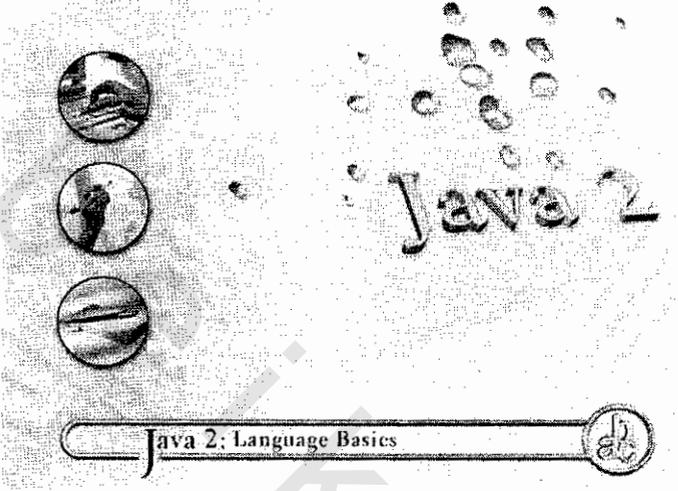


الفصل العشرون



في هذا الفصل نتعرف علي العديد من التطبيقات الهامة للغة Java حيث نتعرف علي العديد من تطبيقات النصوص والتواريخ والمجموعات .Collections

تطبيقات النصوص والتواريخ والمجموعات
String, Date, and Collections Applications

obeyikan.com

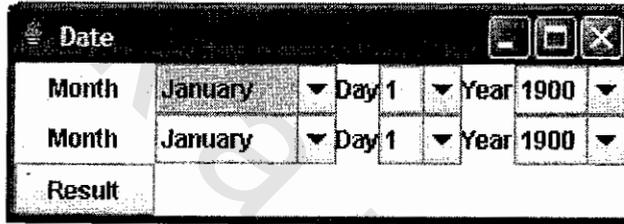
مقدمة:

نتعلم في هذا الفصل مشاريع النصوص والتواريخ والمجموعات Collections.

مثال (1): التعامل مع التواريخ:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال تاريخين في أداة إطار JFrame كما هو واضح في الشكل (1-20) ثم يقوم بطباعة التاريخ الأحدث ، كما يجب أن يتعامل البرنامج مع السنوات الكبيسة والأشهر ذات الثلاثين يوماً وذات الحادي والثلاثين يوماً.



الشكل (1-20) تنفيذ البرنامج

ثانياً: كود البرمجة:

```

1: import javax.swing.*;
2: import java.awt.*;
3: import java.awt.event.*;
4:
5: public class MyDate extends JFrame
6: {
7:     public static final String []month28 =
8:     {"1","2","3","4","5","6","7","8","9","10","11","12","13",
9:     "14","15","16",
10:    "17","18","19","20","21","22","23","24","25","26","27",
11:    "28"};

```

```

9:
10: public static final String []month29 =
    {"1","2","3","4","5","6","7","8","9","10","11","12","13",
    "14","15","16",
11: "17","18","19","20","21","22","23","24","25","26","2
    7","28","29"};
12:
13: public static final String []month30 =
    {"1","2","3","4","5","6","7","8","9","10","11","12","13",
    "14","15","16",
14: "17","18","19","20","21","22","23","24","25","26","2
    7","28","29","30"};
15:
16: public static final String []month31 =
    {"1","2","3","4","5","6","7","8","9","10","11","12","13",
    "14","15","16",
17: "17","18","19","20","21","22","23","24","25","26","2
    7","28","29","30","31"};
18:
19: public static final String []monthNames =
    {"January","February","March","April","May","June",
20: "July","August","September","October","November",
    "December"};
21:
22: public static final int startYear = 1900;
23: public static final int endYear = 3000;
24:
25: public static String[] yearValues;
26:
27: JLabel
    labelMonth1,labelDay1,labelYear1,labelMonth2,labelID
    ay2,labelYear2;
28: JComboBox
    comboMonth1,comboDay1,comboYear1,comboMonth

```

```

        2,comboDay2,comboYear2;
29: JButton result;
30: GridBagLayout grid;
31: GridBagConstraints gbc;
32:
33: MyDate()
34: {
35:     addWindowListener(new WindowAdapter()
36:     {
37:         public void windowClosing(WindowEvent
we)
38:         {
39:             dispose();
40:             System.exit(0);
41:         }
42:     }
43: );
44:
45:     yearValues = new String[endYear - startYear + 1];
46:     for ( int i = 0 ; i <= endYear - startYear ; i++)
47:     {
48:         yearValues[i] = Integer.toString(startYear + i);
49:     }
50:
51:     Container c = getContentPane();
52:
53:     grid = new GridBagLayout();
54:     gbc = new GridBagConstraints();
55:     c.setLayout(grid);
56:
57:     labelMonth1 = new JLabel("Month");
58:     labelDay1 = new JLabel("Day");
59:     labelYear1 = new JLabel("Year");
60:     labelMonth2 = new JLabel("Month");

```

```
61:    labelDay2 = new JLabel("Day");
62:    labelYear2 = new JLabel("Year");
63:
64:    MyComboListener myComboListener = new
    MyComboListener();
65:
66:    comboMonth1 = new JComboBox(monthNames);
67:
68:    comboMonth1.addItemListener(myComboListener);
69:    comboDay1 = new JComboBox(month31);
70:    comboDay1.addItemListener(myComboListener);
71:    comboYear1 = new JComboBox(yearValues);
72:    comboYear1.addItemListener(myComboListener);
73:    comboMonth2 = new JComboBox(monthNames);
74:
75:    comboMonth2.addItemListener(myComboListener);
76:    comboDay2 = new JComboBox(month31);
77:    comboDay2.addItemListener(myComboListener);
78:    comboYear2 = new JComboBox(yearValues);
79:    comboYear2.addItemListener(myComboListener);
80:
81:    ButtonListener b = new ButtonListener();
82:
83:    result = new JButton("Result");
84:    result.addActionListener(b);
85:
86:    gbc.gridx = 1;
87:    gbc.gridy = 1;
88:    grid.setConstraints(labelMonth1,gbc);
89:    c.add(labelMonth1);
90:
91:    gbc.gridx = 2;
92:    gbc.gridy = 1;
93:    grid.setConstraints(comboMonth1,gbc);
```

```
92:    c.add(comboMonth1);
93:
94:    gbc.gridx = 3;
95:    gbc.gridy = 1;
96:    grid.setConstraints(labelDay1,gbc);
97:    c.add(labelDay1);
98:
99:    gbc.gridx = 4;
100:    gbc.gridy = 1;
101:    grid.setConstraints(comboDay1,gbc);
102:    c.add(comboDay1);
103:
104:    gbc.gridx = 5;
105:    gbc.gridy = 1;
106:    grid.setConstraints(labelYear1,gbc);
107:    c.add(labelYear1);
108:
109:    gbc.gridx = 6;
110:    gbc.gridy = 1;
111:    grid.setConstraints(comboYear1,gbc);
112:    c.add(comboYear1);
113:
114:    gbc.gridx = 1;
115:    gbc.gridy = 2;
116:    grid.setConstraints(labelMonth2,gbc);
117:    c.add(labelMonth2);
118:
119:    gbc.gridx = 2;
120:    gbc.gridy = 2;
121:    grid.setConstraints(comboMonth2,gbc);
122:    c.add(comboMonth2);
123:
124:    gbc.gridx = 3;
125:    gbc.gridy = 2;
```

```
126:         grid.setConstraints(labelDay2,gbc);
127:         c.add(labelDay2);
128:
129:         gbc.gridx = 4;
130:         gbc.gridy = 2;
131:         grid.setConstraints(comboDay2,gbc);
132:         c.add(comboDay2);
133:
134:         gbc.gridx = 5;
135:         gbc.gridy = 2;
136:         grid.setConstraints(labelYear2,gbc);
137:         c.add(labelYear2);
138:
139:         gbc.gridx = 6;
140:         gbc.gridy = 2;
141:         grid.setConstraints(comboYear2,gbc);
142:         c.add(comboYear2);
143:
144:         gbc.gridx = 1;
145:         gbc.gridy = 3;
146:         grid.setConstraints(result,gbc);
147:         c.add(result);
148:
149:         setTitle("Date");
150:         pack();
151:         setVisible(true);
152:     }
153:
154:     public static void main(String args[])
155:     {
156:         MyDate myDate = new MyDate();
157:     }
158:
159:     class ButtonListener implements ActionListener
```

```

160:  {
161:      public void actionPerformed(ActionEvent e)
162:      {
163:          Object obj = e.getSource();
164:
165:          String strmonth1 = "";
166:          int month1 = 0;
167:          int day1 = 0;
168:          int year1 = 0;
169:          String strmonth2 = "";
170:          int month2 = 0;
171:          int day2 = 0;
172:          int year2 = 0;
173:
174:          try
175:          {
176:              strmonth1 =
                (String)comboMonth1.getSelectedItemAt();
177:              month1 =
                comboMonth1.getMonthOrder(strmonth1);
178:              day1 =
                Integer.parseInt((String)comboDay1.getSelectedItemAt());
                ;
179:              year1 =
                Integer.parseInt((String)comboYear1.getSelectedItemAt());
                );
180:              strmonth2 =
                (String)comboMonth2.getSelectedItemAt();
181:              month2 =
                comboMonth2.getMonthOrder(strmonth2);
182:              day2 =
                Integer.parseInt((String)comboDay2.getSelectedItemAt());
                ;
183:              year2 =

```

```
Integer.parseInt((String)comboYear2.getSelectedItem()
);
184:         }
185:
186:         catch(Exception ex)
187:         {
188:
189:         JOptionPane.showMessageDialog(MyDate.this,"Wron
g Date Values");
190:         return;
191:     }
192:     if ( obj == result )
193:     {
194:         int newMonth = 0;
195:         int newDay = 0;
196:         int newYear = 0;
197:
198:         if ( year1 > year2 )
199:         {
200:             newMonth = month1;
201:             newDay = day1;
202:             newYear = year1;
203:         }
204:
205:         else if ( year1 < year2 )
206:         {
207:             newMonth = month2;
208:             newDay = day2;
209:             newYear = year2;
210:         }
211:
212:         else if ( year1 == year2 )
213:         {
```

```
214:         if ( month1 > month2 )
215:         {
216:             newMonth = month1;
217:             newDay = day1;
218:             newYear = year1;
219:         }
220:
221:         else if ( month1 < month2 )
222:         {
223:             newMonth = month2;
224:             newDay = day2;
225:             newYear = year2;
226:         }
227:
228:         else if ( month1 == month2 )
229:         {
230:             if ( day1 > day2 )
231:             {
232:                 newMonth = month1;
233:                 newDay = day1;
234:                 newYear = year1;
235:             }
236:
237:             else if ( day1 < day2 )
238:             {
239:                 newMonth = month2;
240:                 newDay = day2;
241:                 newYear = year2;
242:             }
243:
244:             else if ( day1 == day2 )
245:             {
246:                 newMonth = month1;
247:                 newDay = day1;
```

```
248:             newYear = year1;
249:             }
250:         }
251:     }
252:
253:         String message = "New Date is " +
newDay + "/" + newMonth + "/" + newYear;
254:
JOOptionPane.showMessageDialog(MyDate.this, messa
ge);
255:     }
256:
257: }
258: }
259:
260: public int getMonthOrder(String monthName)
261: {
262:     int order = 1;
263:
264:     for ( int i = 0 ; i < monthNames.length ; i++ )
265:     {
266:         if ( monthNames[i].equals(monthName) )
267:         {
268:             order = i + 1;
269:         }
270:     }
271:
272:     return order;
273: }
274:
275: class MyComboListener implements ItemListener
276: {
277:     public void itemStateChanged(ItemEvent e)
278:     {
```

```
279:         Object source = e.getSource();
280:
281:         if ( source == comboDay1 || source ==
            comboDay2 )
282:             {
283:                 return;
284:             }
285:
286:         else if ( source == comboMonth1 || source
            == comboYear1 )
287:             {
288:                 updateComboValues(comboDay1,comboMonth1,com
                    boYear1);
289:             }
290:
291:         else if ( source == comboMonth2 || source
            == comboYear2 )
292:             {
293:                 updateComboValues(comboDay2,comboMonth2,com
                    boYear2);
294:             }
295:         }
296:     }
297:
298:     public void updateComboValues(JComboBox
        comboDay,JComboBox        comboMonth,JComboBox
        comboYear)
299:     {
300:         try
301:         {
302:             String        strmonth        =
                (String)comboMonth.getSelectedItem();
```

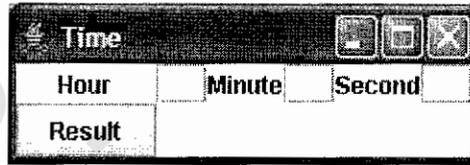
```
303:         int month = getMonthOrder(strmonth);
304:
305:         if ( month == 1 || month == 3 || month == 5
           || month == 7 || month == 8 || month == 10 || month ==
           12 )
306:             {
307:                 comboDay.removeAllItems();
308:
309:                 for ( int i = 0 ; i < month31.length ;
           i++ )
310:                     {
311:                         comboDay.addItem(month31[i]);
312:                     }
313:             }
314:
315:         else if ( month == 4 || month == 6 || month
           == 9 || month == 11 )
316:             {
317:                 comboDay.removeAllItems();
318:
319:                 for ( int i = 0 ; i < month30.length ;
           i++ )
320:                     {
321:                         comboDay.addItem(month30[i]);
322:                     }
323:             }
324:
325:         else if ( month == 2 )
326:             {
327:                 comboDay.removeAllItems();
328:
329:                 int         year         =
           Integer.parseInt((String)comboYear.getSelectedItem());
330:
```

```
331:         if ( year % 4 == 0 )
332:             {
333:                 for ( int i = 0 ; i < month29.length ;
334:                     i++ )
335:                     {
336:                         comboDay.addItem(month29[i]);
337:                     }
338:             }
339:         else
340:             {
341:                 for ( int i = 0 ; i < month28.length ;
342:                     i++ )
343:                     {
344:                         comboDay.addItem(month28[i]);
345:                     }
346:             }
347:     }
348:
349:     catch(Exception e)
350:     {
351:         return;
352:     }
353: }
354: }
```

مثال (2): التعامل مع الوقت:

أولاً: هدف المثال:

اكتب برنامجاً يقوم بإدخال الزمن بالساعة والدقيقة والثانية في أداة إطار JFrame كما هو واضح في الشكل (2-20) ثم يقوم بإضافة ثانية واحدة علي هذا الزمن ثم طباعة الزمن الجديد.



الشكل (2-20) تنفيذ البرنامج

ثانياً: كود البرمجة:

```

1: import javax.swing.*;
2: import java.awt.*;
3: import java.awt.event.*;
4: import java.util.*;
5:
6: public class MyTime extends JFrame
7: {
8:     JLabel labelHour, labelMinute, labelSecond;
9:     JTextField textHour, textMinute, textSecond;
10:    JButton result;
11:    GridBagLayout grid;
12:    GridBagConstraints gbc;
13:
14:    MyTime()
15:    {
16:        addWindowListener(new WindowAdapter()
17:        {
18:            public void windowClosing(WindowEvent
19:            we)

```

```
20:         dispose();
21:         System.exit(0);
22:     }
23: }
24: );
25:
26:     Container c = getContentPane();
27:
28:     grid = new GridBagLayout();
29:     gbc = new GridBagConstraints();
30:     c.setLayout(grid);
31:
32:     labelHour = new JLabel("Hour");
33:     labelMinute = new JLabel("Minute");
34:     labelSecond = new JLabel("Second");
35:
36:     textHour = new JTextField(2);
37:     textMinute = new JTextField(2);
38:     textSecond = new JTextField(2);
39:
40:     ButtonListener b = new ButtonListener();
41:
42:     result = new JButton("Result");
43:     result.addActionListener(b);
44:
45:     gbc.gridx = 1;
46:     gbc.gridy = 1;
47:     grid.setConstraints(labelHour,gbc);
48:     c.add(labelHour);
49:
50:     gbc.gridx = 2;
51:     gbc.gridy = 1;
52:     grid.setConstraints(textHour,gbc);
53:     c.add(textHour);
54:
```

```
55:    gbc.gridx = 3;
56:    gbc.gridy = 1;
57:    grid.setConstraints(labelMinute,gbc);
58:    c.add(labelMinute);
59:
60:    gbc.gridx = 4;
61:    gbc.gridy = 1;
62:    grid.setConstraints(textMinute,gbc);
63:    c.add(textMinute);
64:
65:    gbc.gridx = 5;
66:    gbc.gridy = 1;
67:    grid.setConstraints(labelSecond,gbc);
68:    c.add(labelSecond);
69:
70:    gbc.gridx = 6;
71:    gbc.gridy = 1;
72:    grid.setConstraints(textSecond,gbc);
73:    c.add(textSecond);
74:
75:    gbc.gridx = 1;
76:    gbc.gridy = 2;
77:    grid.setConstraints(result,gbc);
78:    c.add(result);
79:
80:    setTitle("Time");
81:    pack();
82:    setVisible(true);
83: }
84:
85: public static void main(String args[])
86: {
87:     MyTime myTime = new MyTime();
88: }
89:
```

```

90: class ButtonListener implements ActionListener
91: {
92:     public void actionPerformed(ActionEvent e)
93:     {
94:         Object obj = e.getSource();
95:
96:         int hour = 0;
97:         int minute = 0;
98:         int second = 0;
99:
100:        try
101:        {
102:            hour =
Integer.parseInt(textHour.getText());
103:            minute =
Integer.parseInt(textMinute.getText());
104:            second =
Integer.parseInt(textSecond.getText());
105:        }
106:
107:        catch(Exception ex)
108:        {
109:
JOptionPane.showMessageDialog(MyTime.this,"Wro
ng Time Values");
110:            return;
111:        }
112:
113:        if ( obj == result )
114:        {
115:            Calendar time =
Calendar.getInstance();
116:            time.set(2006,8,30,hour,minute,second);
117:            time.add(Calendar.SECOND,1);

```

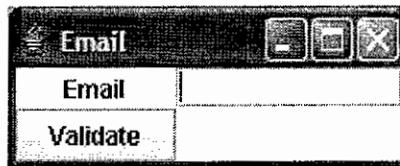
```

118:
119:         int             newHour             =
           time.get(Calendar.HOUR);
120:         int             newMinute           =
           time.get(Calendar.MINUTE);
121:         int             newSecond          =
           time.get(Calendar.SECOND);
122:
123:         String message = "New Time is " +
           newHour + ":" + newMinute + ":" + newSecond;
124:         JOptionPane.showMessageDialog(MyTime.this, messa
           ge);
125:     }
126: }
127: }
128: }
    
```

مثال (3): التعامل مع النصوص:

أولاً: هدف المثال:

في هذا المثال نريد إنشاء برنامج يحتوي علي أداة إطار JFrame كما هو واضح في الشكل (20-3) ، وعلي المستخدم إدخال بريده الإلكتروني E-mail ، ونريد التأكد من أن بريده الإلكتروني E-mail يحتوي علي الحرف @ وأنه ينتهي بالحروف (.com) ، وبعد التأكد من صحة إدخال المستخدم ، فإننا نظهر رسالة لنبين اسم المستخدم User Name واسم الشركة Company Name.



الشكل (20-3) تنفيذ البرنامج

ثانياً: كود البرمجة:

```
1: import javax.swing.*;
2: import java.awt.*;
3: import java.awt.event.*;
4:
5: public class MyEmail extends JFrame
6: {
7:     JLabel labelEmail;
8:     JTextField textEmail;
9:     JButton validate;
10:    GridBagLayout grid;
11:    GridBagConstraints gbc;
12:
13:    MyEmail()
14:    {
15:        addWindowListener(new WindowAdapter()
16:            {
17:                public void windowClosing(WindowEvent
18:                we)
19:                {
20:                    dispose();
21:                    System.exit(0);
22:                }
23:            }
24:        );
25:        Container c = getContentPane();
26:
27:        grid = new GridBagLayout();
28:        gbc = new GridBagConstraints();
29:        c.setLayout(grid);
30:
31:        labelEmail = new JLabel("Email");
32:
```

```
33:    textEmail = new JTextField(10);
34:
35:    textEmail.addKeyListener(new KeyAdapter()
36:    {
37:        public void keyReleased(KeyEvent e)
38:        {
39:            char c = e.getKeyChar();
40:
41:            if ( c == '\n' )
42:            {
43:                validate.doClick();
44:            }
45:        }
46:    }
47: );
48:
49:    ButtonListener b = new ButtonListener();
50:
51:    validate = new JButton("Validate");
52:    validate.addActionListener(b);
53:
54:    gbc.gridx = 1;
55:    gbc.gridy = 1;
56:    grid.setConstraints(labelEmail,gbc);
57:    c.add(labelEmail);
58:
59:    gbc.gridx = 2;
60:    gbc.gridy = 1;
61:    grid.setConstraints(textEmail,gbc);
62:    c.add(textEmail);
63:
64:    gbc.gridx = 1;
65:    gbc.gridy = 2;
66:    grid.setConstraints(validate,gbc);
```

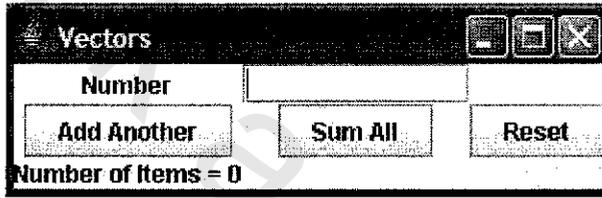
```
67:     c.add(validate);
68:
69:     setTitle("Email");
70:     pack();
71:     setVisible(true);
72: }
73:
74: public static void main(String args[])
75: {
76:     MyEmail myEmail = new MyEmail();
77: }
78:
79: class ButtonListener implements ActionListener
80: {
81:     public void actionPerformed(ActionEvent e)
82:     {
83:         Object obj = e.getSource();
84:
85:         if ( obj == validate )
86:         {
87:             String email = textEmail.getText();
88:
89:             if ( email == null || email.trim().equals("")
90:             )
91:             {
92:                 JOptionPane.showMessageDialog(MyEmail.this,"Plea
93: se enter your email");
94:                 textEmail.selectAll();
95:                 textEmail.requestFocus();
96:                 return;
97:             }
98:
99:             if ( email.indexOf("@",0) < 0 )
```

```
98:         {
99:             JOptionPane.showMessageDialog(MyEmail.this,"The
                email should contain '@' character");
100:                 textEmail.selectAll();
101:                 textEmail.requestFocus();
102:                 return;
103:             }
104:
105:             else if ( email.lastIndexOf(".com",20)
                < 0 )
106:                 {
107:                     JOptionPane.showMessageDialog(MyEmail.this,"The
                        email should end with .com");
108:                     textEmail.selectAll();
109:                     textEmail.requestFocus();
110:                     return;
111:                 }
112:
113:             else
114:                 {
115:                     String      username      =
                        email.substring(0,email.indexOf("@",0));
116:                     String      companyname    =
                        email.substring(email.indexOf("@",0)
                            +
                            1,email.length());
117:                     JOptionPane.showMessageDialog(MyEmail.this,"Use
                        r Name: " + username + "\n" + "Company Name: " +
                        companyname);
118:                 }
119:             }
120:         }
121:     }
122: }
```

مثال (4): التعامل مع المجموعات Collections:

أولاً: هدف المثال:

في هذا المثال نريد إنشاء برنامج يحتوي علي أداة إطار JFrame كما هو واضح في الشكل (20-4) ، بحيث يقوم المستخدم بإدخال رقم ثم يضغط علي الزر Add Another ليقوم البرنامج بحفظ الرقم في الذاكرة ، ويظل المستخدم يدخل الأرقام ويقوم البرنامج بحفظ جميع الأرقام الجديدة ويوضح عدد الأرقام الموجود في الذاكرة ، فإذا رغب المستخدم في جمع الأرقام الموجودة في الذاكرة فإنه يضغط علي الزر Sum All ، أما إذا رغب في مسح الموجود والبدء من جديد فإنه يضغط علي الزر Reset.



الشكل (20-4) تنفيذ البرنامج

ثانياً: كود البرمجة:

```

1: import javax.swing.*;
2: import java.awt.*;
3: import java.awt.event.*;
4: import java.util.*;
5:
6: public class MyNumbers extends JFrame
7: {
8:     JLabel labelNum,labelNumberOfNumbers;
9:     JTextField textNum;
10:    JButton addAnother,sumAll,reset;
11:    Vector vectorNumbers = new Vector();
12:    GridBagLayout grid;
13:    GridBagConstraints gbc;
14:

```

```
15: MyNumbers()
16: {
17:     addWindowListener(new WindowAdapter()
18:         {
19:             public void windowClosing(WindowEvent
we)
20:                 {
21:                     dispose();
22:                     System.exit(0);
23:                 }
24:     }
25: );
26:
27:     Container c = getContentPane();
28:
29:     grid = new GridBagLayout();
30:     gbc = new GridBagConstraints();
31:     c.setLayout(grid);
32:
33:     labelNum = new JLabel("Number");
34:     labelNumberOfNumbers = new JLabel("Number
of Items = 0");
35:
36:     textNum = new JTextField(10);
37:
38:     ButtonListener b = new ButtonListener();
39:
40:     addAnother = new JButton("Add Another");
41:     addAnother.addActionListener(b);
42:     sumAll = new JButton("Sum All");
43:     sumAll.addActionListener(b);
44:     reset = new JButton("Reset");
45:     reset.addActionListener(b);
46:
```

```
47:    gbc.gridx = 1;
48:    gbc.gridy = 1;
49:    grid.setConstraints(labelNum,gbc);
50:    c.add(labelNum);
51:
52:    gbc.gridx = 2;
53:    gbc.gridy = 1;
54:    grid.setConstraints(textNum,gbc);
55:    c.add(textNum);
56:
57:    gbc.gridx = 1;
58:    gbc.gridy = 2;
59:    grid.setConstraints(addAnother,gbc);
60:    c.add(addAnother);
61:
62:    gbc.gridx = 2;
63:    gbc.gridy = 2;
64:    grid.setConstraints(sumAll,gbc);
65:    c.add(sumAll);
66:
67:    gbc.gridx = 3;
68:    gbc.gridy = 2;
69:    grid.setConstraints(reset,gbc);
70:    c.add(reset);
71:
72:    gbc.gridx = 1;
73:    gbc.gridy = 3;
74:    grid.setConstraints(labelNumberOfNumbers,gbc);
75:    c.add(labelNumberOfNumbers);
76:
77:    setTitle("Vectors");
78:    pack();
79:    setVisible(true);
80: }
```

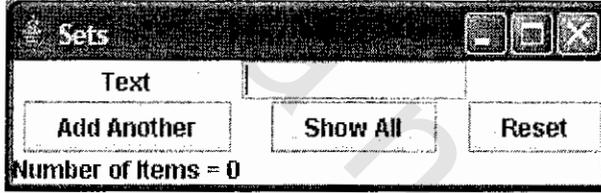
```
81:
82: public static void main(String args[])
83: {
84:     MyNumbers myNumbers = new MyNumbers();
85: }
86:
87: class ButtonListener implements ActionListener
88: {
89:     public void actionPerformed(ActionEvent e)
90:     {
91:         Object obj = e.getSource();
92:
93:         if ( obj == reset )
94:         {
95:             vectorNumbers.clear();
96:             labelNumberOfNumbers.setText("Number
of Items = " + vectorNumbers.size());
97:         }
98:
99:         else if ( obj == sumAll )
100:        {
101:            double sum = 0;
102:            for ( int i = 0 ; i <
vectorNumbers.size() ; i++ )
103:            {
104:                double n =
((Double)vectorNumbers.get(i)).doubleValue();
105:                sum += n;
106:            }
107:
108:
109:            JOptionPane.showMessageDialog(MyNumbers.this,"S
um = " + sum);
109:        }
```

```
110:
111:         if ( obj == addAnother )
112:         {
113:             double num = 0;
114:
115:             try
116:             {
117:                 num =
Double.parseDouble(textNum.getText());
118:             }
119:
120:             catch(Exception ex)
121:             {
122:
JOptionPane.showMessageDialog(MyNumbers.this,"P
lease Enter a Number");
123:                 textNum.selectAll();
124:                 textNum.requestFocus();
125:                 return;
126:             }
127:
128:             Double d = new Double(num);
129:             vectorNumbers.add(d);
130:
labelNumberOfNumbers.setText("Number of Items =
" + vectorNumbers.size());
131:                 textNum.setText("");
132:                 textNum.requestFocus();
133:             }
134:         }
135:     }
136: }
```

مثال (5): طباعة القيم الفريدة Unique:

أولاً: هدف المثال:

في هذا المثال نريد إنشاء برنامج يحتوي علي أداة إطار JFrame كما هو واضح في الشكل (20-5) ، بحيث يقوم المستخدم بإدخال نص ثم يضغط علي الزر Add Another ليقوم البرنامج بحفظ النص في الذاكرة بشرط ألا يكون قد سبق للمستخدم إدخال هذا النص من قبل ، وفي هذه الحالة يطبع البرنامج رسالة خطأ للمستخدم ، ويظل المستخدم يدخل النصوص ويقوم البرنامج بحفظ جميع النصوص الجديدة ويوضح عدد النصوص الموجود في الذاكرة ، فإذا رغب المستخدم في عرض النصوص الموجودة في الذاكرة فإنه يضغط علي الزر Show All ، أما إذا رغب في مسح الموجود والبدء من جديد فإنه يضغط علي الزر Reset.



الشكل (20-5) تنفيذ البرنامج

ثانياً: كود البرمجة:

```

1: import javax.swing.*;
2: import java.awt.*;
3: import java.awt.event.*;
4: import java.util.*;
5:
6: public class MyStrings extends JFrame
7: {
8:     JLabel labelString,labelNumberOfNumbers;
9:     JTextField text;
10:    JButton addAnother,showAll,reset;
11:    TreeSet set = new TreeSet();

```

```
12: GridBagLayout grid;
13: GridBagConstraints gbc;
14:
15: MyStrings()
16: {
17:     addWindowListener(new WindowAdapter()
18:     {
19:         public void windowClosing(WindowEvent
we)
20:         {
21:             dispose();
22:             System.exit(0);
23:         }
24:     }
25: );
26:
27: Container c = getContentPane();
28:
29: grid = new GridBagLayout();
30: gbc = new GridBagConstraints();
31: c.setLayout(grid);
32:
33: labelString = new JLabel("Text");
34: labelNumberOfNumbers = new JLabel("Number
of Items = 0");
35:
36: text = new JTextField(10);
37:
38: ButtonListener b = new ButtonListener();
39:
40: addAnother = new JButton("Add Another");
41: addAnother.addActionListener(b);
42: showAll = new JButton("Show All");
43: showAll.addActionListener(b);
44: reset = new JButton("Reset");
```

```
45:    reset.addActionListener(b);
46:
47:    gbc.gridx = 1;
48:    gbc.gridy = 1;
49:    grid.setConstraints(labelString,gbc);
50:    c.add(labelString);
51:
52:    gbc.gridx = 2;
53:    gbc.gridy = 1;
54:    grid.setConstraints(text,gbc);
55:    c.add(text);
56:
57:    gbc.gridx = 1;
58:    gbc.gridy = 2;
59:    grid.setConstraints(addAnother,gbc);
60:    c.add(addAnother);
61:
62:    gbc.gridx = 2;
63:    gbc.gridy = 2;
64:    grid.setConstraints(showAll,gbc);
65:    c.add(showAll);
66:
67:    gbc.gridx = 3;
68:    gbc.gridy = 2;
69:    grid.setConstraints(reset,gbc);
70:    c.add(reset);
71:
72:    gbc.gridx = 1;
73:    gbc.gridy = 3;
74:    grid.setConstraints(labelNumberOfNumbers,gbc);
75:    c.add(labelNumberOfNumbers);
76:
77:    setTitle("Sets");
78:    pack();
```

```
79:     setVisible(true);
80: }
81:
82: public static void main(String args[])
83: {
84:     MyStrings myStrings = new MyStrings();
85: }
86:
87: class ButtonListener implements ActionListener
88: {
89:     public void actionPerformed(ActionEvent e)
90:     {
91:         Object obj = e.getSource();
92:
93:         if ( obj == reset )
94:         {
95:             set.clear();
96:             labelNumberOfNumbers.setText("Number
of Items = " + set.size());
97:         }
98:
99:         else if ( obj == showAll )
100:        {
101:            String message = "";
102:
103:            Iterator iterator = set.iterator();
104:
105:            while ( iterator.hasNext() )
106:            {
107:                message += iterator.next() + "\n";
108:            }
109:
110:            JOptionPane.showMessageDialog(MyStrings.this,mes
```

```

    sage);
111:         }
112:
113:         if ( obj == addAnother )
114:         {
115:             String inputText = text.getText();
116:
117:             if ( set.contains(inputText) )
118:             {
119:
120:                 JOptionPane.showMessageDialog(MyStrings.this,"This
                s text was entered before\nand will not be added");
121:                 return;
122:             }
123:             set.add(inputText);
124:
125:             labelNumberOfNumbers.setText("Number of Items =
            " + set.size());
126:             text.setText("");
127:             text.requestFocus();
128:         }
129:     }
130: }

```

ملخص الفصل:

تعلمنا في هذا الفصل العديد من تطبيقات النصوص والتواريخ والمجموعات
Collections.

في الفصل القادم نتعلم - بإذن الله - العديد من التطبيقات الأخرى ، فتابع معنا
الفصل القادم.