William O’Sullivan

CIS 1500

Powerball Program

12/5/2013

Mega Millions Program

Purpose – To display a gui that generates random numbers for the user and winning picks in a Mega Millions situation, then compares them and adds a running total.

Visualize – The program will have a panel that asks for how many tickets to purchase, with a go button, then will generate random winning numbers and compare them to the users generated winning numbers with a running net gain/loss.

Pseudocode-

Import necessary tools

Create the buttons, panels, and text fields

Set Window dimensions

Set ints for the variables to create the picks for the user and winning draws

Set arrays for the winning picks and user picks

Create the public class for the Mega Millions program

Build panels

For each panel, put logical buttons and functions into each

Set for with if-else-if to display the results

Compare boolean if true or not to see if numbers match

End program

Logical Errors – All over the place. I could not get the function of 5 numbers without the separate mega millions number to compile. After rewriting, googling, and even trying to create classes for each individual function, program would not compile.

Code -

import javax.swing.\*;  
import java.awt.event.\*;  
import java.awt.\*;  
import java.util.Random;  
  
public class MegaMillions extends JFrame  
{  
   
 private JButton button1;  
 private JPanel panel1;  
 private JPanel userPicks;  
 private JPanel lotteryPanel;  
 private JPanel resultsPanel;  
 private JLabel messageLabel;  
 private JTextField ticketsTextField;  
 private final int WINDOW\_WIDTH = 400;  
 private final int WINDOW\_HEIGHT = 100;  
   
 Random randomNumbers = new Random();  
   
 final int NUMBERS = 5;  
 final int MAX\_NUMBER = 75;  
 final int MEGAMILLIONS = 1;  
 final int MAX\_MEGAMILLIONS = 15;  
 boolean arraysEqual = true;  
 int matches = 0;  
 int[] userPicks;  
 int[] winningPicks;  
 int loop;  
   
 public MegaMillions()  
 {  
   
 setTitle("Mega Millions");  
   
 setSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);  
   
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
   
 setLayout(new BorderLayout());  
   
 buildPanel();  
 buildUserPanel();  
 buildLotteryPanel();  
 buildResultsPanel();  
   
 add(instructionPanel, BorderLayout.NORTH);  
 add(userPanel, BorderLayout.WEST);  
 add(lotteryPanel, BorderLayout.EAST);  
 add(resultsPanel, BorderLayout.SOUTH);  
   
 pack();  
 setVisible(true);  
  
   
 }  
   
 private void buildPanel()  
 {  
   
 messageLabel = new JLabel("Enter the number of tickets to purchase");  
 ticketsTextField = new JTextField(10);  
 button1 = new JButton("GO!");  
   
 button1.addActionListener(new ButtonListener());  
   
 panel = new JPanel();  
 panel.add(messageLabel);  
 panel.add(ticketsTextField);  
 panel.add(button1);  
   
 }  
   
 private void buildUserPanel()  
 {  
   
 userPanel = newJPanel();  
 userPicks = new JTextField[NUMBERS];  
   
 for (loop = 0; loop <= NUMBERS; loop ++)  
 {  
   
 userPicks[loop] = new JTextField(userPicks);  
 userPanel.add(userPicks[loop]);  
   
 }  
 }  
   
 private void buildLotterPanel()  
 {  
 lotteryPanel = new JPanel();  
   
 draws = new int[MAX\_NUMBER];  
   
 for (loop = o; loop <= NUMBERS; loop ++)  
 {  
 winningNumbers[loop] = new JLabel("" + randomNumbers.nextInt(MAX\_NUMBER));  
   
 draws[loop] ++;  
   
 }  
 }   
   
 private void buildResultsPanel()  
 {  
 resultsPanel = new JPanel();  
 resultsPanel.add(results);  
   
 }   
   
 private class ButtonListener implements ActionListener  
 {  
   
 public void actionPerformed(ActionEvent e)  
 {  
 while (arraysEqual && loop < winningPicks)  
 {  
 if (winningPicks[loop].equals (userPicks[loop]))  
 {  
 arraysEqual = true;  
 matches++;  
 loop++;  
 }  
 else  
 {  
 arraysEqual = false;  
 loop++;  
 }  
 if (matches <= NUMBERS)  
 {  
 results = new JLabel("You matched " + matches + " numbers.");  
 }  
 else  
 {  
 results = new JLabel("You matched all the numbers! You win the Grand Prize!");  
 }   
 }   
 }  
 }   
   
 public static void main (String[] args)  
 {  
   
 MegaMillions megaMillions = new MegaMillions();  
   
 }  
}

Runtime errors- Still could not get my original code to work. It was just way too scattered and all over the place, nearly beyond repair. After seeing the in class example it made much more sense and was able to easily manipulate the code given for a Mega Millions scenario, but my original code after a couple hours still was giving 25 compiling errors. I got it down from 47, but I think thats about as far as I could go without entirely rewriting the program. Both codes are submitted for reveiw.

Original program was not validated, would not compile or run. Modified example program ran correctly and verified.