title>Calculator</title>
</head>
<body>
<applet code="CalculatorGUI" width=480 height=120>
<param name="message" value="This message was passed in from the
applet tag" />
</applet>
</body>
</html>
```



Part B:
Objective: Synchronize a method so Deposits work
Solution: Synchronize the deposit method in the Account class

```
class Account {

    // original deposit method declaration:
    // void deposit(int amount, String name) {
    // new deposit declaration:
    synchronized void deposit(int amount, String name) {

        int balance;

        System.out.println(name + " trying to deposit " + amount);
        System.out.println(name + " getting balance...");

        balance = getBalance();

        System.out.println(name + " balance got is " + balance);

        balance += amount;

        System.out.println(name + " setting balance...");
```

```
        setBalance(balance);

        System.out.println(name + " new balance set to " +
Deposit.balance);

    }

...

}
```

Output:

```
$ java Deposit
#1 trying to deposit 1000
#1 getting balance...
#1 balance got is 1000
#1 setting balance...
#1 new balance set to 2000
                            #2 trying to deposit 1000
                            #2 getting balance...
                            #2 balance got is 2000
                            #2 setting balance...
                            #2 new balance set to 3000
*** Final balance is 3000
```