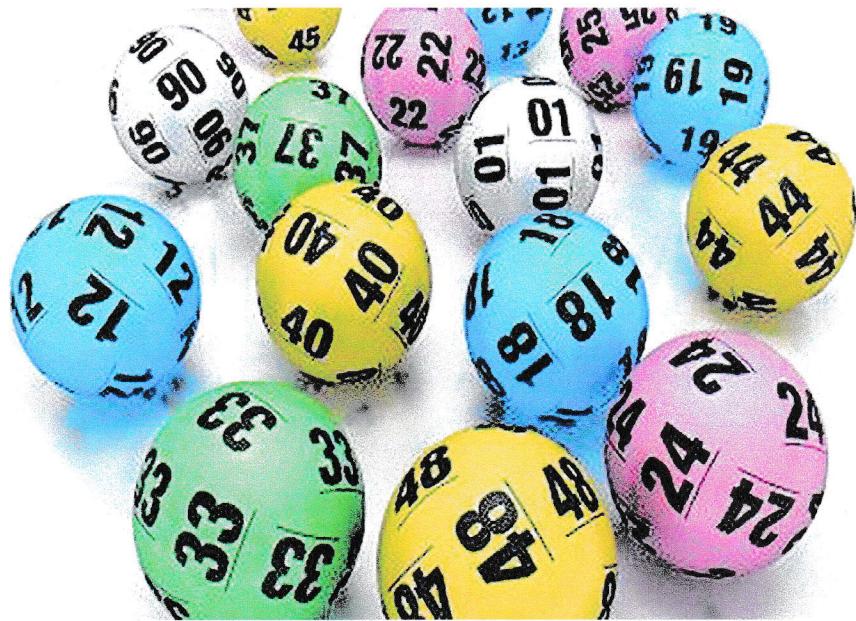


## CIS 162 Project 4

### Mega Millions Lottery Simulation



100  
100

"I pledge that this work is entirely mine, and mine alone. Except for any code provided by my instructor and the code has been commented to reflect where I was assisted by a fellow student."



"I spent 15 hours on this project on March 13, 18, 19, 21 reading the book, designing a solution, writing code, fixing errors and putting together the printed document."



## Class LotteryTicket - project4

1/4

```
1  /**
2  * Project 4
3  * Progam for LotteryTicket.
4  *
5  * Aaron Teague
6  *
7  */
8
9
10 //import number format
11 import java.text.NumberFormat;
12
13 public class LotteryTicket
14 {
15     /** declare variables */
16     private String first;
17     private String last;
18     private int day;
19     private int month;
20     private int year;
21     private String city;
22     private String state;
23     private int zip;
24     private int b1;
25     private int b2;
26     private int b3;
27     private int b4;
28     private int b5;
29     private int megaball;
30     private double prize;
31
32     /** initialize variables */
33     public LotteryTicket(String info)
34     {
35         // Initialise instance variables
36         String [] tokens = info.split(",");
37         first = tokens[0].trim();
38         last = tokens[1].trim();
39         city = tokens[2].trim();
40         state = tokens[3].trim();
41         zip = Integer.parseInt(tokens[4].trim());
42         month = Integer.parseInt(tokens[5].trim());
43         day = Integer.parseInt(tokens[6].trim());
44         year = Integer.parseInt(tokens[7].trim());
45         b1 = Integer.parseInt(tokens[8].trim());
46         b2 = Integer.parseInt(tokens[9].trim());
47         b3 = Integer.parseInt(tokens[10].trim());
48         b4 = Integer.parseInt(tokens[11].trim());
49         b5 = Integer.parseInt(tokens[12].trim());
```

**Class LotteryTicket - project4 (continued)**

2/4

```
50         megaball = Integer.parseInt(tokens[13].trim());  
51     }  
52  
53     /** Get first name. */  
54     public String getFirst()  
55     {  
56         return first;  
57     }  
58  
59     /** Get last name. */  
60     public String getLast()  
61     {  
62         return last;  
63     }  
64  
65     /** Get the city */  
66     public String getCity()  
67     {  
68         return city;  
69     }  
70  
71     /** Get the state. */  
72     public String getState()  
73     {  
74         return state;  
75     }  
76  
77     /** Get the zipcode */  
78     public int getZipcode()  
79     {  
80         return zip;  
81     }  
82  
83     /** Get the day of the DOB */  
84     public int getDay()  
85     {  
86         return day;  
87     }  
88  
89     /** Get the month of the DOB */  
90     public int getMonth()  
91     {  
92         return month;  
93     }  
94  
95     /** Get the year of the DOB */  
96     public int getYear()  
97     {  
98 }
```



## Class LotteryTicket - project4 (continued)

3/4

```
99         return year;
100    }
101
102   /** Get the prize */
103   public double getPrize()
104   {
105       return prize;
106   }
107
108   /** Set the prize */
109   public void setPrize(double amount)
110   {
111       if( amount > 0)
112           prize = amount;
113   }
114
115   /** Checks if the value sent matches one of the ball values */
116   public boolean hasBall(int val)
117   {
118       if(val == b1 || val == b2 || val == b3 || val == b4 || val == b5)
119       {
120           return true;
121       }
122       else
123       {
124           return false;
125       }
126   }
127
128   /** Checks if the value matches the Megaball value. */
129   public boolean hasMegaball(int val)
130   {
131       if( val == megaball)
132       {
133           return true;
134       }
135       else
136       {
137           return false;
138       }
139   }
140
141   /** Allows the to print the contents of the object in plain text */
142   public String toString( )
143   {
144       NumberFormat fmt = NumberFormat.getCurrencyInstance();
145       return first + " " + last + "\n" + city + ", " + state + " " + zip
146       p + "\n" + b1 + " " + b2 +
           " " + b3 + " " + b4 + " " + b5 + "\t" + megaball + "\n" +
```

Mar 21, 2014 9:34:01 PM

**Class LotteryTicket - project4 (continued)**

4/4

```
147             "Prize: " + fmt.format(prize);
148     }
149
150     /** The main tests all aspect of LotteryTicket */
151     public static void main(String args[])
152     {
153         LotteryTicket a = new LotteryTicket("Amy, Zu, Phoenix, AZ, 78234, 4/20
154 /1960, 4, 12, 15, 36, 67, 12");
155         a.setPrize(1000);
156         System.out.println(a.toString());
157         System.out.println(a.getFirst());
158         System.out.println(a.getLast());
159         System.out.println(a.getCity());
160         System.out.println(a.getState());
161         System.out.println(a.getZipcode());
162         System.out.println(a.getDay());
163         System.out.println(a.getMonth());
164         System.out.println(a.getYear());
165         a.setPrize(100);
166         System.out.println(a.getPrize());
167         System.out.println(a.hasBall(12));
168     }
169 }
170 }
```

## Class LotteryMachine - project4

1/9

```
1  /**
2  * Project 4
3  * Lottery Machine code
4  *
5  * Author:Aaron Teague
6  *
7  */
8
9 import java.util.ArrayList;
10 import java.util.Random;
11 import java.util.Arrays;
12 import java.util.Scanner;
13 import java.io.*;
14 import java.text.NumberFormat;
15
16 public class LotteryMachine
17 {
18
19     private ArrayList<LotteryTicket> ticket;
20     private int num1;
21     private int num2;
22     private int num3;
23     private int num4;
24     private int num5;
25     private int megaball;
26
27     /** Constructor instantiate instance variables */
28     public LotteryMachine()
29     {
30         ticket = new ArrayList<LotteryTicket>();
31         num1 = 0;
32         num2 = 0;
33         num3 = 0;
34         num4 = 0;
35         num5 = 0;
36         megaball = 0;
37     }
38
39     /** Adds LotteryTicket object t to the LotteryTicket ArrayList ticket */
40     public void addTicket (LotteryTicket t)
41     {
42         ticket.add(t);
43     }
44
45     /** Returns the amount of tickets in the ticket ArrayList*/
46     public int countTickets()
47     {
48         return ticket.size();
```

**Class LotteryMachine - project4 (continued)**

2/9

```
49     }
50
51     /** Creates the winning numbers by creating random numbers*/
52     private void pickNumbers()
53     {
54         // I was assisted by Jordan Zomerlei in programming this method
55
56         //Creates an array list to hold numbers 1-75
57         ArrayList<Integer> lottoarray = new ArrayList<Integer>();
58
59         for(int i = 1; i <= 75; i++)
60         {
61             lottoarray.add(i);
62         }
63
64         //Creates random generator
65         Random generator = new Random();
66
67         int a = generator.nextInt(75);
68
69         //Create and assign random numbers to integers and removes from list
70         num1 = lottoarray.get(a);
71         lottoarray.remove(a);
72         a = generator.nextInt(74);
73         num2 = lottoarray.get(a);
74         lottoarray.remove(a);
75         a = generator.nextInt(73);
76         num3 = lottoarray.get(a);
77         lottoarray.remove(a);
78         a = generator.nextInt(72);
79         num4 = lottoarray.get(a);
80         lottoarray.remove(a);
81         a = generator.nextInt(71);
82         num5 = lottoarray.get(a);
83         lottoarray.remove(a);
84
85         //sorts the numbers in ascending numbers
86         int [] nums = {num1,num2,num3,num4,num5};
87         Arrays.sort(nums);
88         num1 = nums[0];
89         num2 = nums[1];
90         num3 = nums[2];
91         num4 = nums[3];
92         num5 = nums[4];
93         megaball = generator.nextInt(15) + 1;
94
95     }
96 }
```

## Class LotteryMachine - project4 (continued)

3/9

```
97  /** Counts how many ball numbers LotteryTicket t contains */
98  private int countMatches(LotteryTicket t)
99  {
100     int i = 0;
101     if(t.hasBall(num1))
102         i++;
103     if(t.hasBall(num2))
104         i++;
105     if(t.hasBall(num3))
106         i++;
107     if(t.hasBall(num4))
108         i++;
109     if(t.hasBall(num5))
110         i++;
111
112     return i;
113 }
114
115 /** Calcuates the pay out for each LotteryTicket */
116 private void makePayouts()
117 {
118     for(LotteryTicket t: ticket)
119     {
120         switch(countMatches(t)) 
121     }
122
123     case 5:
124     if(t.hasMegaBall(megaball))
125     {
126         t.setPrize(5000000);
127     }
128     else
129     {
130         t.setPrize(1000000);
131     }
132     break;
133     case 4:
134     if(t.hasMegaBall(megaball))
135     {
136         t.setPrize(5000);
137     }
138     else
139     {
140         t.setPrize(500);
141     }
142     break;
143     case 3:
144     if(t.hasMegaBall(megaball))
145     {
```

## Class LotteryMachine - project4 (continued)

4/9

```
146             t.setPrize(50);
147         }
148     else
149     {
150         t.setPrize(5);
151     }
152     break;
153 case 2:
154 if(t.hasMegaBall(megaball))
155 {
156     t.setPrize(5);
157 }
158 break;
159 case 1:
160 if(t.hasMegaBall(megaball))
161 {
162     t.setPrize(2);
163 }
164 break;
165 case 0:
166 if(t.hasMegaBall(megaball))
167 {
168     t.setPrize(1);
169 }
170 break;
171 default:
172     t.setPrize(0);
173
174 }
175
176 }
177
178 }
179
180 /** Prints out the chosen numbers */
181 private String formatNumbers()
182 {
183     return "Selected Numbers: " + num1 + " " + num2 + " " + num3 + "
184 " + num4 + " " + num5 + "\t" + megaball;
185 }
186
187 /** Picks the winning numbers and makes the payouts to the tickets */
188 public void drawTicket()
189 {
190     //calls the pickNumbers() method
191     pickNumbers();
192     //calls the pickNumbers() method
193     makePayouts();
194 }
```

Mar 21, 2014 9:28:43 PM

```
194
195     /** Assigns the lotto balls and mega ball to the values passed in and
196     * makes payments */
197     public void drawTicket(int b1, int b2, int b3, int b4, int b5, int m)
198     {
199         num1 = b1;
200         num2 = b2;
201         num3 = b3;
202         num4 = b4;
203         num5 = b5;
204         megaball = m;
205         makePayouts();
206     }
207
208     /** Reads the file, creates LotteryTicket objects and adds them to A
209     * rraysList ticket */
210     public void readTickets(String filename)
211     {
212         try
213         {
214             File f = new File(filename);
215             Scanner sc = new Scanner(f);
216             String info;
217
218             while(sc.hasNext())
219             {
220                 info = sc.nextLine();
221
222                 //remember to remove this after it works
223                 //System.out.println(info);
224
225                 LotteryTicket x = new LotteryTicket(info);
226                 ticket.add(x);
227
228             }
229             sc.close();
230         }
231         catch(IOException e)
232         {
233             System.out.println("Failed to read the data file: " +
234                         filename);
235         }
236
237     /** Creates a report that calculates the number of tickets sold, aver
238     * age price, and prints the biggest winner */
239     public String createReport()
240     {
```

**Class LotteryMachine - project4 (continued)**

6/9

```
240     double total = 0.0;
241     NumberFormat fmt = NumberFormat.getCurrencyInstance();
242
243     for(LotteryTicket t: ticket)
244     {
245         total = total + t.getPrize();
246     }
247
248     return "Report for All Sales" + "\n" +
249     formatNumbers() + "\n" +
250         //Calculates total tickets
251     "Tickets sold: " + ticket.size() + "\n" +
252         //Calculates Average price
253     "Average prize: " + fmt.format(total/ticket.size()) + "\n" + "\n"
254     +
255     "Biggest Winner" + "\n" +
256     getBiggestWinner().toString();
257     //Finds biggest winner
258 }
259
260     /** Creates a report that calculates the number of tickets sold, average
261     price, based on the selected State */
262     public String createReport(String st)
263     {
264         double a = 0.0;
265         int b = 0;
266         double x = 0;
267         NumberFormat fmt = NumberFormat.getCurrencyInstance();
268
269         for(LotteryTicket t: ticket)
270         {
271             if(t.getState().equals(st))
272             {
273                 a = a + t.getPrize();
274                 b++;
275             }
276             if(b == 0)
277             {
278                 x = 0;
279             }
280             else
281             {
282                 x = a/b;
283             }
284             return "Report for all sales in " + st + "\n" +
285             formatNumbers() + "\n" +
286             "Tickets sold: " + b + "\n" +
```

## Class LotteryMachine - project4 (continued)

7/9

```
287     "Average prize: " + fmt.format(x) + "\n" + "\n" +
288     "Biggest Winner" + "\n" +
289     getBiggestWinner().toString();
290
291 }
292
293 /** Finds the biggest winner */
294 public LotteryTicket getBiggestWinner()
295 {
296     LotteryTicket biggest = ticket.get(0);
297     for(LotteryTicket t: ticket)
298     {
299         if(t.getPrize() > biggest.getPrize())           ✓
300             biggest = t;
301     }
302     return biggest;
303 }
304
305
306 /** Finds the oldest player */
307 public LotteryTicket getOldestPlayer()
308 {
309     LotteryTicket oldest = ticket.get(0);
310     for(LotteryTicket t: ticket)
311     {
312
313         if(t.getYear() < oldest.getYear())
314         {
315             oldest = t;
316         }
317         else if(t.getYear() == oldest.getYear())
318         {
319             if(t.getMonth() < oldest.getMonth())
320             {
321                 oldest = t;
322             }
323             else if(t.getMonth() == oldest.getMonth())
324             {
325                 if(t.getDay() < oldest.getDay())
326                 {
327                     oldest = t;
328                 }
329             }
330         }
331     }
332
333     return oldest;
334 }
335 }
```

## Class LotteryMachine - project4 (continued)

8/9

```
336
337     /** Finds winners that are greater than or equal to amount */
338     public ArrayList <LotteryTicket> getMajorWinners(double amount)
339     {
340         ArrayList<LotteryTicket> x = new ArrayList<LotteryTicket>();
341         for(LotteryTicket t: ticket)
342         {
343             if(t.getPrize() >= amount)
344             {
345                 x.add(t);
346             }
347         }
348         return x;
349     }
350
351
352     /** Draws tickets until a player wins a Jackpot. After ever drawing t
353      hat doesn't yield a jackpot 1.5million dollars is added to the Jackpot */
353     public String multipleGames()
354     {
355         //Holds the true/false if Jackpot was won
356         boolean jackpotWinner = false;
357         double prize = 50000000;
358         //Keeps track of how many drawings have occurred
359         int i = 0;
360         //Starts a new drawing
361         drawTicket();
362         //Check to see if a jackpot has been won
363         while(jackpotWinner == false)
364         {
365
366             //Iterates through ticket
367             for(LotteryTicket t: ticket)
368             {
369
370                 megaball
371                 //Checks if the LotteryTick objects has all 5 matches and
372                 if(t.hasMegaBall(megaball) && countMatches(t) == 5)
373                 {
374                     jackpotWinner = true;
375                     t.setPrize(prize);
376                 }
377                 //Increments the amount of drawings
378                 i++;
379                 //Adds 1.5million to the jackpot,
380                 prize += 1500000;
381                 //Starts new drawing
382                 drawTicket();
```

Mar 21, 2014 9:28:44 PM

```
383
384      }
385      return  "Biggest Winner" + "\n" +
386      "Number of games: " + i + "\n" +
387      getBiggestWinner().toString( );
388
389  }
390
391 }
392 }
```

## Class example - project4

1/2

```
1  /**
2  * Examples of testing code.
3  *
4  * Author: Aaron Teague
5  *
6  */
7
8 import java.util.ArrayList;
9 public class example
10 { public static void main(String args[]){
11     LotteryMachine m = new LotteryMachine();
12     String info;
13     String info2;
14     String info3;
15
16     info = "Jane, Smith, San Francisco, CA, 49401, 2/14/1983, 1,2,3,4
17 ,5,6";
18     LotteryTicket tix = new LotteryTicket(info);
19     m.addTicket(tix);
20
21     info = "Aaron, Teague, Zeeland, MI, 49464, 7/26/1985, 8,27,34,45,
22 25,6";
23     LotteryTicket tix2 = new LotteryTicket(info);
24     m.addTicket(tix2);
25
26     info = "Barak, Obama, Washington DC, VA, 20004, 7/26/1985, 6,5,4,
27 3,2,1";
28     LotteryTicket tix3 = new LotteryTicket(info);
29     m.addTicket(tix3);
30
31
32     m.readTickets("TicketInfo.txt");
33
34     m.drawTicket();
35
36     System.out.println(m.createReport());
37     System.out.println("");
38     System.out.println(m.createReport("NY"));
39     System.out.println("");
40     System.out.println(m.getOldestPlayer().toString());
41     System.out.println("");
42     System.out.println(m.getBiggestWinner().toString());
43     System.out.println("");
44     System.out.print(m.multipleGames());
45
46 }
```

Mar 21, 2014 8:52:32 PM

## Class GUI - project4

1/7

```
1 import javax.swing.*;
2 import java.awt.event.*;
3 import java.awt.*;
4 import java.util.*;
5 import java.io.*;

6 ****
7 * GUI front end for Lottery Simulation
8 *
9 * @author Scott Grissom
10 * @version February 1, 2013
11 *
12 ****
13 public class GUI extends JPanel{
14     /** buttons */
15     JButton randomNum = new JButton ("Random Numbers");
16     JButton pickNum = new JButton ("Pick Numbers");
17     JButton multiGames = new JButton ("Multiple Games");
18     JButton bigWin = new JButton ("Biggest Winner");
19     JButton oldWin = new JButton ("Oldest Winner");
20     JButton winners = new JButton ("All Major Winners");

21
22     JLabel checkWinners = new JLabel ("Check Winners");
23     JLabel prizeL = new JLabel ("Prize $");
24     JLabel stateL = new JLabel ("ST");
25
26     JTextField prizeT = new JTextField();
27     JTextField stateT = new JTextField();
28     LotteryMachine lm; ✓
29
30     /** text fields */
31
32     /** results box */
33     private JTextArea results;
34     private JFrame theGUI;
35
36
37     /** menu items */
38     private JMenuBar menus;
39     private JMenu fileMenu;
40     private JMenu reportsMenu;
41     private JMenuItem quitItem;
42     private JMenuItem openItem;
43     private JMenuItem stateItem;
44     private JMenuItem reportItem;

45
46     public static void main(String arg[]){
47         // the tradition five lines of code
48         // normally place here are
49         // inserted throughout the constructor
```

```
50         new GUI();
51     }
52
53     /*********************************************************************
54     */
55     /* Constructor - instantiates and displays all of the GUI components
56     */
57     public GUI(){
58         //new LotteryMachine and reads tickets
59         lm = new LotteryMachine();
60         lm.readTickets("TicketInfo.txt");
61
62         theGUI = new JFrame("Mega Million Lottery");
63         theGUI.setVisible(true);
64         theGUI.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
65
66         // create the Results Area for the Center area
67         results = new JTextArea(20,20);
68         JScrollPane scrollPane = new JScrollPane(results);
69         theGUI.add(BorderLayout.CENTER, scrollPane);
70
71         // create the South Panel
72         JPanel southPanel = new JPanel();
73         southPanel.setLayout(new BoxLayout(southPanel, BoxLayout.X_AXIS));
74 ;
75         theGUI.add(BorderLayout.SOUTH, southPanel);
76         southPanel.add(prizeL);
77         southPanel.add(prizeT);
78         southPanel.add(stateL);
79         southPanel.add(stateT);✓
80
81         // create the East Panel
82         JPanel eastPanel = new JPanel();
83         eastPanel.setLayout(new BoxLayout(eastPanel, BoxLayout.Y_AXIS));
84         theGUI.add(BorderLayout.EAST, eastPanel);
85         eastPanel.add(Box.createVerticalGlue());
86         eastPanel.add(new JLabel("Draw Ticket"));
87         eastPanel.add(Box.createVerticalGlue());
88
89         eastPanel.add(randomNum);
90         ButtonListener listener = new ButtonListener();
91         randomNum.addActionListener(listener);
92
93         eastPanel.add(Box.createVerticalGlue());
94         eastPanel.add(pickNum);
95         pickNum.addActionListener(listener);
```

```
96         eastPanel.add(Box.createVerticalGlue());
97         eastPanel.add(multiGames);
98         multiGames.addActionListener(listener);
99
100        eastPanel.add(Box.createVerticalGlue());
101        eastPanel.add(checkWinners);
102
103        eastPanel.add(Box.createVerticalGlue());
104        eastPanel.add(bigWin);
105
106        bigWin.addActionListener(listener);
107
108        eastPanel.add(Box.createVerticalGlue());
109        eastPanel.add(oldWin);
110        oldWin.addActionListener(listener);
111
112        eastPanel.add(Box.createVerticalGlue());
113        eastPanel.add(winners);
114        winners.addActionListener(listener);
115
116        // set up File menus
117        setupMenus();
118        theGUI.pack();
119    }
120
121    /*****
122     * List all entries given an ArrayList of tickets
123
124     @param tix list of all tickets
125     *****/
126
127    public void displayTickets(ArrayList <LotteryTicket> tix) {
128        for (LotteryTicket t: tix){
129            results.append(t + "\n");
130        }
131    }
132
133    //Accepts values for ball from user and passes them to the drawTicket
134    Method
135    private void pickNumbers()
136    {
137
138        String str = JOptionPane.showInputDialog("Enter Ball #1:");
139        int b1 = Integer.parseInt(str);
140
141        String str2 = JOptionPane.showInputDialog("Enter Ball #2:");
142        int b2 = Integer.parseInt(str2);
143
144        String str3 = JOptionPane.showInputDialog("Enter Ball #3:");
145
146    }
```

```
142     int b3 = Integer.parseInt(str3);
143     String str4 = JOptionPane.showInputDialog("Enter Ball #4:");
144     int b4 = Integer.parseInt(str4);
145     String str5 = JOptionPane.showInputDialog("Enter Ball #5:");
146     int b5 = Integer.parseInt(str5);
147     String strM = JOptionPane.showInputDialog("Enter Megaball:");
148     int m = Integer.parseInt(strM);
149
150     lm.drawTicket(b1, b2, b3, b4, b5, m);
151 }
152 */
153
154 ****
155 * Respond to menu selections and button clicks
156
157 @param e the button or menu item that was selected
158 ****
159 */
160 private class ButtonListener implements ActionListener{
161
162     public void actionPerformed(ActionEvent e){
163
164         LotteryTicket c = null;
165
166         // menu item - quit
167         if (e.getSource() == quitItem) {
168             System.exit(1);
169         }
170
171         //Finds oldest player
172         if(e.getSource() == oldWin){
173             results.setText("Oldest Player" + "\n" +
174             lm.getOldestPlayer().toString());
175         }
176
177         //Finds biggest winner
178         if(e.getSource() == bigWin){
179             results.setText("Biggest Winner" + "\n");
180             results.append(lm.getBiggestWinner().toString());
181         }
182
183         //Creates random numbers and pays out tickets
184         if(e.getSource() == randomNum){
185             lm.drawTicket();
186         }
187
188         //Plays multiple games
189         if(e.getSource() == multiGames)
```

**Class GUI - project4 (continued)**

5/7

```
189         {
190             results.setText(lm.multipleGames());
191         }
192
193         //Prints Winners whose prize is as big or bigger than the amo
194         unt passed
195         if(e.getSource() == winners)
196         {
197             //Displays error message if Prize is left blank
198             if(prizeT.getText().equals(""))
199             {
200                 JOptionPane.showMessageDialog(null,
201                     "Please enter an amount for Prize.",
202                     "Prize error",
203                     JOptionPane.ERROR_MESSAGE);
204             }
205             else
206             {
207                 double a = Double.parseDouble(prizeT.getText());
208                 displayTickets(lm.getMajorWinners(a));
209             }
210
211             //Prints report of information about tickets
212             if(e.getSource() == reportItem)
213             {
214                 results.setText(lm.createReport());
215             }
216
217             //Prints report based the State passed
218             if(e.getSource() == stateItem)
219             {
220                 //Prints error message if State is left blank
221                 if(stateT.getText().equals(""))
222                 {
223                     JOptionPane.showMessageDialog(null,
224                         "Please enter a state.",
225                         "State error",
226                         JOptionPane.ERROR_MESSAGE);
227                 }
228                 else
229                 {
230                     results.setText(lm.createReport(stateT.getText()));
231                 }
232             }
233
234             //Picks numbers
235             if(e.getSource() == pickNum)
236             {
```

```
237         pickNumbers();
238     }
239
240     }
241   }
242 }
*****
* Set up the menu items
***** */
246 private void setupMenus() {
247
248     // create menu components
249     fileMenu = new JMenu("File");
250     quitItem = new JMenuItem("Quit");
251     openItem = new JMenuItem("Open...");
252     reportsMenu = new JMenu("Reports");
253     stateItem = new JMenuItem("by State");
254     reportItem = new JMenuItem("All Tickets");
255
256     // assign action listeners
257     ButtonListener ml = new ButtonListener();
258     quitItem.addActionListener(ml);
259     openItem.addActionListener(ml);
260     stateItem.addActionListener(ml);
261     reportItem.addActionListener(ml);
262
263     // display menu components
264     fileMenu.add(openItem);
265     fileMenu.add(quitItem);
266     reportsMenu.add(reportItem);
267     reportsMenu.add(stateItem);
268     menus = new JMenuBar();
269
270     menus.add(fileMenu);
271     menus.add(reportsMenu);
272     theGUI.setJMenuBar(menus);
273 }
274
275 ****
*
276 In response to the menu selection - open a data file
***** */
278 private void openFile(){
279     JFileChooser fc = new JFileChooser(new File(System.getProperty("user.dir")));
280     int returnVal = fc.showOpenDialog(theGUI);
```

**Class GUI - project4 (continued)**

7/7

```
281     // did the user select a file?  
282     if (returnVal == JFileChooser.APPROVE_OPTION) {  
283         String filename = fc.getSelectedFile().getName();  
284         //db.readTickets(filename);  
285     }  
286 }  
287 }  
288 }
```