

```

1  import java.awt.BorderLayout;
2  import java.awt.Color;
3  import java.awt.Dimension;
4  import java.awt.FlowLayout;
5  import java.awt.Font;
6  import java.awt.GridLayout;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import java.time.LocalDate;
10 import java.time.YearMonth;
11 import java.time.format.DateTimeFormatter;
12 import java.util.Calendar;
13 import java.util.GregorianCalendar;
14
15 import javax.swing.BorderFactory;
16 import javax.swing.JButton;
17 import javax.swing.JFrame;
18 import javax.swing.JLabel;
19 import javax.swing.JPanel;
20 import javax.swing.JTextArea;
21 import javax.swing.border.EmptyBorder;
22
23 public class CalendarTest {
24
25     protected static String[] monthNames = { "January", "February", "March", "April",
26         "May", "June", "July", "August", "September", "October", "November",
27         "December" };
28
29     protected static String[] dayNames = { "Su", "Mo", "Tu", "We", "Th", "Fr", "Sa" };
30     public static int monI, dayI, yearI, dayS;
31
32     public static boolean arrow = false;
33
34     public static JFrame frame = new JFrame("Calendar");
35
36     public static JPanel dayPanel = new JPanel();
37     public static JPanel westPanel = new JPanel();
38     public static JPanel monthPanel = new JPanel();
39
40
41     public static void createFrame() {
42
43         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44         frame.setSize(800,600);
45         frame.setLayout(new BorderLayout());
46         frame.setVisible(true);
47
48         frame.add(createTitleGUI(), BorderLayout.PAGE_START);
49
50         //center space
51         JPanel blankPanel = new JPanel();
52         blankPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
53         blankPanel.setBackground(Color.WHITE);
54         frame.add(blankPanel, BorderLayout.CENTER);
55
56         //add WEST side
57         frame.add(createWestGUI(), BorderLayout.LINE_START);
58
59         //add SOUTH side
60         frame.add(createSouthGUI(), BorderLayout.PAGE_END);
61
62         //add EAST side
63         frame.add(createEastGUI(), BorderLayout.EAST);
64         //System.out.println("hi");
65     }
66
67     protected static JPanel createEastGUI() {
68
69         //ok and cancel are under createSouthGUI(), they are supposed to save or delete

```

the data but I don't know how to store or delete them.

```
70
71 JPanel notePanel = new JPanel();
72 notePanel.setLayout(new BorderLayout());
73
74 JLabel fillLabel = new JLabel("                                Add your notes
below                                ");
75 fillLabel.setFont(new Font(" ", Font.BOLD, 14));
76 fillLabel.setBackground(Color.LIGHT_GRAY);
77
78
79 JTextArea noteTA = new JTextArea("");
80 notePanel.add(fillLabel, BorderLayout.PAGE_START);
81 notePanel.add(noteTA, BorderLayout.CENTER);
82
83 return notePanel;
84 }
85
86 protected static JPanel createWestGUI() {
87
88
89
90     westPanel.setLayout(new BorderLayout());
91     westPanel.add(monthLabelGUI(), BorderLayout.PAGE_START);
92     westPanel.add(createDaysGUI(), BorderLayout.CENTER);
93
94     return westPanel;
95 }
96
97 protected static JPanel monthLabelGUI() {
98
99     monthPanel.setLayout(new BorderLayout());
100    monthPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
101
102    if(dayS == 0)
103        dayS = 1;
104
105    if (!arrow) {
106
107        //get month
108        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd");
109        LocalDate localDate = LocalDate.now();
110        //System.out.println(dtf.format(localDate)); //2016/11/26
111        String mon = dtf.format(localDate).substring(5,7);
112        monI = Integer.parseInt(mon) - 1;
113        if (monI < 0)
114            monI = 11;
115
116        //get day
117        String day = dtf.format(localDate).substring(8,10);
118        dayI = Integer.parseInt(day);
119
120        //get year
121        String year = dtf.format(localDate).substring(0,4);
122        yearI = Integer.parseInt(year);
123
124        //display year
125        JLabel monthLabel = new JLabel("                                " + monthNames[monI] + " " +
dayI + " ", " " + yearI);
126        monthLabel.setFont(new Font(" ", Font.BOLD, 14));
127        monthLabel.setBackground(Color.LIGHT_GRAY);
128        monthPanel.add(monthLabel, BorderLayout.CENTER);
129    }
130
131    //display buttons
132    JButton left = new JButton("<");
133    left.setBackground(Color.LIGHT_GRAY);
134    JButton right = new JButton(">");
135    right.setBackground(Color.LIGHT_GRAY);
```

```

136 monthPanel.add(left, BorderLayout.LINE_START);
137 monthPanel.add(right, BorderLayout.LINE_END);
138
139 left.addActionListener(new ActionListener()
140 {
141     public void actionPerformed(ActionEvent e)
142     {
143         arrow = true;
144         monI = monI - 1;
145         if (monI < 0) {
146             monI = 11;
147             yearI = yearI - 1;
148         }
149         monthPanel.removeAll();
150         JLabel monthLabel = new JLabel(" " + monthNames[monI] + " " +
151             dayS + ", " + yearI);
152         monthLabel.setFont(new Font(" ", Font.BOLD, 14));
153         monthLabel.setBackground(Color.WHITE);
154         monthPanel.add(monthLabel);
155
156         //buttons
157         monthPanel.add(left, BorderLayout.LINE_START);
158         monthPanel.add(right, BorderLayout.LINE_END);
159         monthPanel.repaint();
160         monthPanel.revalidate();
161         frame.add(createWestGUI(), BorderLayout.LINE_START);
162         monthPanel.add(monthLabel);
163     }
164 });
165
166 right.addActionListener(new ActionListener()
167 {
168     public void actionPerformed(ActionEvent e)
169     {
170         arrow = true;
171         monI = monI + 1;
172         if (monI > 11) {
173             monI = 0;
174             yearI = yearI + 1;
175         }
176         monthPanel.removeAll();
177         JLabel monthLabel = new JLabel(" " + monthNames[monI] + " " +
178             dayS + ", " + yearI);
179         monthLabel.setFont(new Font(" ", Font.BOLD, 14));
180         monthLabel.setBackground(Color.WHITE);
181         monthPanel.add(monthLabel);
182
183         //buttons
184         monthPanel.add(left, BorderLayout.LINE_START);
185         monthPanel.add(right, BorderLayout.LINE_END);
186         monthPanel.repaint();
187         monthPanel.revalidate();
188         frame.add(createWestGUI(), BorderLayout.LINE_START);
189         monthPanel.add(monthLabel);
190     }
191 });
192
193 return monthPanel;
194 }
195
196 protected static JPanel createTitleGUI() {
197     JPanel titlePanel = new JPanel();
198     titlePanel.setBackground(Color.WHITE);
199     JLabel titleLabel = new JLabel("Yahooogle Calendar");
200     titleLabel.setFont(new Font(" ", Font.BOLD, 30));
201     titlePanel.add(titleLabel);
202

```

```

203         return titlePanel;
204     }
205
206     protected static JPanel createSouthGUI() {
207
208         JPanel buttonsPanel = new JPanel();
209         buttonsPanel.setBackground(Color.WHITE);
210         JButton okButton = new JButton("OK");
211         JButton cancelButton = new JButton("cancel");
212         JPanel blank1 = new JPanel();
213         blank1.setBackground(Color.WHITE);
214         JPanel blank2 = new JPanel();
215         blank2.setBackground(Color.WHITE);
216         JPanel blank3 = new JPanel();
217         blank3.setBackground(Color.WHITE);
218
219         buttonsPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
220
221         buttonsPanel.setLayout(new GridLayout(1,6,10,10));
222         buttonsPanel.add(blank1);
223         buttonsPanel.add(blank2);
224         buttonsPanel.add(okButton);
225         buttonsPanel.add(cancelButton);
226
227         okButton.addActionListener(new ActionListener() {
228             public void actionPerformed(ActionEvent e) {
229                 String identification = Integer.toString(yearI) +
230                     Integer.toString(monI) + Integer.toString(dayS);
231                 dateInfo di = new dateInfo("");
232                 //di.setId(identification);
233                 //di.setInfo("Hello World");
234                 //System.out.println(di.getInfo());
235             }
236         });
237
238         return buttonsPanel;
239     }
240
241     protected static JPanel createDaysGUI() {
242
243         dayPanel.removeAll();
244         dayPanel = new JPanel();
245         dayPanel.setBackground(Color.WHITE);
246         dayPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
247         dayPanel.setLayout(new GridLayout(0, dayNames.length));
248
249         for (int i = 0; i < dayNames.length; i++) {
250             JPanel dPanel = new JPanel(true);
251             dPanel.setBorder(BorderFactory.createLineBorder(Color.BLACK));
252             JLabel dLabel = new JLabel(dayNames[i]);
253             dPanel.add(dLabel);
254             dayPanel.add(dPanel);
255         }
256
257         //getting first day dynamically
258         Calendar mycal = new GregorianCalendar(yearI, monI, dayI);
259         mycal.set(Calendar.DAY_OF_MONTH, 1);
260         String fDay = mycal.getTime().toString().substring(0,3);
261
262         int space = 0;
263
264         //add buffer space based on first day
265         switch (fDay) {
266
267             case "Sun" :
268                 break;
269
270             case "Mon" :

```

```

271         space = 1;
272         break;
273
274     case "Tue" :
275         space = 2;
276         break;
277
278     case "Wed" :
279         space = 3;
280         break;
281
282     case "Thu" :
283         space = 4;
284         break;
285
286     case "Fri" :
287         space = 5;
288         break;
289
290     case "Sat" :
291         space = 6;
292         break;
293
294     }
295
296     for (int k = 0; k < space; k++) {
297         JButton empty = new JButton();
298         empty.setBackground(Color.LIGHT_GRAY);
299         dayPanel.add(empty);
300     }
301
302     JPanel dPanel = new JPanel();
303     dPanel.setBorder(BorderFactory.createLineBorder(Color.BLACK));
304     dPanel.add(new JLabel());
305     int nDays = mycal.getActualMaximum(Calendar.DAY_OF_MONTH);
306     JButton[] num = new JButton[32];
307     for (int i = 1; i < nDays + 1; i++) {
308         num[i] = new JButton(Integer.toString(i));
309         num[i].setBackground(Color.LIGHT_GRAY);
310         dayPanel.add(num[i]);
311         String identification;
312         if (i < 10)
313             identification = Integer.toString(yearI) + Integer.toString(monI) + "0"
314                 + Integer.toString(i);
315         else
316             identification = Integer.toString(yearI) + Integer.toString(monI) +
317                 Integer.toString(i);
318         dateInfo di = new dateInfo(identification);
319         di.setId(identification);
320         num[i].addActionListener(new ActionListener() {
321             public void actionPerformed(ActionEvent e) {
322                 arrow = true;
323                 String substring =
324                     identification.substring(Math.max(identification.length() - 2, 0));
325                 dayS = Integer.parseInt(substring);
326
327                 //display buttons
328                 JButton left = new JButton("<");
329                 left.setBackground(Color.LIGHT_GRAY);
330                 JButton right = new JButton(">");
331                 right.setBackground(Color.LIGHT_GRAY);
332                 monthPanel.add(left, BorderLayout.LINE_START);
333                 monthPanel.add(right, BorderLayout.LINE_END);
334
335                 //action listener left
336                 left.addActionListener(new ActionListener()
337                 {
338                     public void actionPerformed(ActionEvent e)
339                     {

```

```

337         arrow = true;
338         monI = monI - 1;
339         if (monI < 0) {
340             monI = 11;
341             yearI = yearI - 1;
342         }
343         monthPanel.removeAll();
344         JLabel monthLabel = new JLabel(" " +
345             monthNames[monI] + " " + dayS + ", " + yearI);
346         monthLabel.setFont(new Font(" ", Font.BOLD, 14));
347         monthLabel.setBackground(Color.WHITE);
348         monthPanel.add(monthLabel);
349
350         //buttons
351         monthPanel.add(left, BorderLayout.LINE_START);
352         monthPanel.add(right, BorderLayout.LINE_END);
353         monthPanel.repaint();
354         monthPanel.revalidate();
355         frame.add(createWestGUI(), BorderLayout.LINE_START);
356         monthPanel.add(monthLabel);
357     }
358 });
359
360 //action listener right
361 right.addActionListener(new ActionListener()
362 {
363     public void actionPerformed(ActionEvent e)
364     {
365         arrow = true;
366         monI = monI + 1;
367         if (monI > 11) {
368             monI = 0;
369             yearI = yearI + 1;
370         }
371         monthPanel.removeAll();
372         JLabel monthLabel = new JLabel(" " +
373             monthNames[monI] + " " + dayS + ", " + yearI);
374         monthLabel.setFont(new Font(" ", Font.BOLD, 14));
375         monthLabel.setBackground(Color.WHITE);
376         monthPanel.add(monthLabel);
377
378         //buttons
379         monthPanel.add(left, BorderLayout.LINE_START);
380         monthPanel.add(right, BorderLayout.LINE_END);
381         monthPanel.repaint();
382         monthPanel.revalidate();
383         frame.add(createWestGUI(), BorderLayout.LINE_START);
384         monthPanel.add(monthLabel);
385     }
386 });
387
388 //reset month panel
389 monthPanel.removeAll();
390 JLabel monthLabel = new JLabel(" " + monthNames[monI]
391     + " " + dayS + ", " + yearI);
392 monthLabel.setFont(new Font(" ", Font.BOLD, 14));
393 monthLabel.setBackground(Color.WHITE);
394 monthPanel.add(monthLabel);
395
396 //buttons
397 monthPanel.add(left, BorderLayout.LINE_START);
398 monthPanel.add(right, BorderLayout.LINE_END);
399 monthPanel.repaint();
400 monthPanel.revalidate();
401 }
402
403 });
404 }

```

```

403         dayPanel.repaint();
404         dayPanel.revalidate();
405
406         return dayPanel;
407     }
408
409
410     public static void main(String[] args) {
411         javax.swing.SwingUtilities.invokeLater(new Runnable() {
412             public void run() {
413                 createFrame();
414             }
415         });
416     }
417 }
418
419
420 public class dateInfo {
421     private static String id;
422     public static String info;
423
424     //create a dateInfo object with given values
425     public dateInfo(String identification) {
426         setId(identification);
427         //System.out.println("made date with id " + getId());
428     }
429
430     public dateInfo(String identification, String information) {
431         info = information;
432         //System.out.println("id = " + getId());
433         //System.out.println("saved " + info);
434     }
435
436     public String getInfo() {
437         //System.out.println("returned info");
438         return info;
439     }
440
441     public static String getId() {
442         return id;
443     }
444
445     public static void setId(String id) {
446         dateInfo.id = id;
447     }
448
449     public void setInfo(String information) {
450         info = information;
451         //System.out.println("id = " + getId());
452         //System.out.println("saved " + info);
453     }
454 }
455
456
457

```