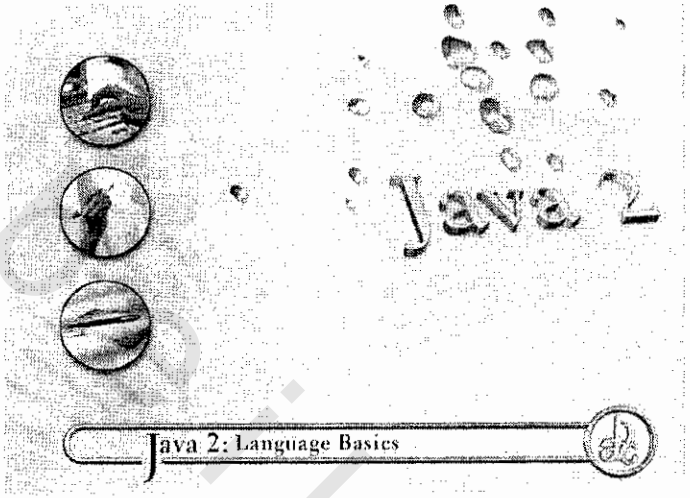
الفصل السادس عشر



مدير التخطيط Layout Manager

في هذا الفصل سوف نتناول كيفية تصميم الواجهة الرسومية GUI بطريقة متقدمة بحيث يمكننا وضع أكثر من أداة في الواجهة الرسومية GUI ويتم ذلك عن طريق ما يعرف باسم مدير التخطيط Layout Manager.

سنتناول في هذا الفصل أشهر وأكثر مديري التخطيط Layout Managers استعمالاً وهم:

- مدير التخطيط FlowLayout.
- مدير التخطيط GridLayout.
- مدير التخطيط BorderLayout.
- مدير التخطيط GridbagLayout.

مقدمة:

- تناولنا فى الفصل السابق عدداً كبيراً من مجموعة الأدوات المعروفة باسم أدوات مكتبة Swing ، ولاحظنا أننا فى كل مرة لا نستطيع وضع أكثر من أداة فى البرنامج الواحد.
- في هذا الفصل سوف نتناول كيفية وضع أكثر من أداة في الواجهة الرسومية GUI للبرنامج وكيفية تحديد موضع الأدوات في الواجهة الرسومية GUI عن طريق ما يعرف باسم مدير التخطيط Layout Manager.
- سوف نتناول مديري التخطيط Layout Managers التاليين:
  1. مدير التخطيط FlowLayout.
  2. مدير التخطيط GridLayout.
  3. مدير التخطيط BorderLayout.
  4. مدير التخطيط GridBagLayout.

مدير التخطيط FlowLayout:

- مدير التخطيط FlowLayout يضع الأدوات controls واحدة تلو الأخري وبنفس ترتيب إضافتهم. بمعنى أنه يضع أول أداة control تضيفها فى الشاشة ، وعند إضافة أداة control أخري فإنه يتم وضعها بجانب الأداة control الأولي وهكذا ، فإذا كان عرض النافذة frame لا يكفي ، فإنه يتم وضع الأداة على سطر جديد وهكذا. أيضاً عند تغيير حجم النافذة frame فإن ترتيب الأدوات controls يختلف فى الشاشة.
- يعتبر مدير التخطيط FlowLayout أبسط مدير تخطيط Layout Manager ولكنه في نفس الوقت لا يتيح لك التحكم الكامل في شكل نافذة البرنامج لأن وضع الأدوات يختلف حسب عرض النافذة frame وحسب عدد الأدوات الموجودة في النافذة.

تتوافر العديد من دوال البناء Constructors لهذه الفصيلة class كما يلي.

دالة البناء FlowLayout():

تقوم بإظهار الأدوات في منتصف نافذة البرنامج مع ترك مسافة أفقية ورأسية بين كل أداة والتي تليها تساوي 5 بكسل pixel.

دالة البناء FlowLayout(int align):

تقوم بإظهار الأدوات في نافذة البرنامج حسب المحاذاة Alignment المحددة في المعامل Parameter مع ترك مسافة أفقية ورأسية بين كل أداة والتي تليها تساوي 5 بكسل pixel.

المعاملات Parameters:

int align: تمثل المحاذاة Alignment المطلوبة للأدوات. يمكن أن تكون واحدة من القيم التالية:

- FlowLayout.LEFT: تقوم بمحاذاة Alignment الأدوات في يسار نافذة البرنامج.
- FlowLayout.RIGHT: تقوم بمحاذاة Alignment الأدوات في يمين نافذة البرنامج.
- FlowLayout.CENTER: تقوم بمحاذاة Alignment الأدوات في منتصف نافذة البرنامج.

دالة البناء FlowLayout(int align, int hgap, int vgap):

تقوم بإظهار الأدوات في نافذة البرنامج حسب المحاذاة Alignment المحددة في المعامل Parameter مع ترك مسافة أفقية ورأسية بين كل أداة والتي تليها حسب القيم المطلوبة في المعامل Parameter.

المعاملات Parameters:

int align: تمثل المحاذاة Alignment المطلوبة للأدوات. يمكن أن تكون واحدة من القيم التالية:

- FlowLayout.LEFT: تقوم بمحاذاة Alignment الأدوات في يسار نافذة البرنامج.

- FlowLayout.RIGHT: تقوم بمحاذاة Alignment الأدوات في يمين نافذة البرنامج.
- FlowLayout.CENTER: تقوم بمحاذاة Alignment الأدوات في منتصف نافذة البرنامج.

int hgap: المسافة الأفقية المطلوبة بين الأدوات بوحدات البكسل pixel.

int vgap: المسافة الرأسية المطلوبة بين الأدوات بوحدات البكسل pixel.

المثال التالى يوضح مدير التخطيط FlowLayout.

مثال (1): مدير التخطيط FlowLayout:

أولاً: هدف المثال:

توضيح كيفية وضع أكثر من أداة control في نافذة البرنامج باستخدام مدير التخطيط FlowLayout.

ثانياً: كود البرمجة:

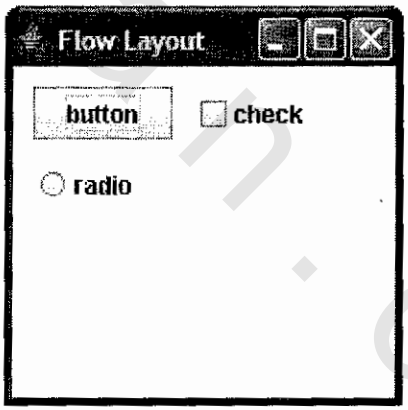
```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class MyFlowLayout extends JFrame
5:  {
6:   JButton b1;
7:   JCheckBox c1;
8:   JRadioButton r1;
9:
10:  MyFlowLayout()
11:  {
12:       Container c = getContentPane();
13:       c.setLayout(new
     FlowLayout(FlowLayout.LEFT,10,10));
14:
```

```
15:       b1 = new JButton("button");
16:       c1 = new JCheckBox("check");
17:       r1 = new JRadioButton ("radio");
18:       c.add(b1);
19:       c.add(c1);
20:       c.add(r1);
21:
22:       setTitle("Flow Layout");
23:       setSize(200,200);
24:
      setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOS
    E);
25:       setVisible(true);
26:  }
27:
28:  public static void main(String args[])
29:  {
30:       MyFlowLayout flow = new MyFlowLayout();
31:  }
32:}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyFlowLayout التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JButton.
- في السطر رقم 7 يتم إنشاء متغير من نوع JCheckBox.
- في السطر رقم 8 يتم إنشاء متغير من نوع JRadioButton.
- في السطر رقم 10 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 13 يتم تحديد FlowLayout ليصبح هو مدير التخطيط Layout Manager لنافذة البرنامج عن طريق الدالة ()setLayout.

- في السطر رقم 15 يتم إنشاء هدف Object من نوع JButton والذي يمثل أداة الزر JButton.
- في السطر رقم 16 يتم إنشاء هدف Object من نوع JCheckBox والذي يمثل أداة صندوق التحقق JCheckBox.
- في السطر رقم 17 يتم إنشاء هدف Object من نوع JRadioButton والذي يمثل أداة زر الراديو JRadioButton.
- في السطور من 28 إلي 31 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyFlowLayout وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-1).

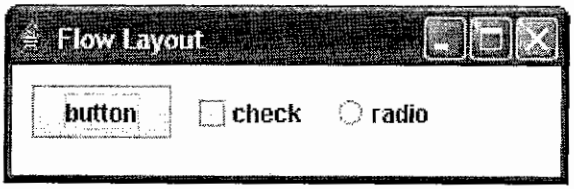


الشكل (16-1) مدير التخطيط FlowLayout

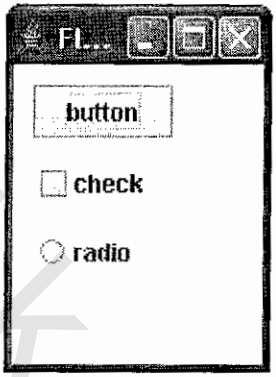
ملحوظات:

يمكنك بالطبع وضع أى أداة control بدلاً من JButton أو JRadioButton أو JCheckBox ولكننا فى الأمثلة القادمة سنضع JButton فقط للتسهيل.

لاحظ أنه عند تغيير حجم النافذة فإن مكان الأدوات controls يتغير كما هو واضح في الشكل (16-2) والشكل (16-3).



الشكل (16-2) مدير التخطيط FlowLayout



الشكل (16-3) مدير التخطيط FlowLayout

مدير التخطيط GridLayout:

- مدير التخطيط GridLayout يجعل النافذة frame مثل الشبكة على هيئة مستطيل يتم تحديد عدد الصفوف والأعمدة الموجودة فيه.
- يتميز GridLayout بأن جميع الأدوات في الشبكة تظهر بنفس الحجم وبالتالي لا يمكن أن يستخدم GridLayout إلا إذا كنت تريد أن تكون جميع الأدوات في نافذة البرنامج بنفس الحجم بالضبط.

تتوافر العديد من دوال البناء Constructors لهذه الفصيلة class كما يلي.

دالة البناء GridLayout():

تقوم بإنشاء شبكة تحتوي علي أداة واحدة فقط أي صف واحد وعمود واحد.

دالة البناء GridLayout(int rows, int cols):

تقوم بإنشاء شبكة تحتوي علي عدد الصفوف والأعمدة المحددين في المعامل Parameter.

المعاملات Parameters:

int rows: يمثل عدد الصفوف المطلوب في الشبكة.

int cols: يمثل عدد الأعمدة المطلوب في الشبكة.

دالة البناء GridLayout(int rows, int cols, int hgap, int vgap):

تقوم بإنشاء شبكة تحتوي علي عدد الصفوف والأعمدة المحددين في المعامل Parameter مع ترك مسافة أفقية ورأسية بين كل أداة والتي تليها حسب القيم المطلوبة في المعامل Parameter.

المعاملات Parameters:

int rows: يمثل عدد الصفوف المطلوب في الشبكة.

int cols: يمثل عدد الأعمدة المطلوب في الشبكة.

int hgap: المسافة الأفقية المطلوبة بين الأدوات بوحدات البكسل pixel.

int vgap: المسافة الرأسية المطلوبة بين الأدوات بوحدات البكسل pixel.

المثال التالى يوضح مدير التخطيط GridLayout.

مثال (2): مدير التخطيط GridLayout:

أولاً: هدف المثال:

توضيح كيفية وضع أكثر من أداة control باستخدام مدير التخطيط GridLayout.

ثانياً: كود البرمجة:

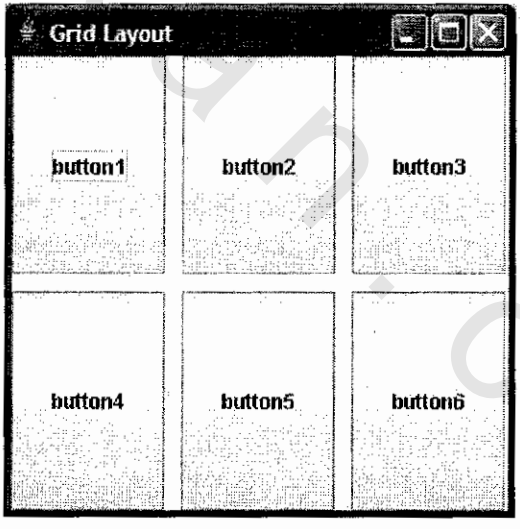
```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class MyGridLayout extends JFrame
5:  {
6:    JButton b1,b2,b3,b4,b5,b6;
7:
8:    MyGridLayout()
9:    {
10:       Container c = getContentPane();
11:       c.setLayout(new GridLayout(2,3,10,10));
```

```
12:
13:       b1 = new JButton("button1");
14:       b2 = new JButton("button2");
15:       b3 = new JButton("button3");
16:       b4 = new JButton("button4");
17:       b5 = new JButton("button5");
18:       b6 = new JButton("button6");
19:
20:       c.add(b1);
21:       c.add(b2);
22:       c.add(b3);
23:       c.add(b4);
24:       c.add(b5);
25:       c.add(b6);
26:
27:       setTitle("Grid Layout");
28:       setSize(300,300);
29:
     setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOS
   E);
30:       setVisible(true);
31:  }
32:
33:  public static void main(String args[])
34:  {
35:       MyGridLayout grid = new MyGridLayout();
36:  }
37:}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyGridLayout التي ترث من الفصيلة JFrame حتى يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.

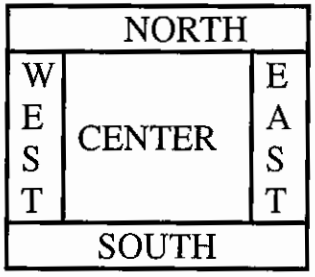
- في السطر رقم 6 يتم إنشاء 6 متغيرات من نوع JButton.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم تحديد GridLayout ليصبح هو مدير التخطيط Layout Manager لنافذة البرنامج عن طريق الدالة setLayout().
- في السطور من 13 إلي 18 يتم إنشاء 6 أهداف Objects من نوع JButton والذي يمثل أداة الزر JButton.
- في السطور من 33 إلي 36 يتم إنشاء الدالة الرئيسية main() التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyGridLayout وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (4-16).



الشكل (16-4) مدير التخطيط GridLayout

مدير التخطيط BorderLayout:

- مدير التخطيط BorderLayout هو مدير تخطيط يقوم بتقسيم النافذة Frame إلى خمسة مناطق كما هو مبين بالشكل التالي.



وعند إضافة أداة Control جديدة فإننا يجب أن نحدد فى أى منطقة سوف نضيف تلك الأداة كما يتضح لنا في المثال التالي.

تتوافر العديد من دوال البناء Constructors لهذه الفصيلة class كما يلي.

دالة البناء BorderLayout():

تقوم بإنشاء نافذة البرنامج بدون وجود أي مسافات بين الأدوات.

دالة البناء BorderLayout(int hgap, int vgap):

تقوم بإنشاء نافذة البرنامج مع ترك مسافة أفقية ورأسية بين كل أداة والتي تليها حسب القيم المطلوبة في المعامل Parameter.

المعاملات Parameters:

int hgap: المسافة الأفقية المطلوبة بين الأدوات بوحدات البكسل pixel.

int vgap: المسافة الرأسية المطلوبة بين الأدوات بوحدات البكسل pixel.

المثال التالى يوضح مدير التخطيط BorderLayout.

مثال (3): مدير التخطيط BorderLayout:

أولاً: هدف المثال:

توضيح كيفية وضع أكثر من أداة control باستخدام مدير التخطيط BorderLayout.

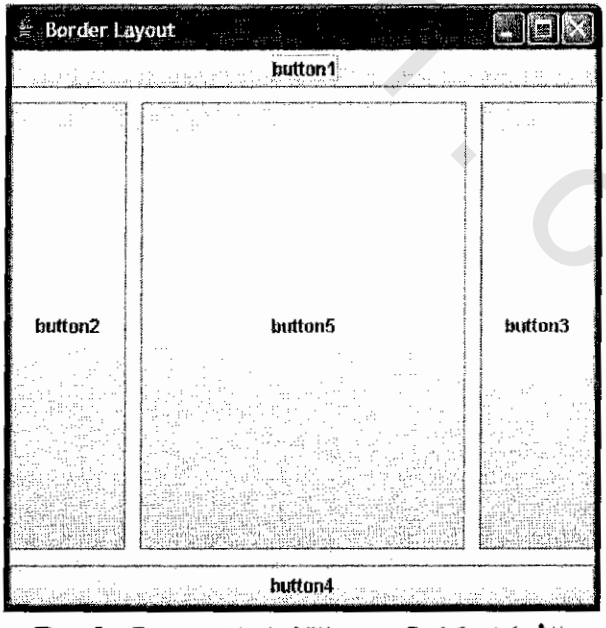
ثانياً: كود البرمجة:

```
1: import javax.swing.*;
2: import java.awt.*;
3:
```

```
4:  public class MyBorderLayout extends JFrame
5:  {
6:   JButton b1,b2,b3,b4,b5;
7:
8:   MyBorderLayout()
9:   {
10:       Container c = getContentPane();
11:       c.setLayout(new BorderLayout(10,10));
12:
13:       b1 = new JButton("button1");
14:       b2 = new JButton("button2");
15:       b3 = new JButton("button3");
16:       b4 = new JButton("button4");
17:       b5 = new JButton("button5");
18:
19:       c.add("North",b1);
20:       c.add("West",b2);
21:       c.add("East",b3);
22:       c.add("South",b4);
23:       c.add("Center",b5);
24:
25:       setTitle("Border Layout");
26:       setSize(400,400);
27:
      setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOS
    E);
28:       setVisible(true);
29:  }
30:
31:  public static void main(String args[])
32:  {
33:       MyBorderLayout border = new MyBorderLayout();
34:  }
35:}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyBorderLayout التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء 5 متغيرات من نوع JButton.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم تحديد BorderLayout ليصبح هو مدير التخطيط Layout Manager لنافذة البرنامج عن طريق الدالة ()setLayout.
- في السطور من 13 إلي 17 يتم إنشاء 6 أهداف Objects من نوع JButton والذي يمثل أداة الزر JButton.
- في السطور من 31 إلي 34 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyBorderLayout وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-5).



الشكل (16-5) مدير التخطيط BorderLayout

ملحوظات:

- في السطور من 19 إلى 23 يتم إضافة الأدوات Controls إلي النافذة. لاحظ أنه لإضافة الزر b1 كتبنا:

```
c.add ("No4rth",b1);
```

حيث أن القيمة الأولى للدالة ()add تحدد المنطقة التى ستضيف فيها الزر والقيمة الثانية تحدد اسم الأداة Controls التى ستضيفها.

- إذا تركت منطقة من الخمس مناطق بلا أى أدوات Control فيها فكأن هذه المنطقة ليست موجودة ، فمثلاً إذا لم تضف أى أداة Control للمنطقة North سيصبح شكل نافذة البرنامج كالآتى:

مدير التخطيط GridBagLayout:

- مدير التخطيط GridBagLayout هو أفضل مدير تخطيط Layout Manager من ناحية المرونة فهو ينظم الأدوات Controls فى صفوف وأعمدة ولكنك تستطيع تغيير حجم ومكان الأداة Control.
- لاستخدام GridBagLayout فإنك تنشئ هدفاً object من نوع الفصيلة class المسماة GridBagLayout وهدفاً object من نوع الفصيلة class المسماة GridBagConstraints التي نستخدمها لوضع قيود وخصائص لكل أداة Control تضيفها ثم تستخدم الدالة setConstraints() الموجودة فى الهدف object الذى من نوع الفصيلة GridBagLayout لكى تحدد القيود والخصائص على أداة control معينة.

أي أنه يتم إنشاء كود مثل الكود التالي:

أولاً: نقوم بإنشاء هدف Object من الأداة المطلوب إضافتها في نافذة البرنامج كالتالي:

```
JLabel labelshopperid = new JLabel("shopper id");
```

ثانياً: نقوم بإنشاء هدفين Objects من نوع GridBagLayout و GridBagConstraints كالتالي:

```
GridBagLayout grid = new GridBagLayout();
GridBagConstraints gbc = new GridBagConstraints();
```

الفصيلة GridBagConstraints تحتوى على المتغيرات التالية:

- gridx و gridy: ويستخدمان لتحديد مكان الأداة Control بحيث أن تحديد قيمة المتغير gridx بالقيمة صفر معناها أن الأداة Control سيتم وضعها فى أقصى الشمال ، أما تحديد قيمة المتغير gridy بالقيمة صفر معناها أن الأداة Control سيتم وضعها فى أعلى نقطة فى النافذة Frame ، فمثلاً يمكننا كتابة التالي.

```
gbc.gridx = 1;
gbc.gridy = 4;
```

بعد تحديد القيود عن طريق المتغيرات فى الهدف object الذى من نوع الفصيلة GridBagConstraints فإننا نستخدم الدالة setConstraints() الموجودة فى الهدف object الذى من نوع الفصيلة GridBagLayout لتحديد القيود على الأداة Control المطلوب إضافتها في نافذة البرنامج ولذلك نمرر قيمتين لهذه الدالة Method.

- القيمة الأولى تحدد الأداة Control التى سيتم وضع قيود عليها.
- القيمة الثانية تحدد الهدف object الذى حدد هذه القيود.

أي كالتالي:

```
grid.setConstraints(labelshopperid,gbc);
```

وأخيراً يمكنك إضافة الأداة المطلوبة إلي نافذة البرنامج بالطريقة العادية.

تتوافر العديد من دوال البناء Constructors لهذه الفصيلة class كما يلي.

دالة البناء GridBagLayout():

تقوم بإنشاء مدير تخطيط Layout Manager من نوع GridBagLayout.

المثال التالى يوضح مدير التخطيط GridBagLayout.

مثال (4): مدير التخطيط GridBagLayout:

أولاً: هدف المثال:

توضيح كيفية وضع أكثر من أداة control باستخدام مدير التخطيط GridBagLayout.

ثانياً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class MyGridBagLayout extends JFrame
5:  {
6:   //declare labels
7:   JLabel
     labelshopperid,labelpass,labelfname,labellname,labelemail,
     labeladd,
8:   labelcity,labelstate,labelcountry,labelzip,labelphone,label
     creditno,
9:   labelcredittype,labelexpiredate;
10:  //end of declare labels
11:
12:  //declare textfield
13:  JTextField
     textshopperid,textfname,textlname,textemail,textadd,
14:  textcity,textstate,textcountry,textzip,textphone,textcreditn
     o,
15:  textexpiredate;
16:  JPasswordField textpass;
17:  //end of declare textfield
18:
19:  //declare buttons
20:  JButton submit,cancel;
21:  //end of declare butons
```

```
22:
23:  //declare combobox
24:  JComboBox combocredittype;
25:  //end of declare combobox
26:
27:  //declare layoutmanager
28:  GridBagLayout grid;
29:  GridBagConstraints gbc;
30:  //end of declare layoutmanager
31:
32:  MyGridBagLayout()
33:  {
34:      //initialize container
35:      Container c = getContentPane();
36:      //end of initialize container
37:
38:      // initialize gridbaglayout
39:      grid = new GridBagLayout();
40:      gbc = new GridBagConstraints();
41:      //end of initialize gridbaglayout
42:
43:      //set gridbaglayout to the container
44:      c.setLayout(grid);
45:      //end set gridbaglayout to the container
46:
47:      //initialize labels
48:      labelshopperid = new JLabel("shopper id");
49:      labelpass = new JLabel("password");
50:      labelfname = new JLabel("first name");
51:      labellname = new JLabel("last name");
52:      labelemail = new JLabel("e-mail");
53:      labeladd = new JLabel("address");
54:      labelcity = new JLabel("city");
55:      labelstate = new JLabel("state");
```

```
56:     labelcountry = new JLabel("country");
57:     labelzip = new JLabel("zip");
58:     labelphone = new JLabel("phone");
59:     labelcreditno = new JLabel("credit card number");
60:     labelcredittype = new JLabel("credit type");
61:     labelexpiredate = new JLabel("expire date");
62:     //end of initialize labels
63:
64:     //initialize textfield
65:     textshopperid = new JTextField(10);
66:     textpass = new JPasswordField(10);
67:     textfname = new JTextField(10);
68:     textlname = new JTextField(10);
69:     textemail = new JTextField(10);
70:     textadd = new JTextField(10);
71:     textcity = new JTextField(10);
72:     textstate = new JTextField(10);
73:     textcountry = new JTextField(10);
74:     textzip = new JTextField(10);
75:     textphone = new JTextField(10);
76:     textcreditno = new JTextField(10);
77:     textexpiredate = new JTextField(10);
78:     //end of initialize textfield
79:
80:     //initialize combobox
81:     String credittype[] = {"master","visa"};
82:     combocredittype = new JComboBox(credittype);

83:     //end of initialize combobox
84:
85:     //initialize button
86:     submit = new JButton("submit");
87:     cancel = new JButton("cancel");
88:     //end of initialize button
```

```
89:
90:     //add controls to container
91:     gbc.gridx = 1;
92:     gbc.gridy = 4;
93:     grid.setConstraints(labelshopperid,gbc);
94:     c.add(labelshopperid);
95:
96:     gbc.gridx = 4;
97:     gbc.gridy = 4;
98:     grid.setConstraints(textshopperid,gbc);
99:     c.add(textshopperid);
100:
101:        gbc.gridx = 1;
102:        gbc.gridy = 7;
103:        grid.setConstraints(labelpass,gbc);
104:        c.add(labelpass);
105:
106:        gbc.gridx = 4;
107:        gbc.gridy = 7;
108:        grid.setConstraints(textpass,gbc);
109:        c.add(textpass);
110:
111:        gbc.gridx = 1;
112:        gbc.gridy = 10;
113:        grid.setConstraints(labelemail,gbc);
114:        c.add(labelemail);
115:
116:        gbc.gridx = 4;
117:        gbc.gridy = 10;
118:        grid.setConstraints(textemail,gbc);
119:        c.add(textemail);
120:
121:        gbc.gridx = 1;
122:        gbc.gridy = 13;
```

```
123:        grid.setConstraints(labelfname,gbc);
124:        c.add(labelfname);
125:
126:        gbc.gridx = 4;
127:        gbc.gridy = 13;
128:        grid.setConstraints(textfname,gbc);
129:        c.add(textfname);
130:
131:        gbc.gridx = 1;
132:        gbc.gridy = 16;
133:        grid.setConstraints(labellname,gbc);
134:        c.add(labellname);
135:
136:        gbc.gridx = 4;
137:        gbc.gridy = 16;
138:        grid.setConstraints(textlname,gbc);
139:        c.add(textlname);
140:
141:        gbc.gridx = 1;
142:        gbc.gridy = 19;
143:        grid.setConstraints(labeladd,gbc);
144:        c.add(labeladd);
145:
146:        gbc.gridx = 4;
147:        gbc.gridy = 19;
148:        grid.setConstraints(textadd,gbc);
149:        c.add(textadd);
150:
151:        gbc.gridx = 1;
152:        gbc.gridy = 22;
153:        grid.setConstraints(labelcity,gbc);
154:        c.add(labelcity);
155:
156:        gbc.gridx = 4;
```

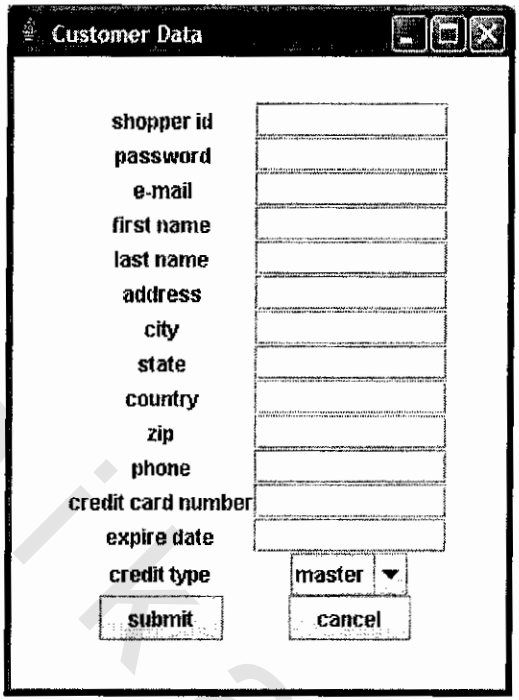
```
157:        gbc.gridy = 22;
158:        grid.setConstraints(textcity,gbc);
159:        c.add(textcity);
160:
161:        gbc.gridx = 1;
162:        gbc.gridy = 25;
163:        grid.setConstraints(labelstate,gbc);
164:        c.add(labelstate);
165:
166:        gbc.gridx = 4;
167:        gbc.gridy = 25;
168:        grid.setConstraints(textstate,gbc);
169:        c.add(textstate);
170:
171:        gbc.gridx = 1;
172:        gbc.gridy = 28;
173:        grid.setConstraints(labelcountry,gbc);
174:        c.add(labelcountry);
175:
176:        gbc.gridx = 4;
177:        gbc.gridy = 28;
178:        grid.setConstraints(textcountry,gbc);
179:        c.add(textcountry);
180:
181:        gbc.gridx = 1;
182:        gbc.gridy = 31;
183:        grid.setConstraints(labelzip,gbc);
184:        c.add(labelzip);
185:
186:        gbc.gridx = 4;
187:        gbc.gridy = 31;
188:        grid.setConstraints(textzip,gbc);
189:        c.add(textzip);
190:
```

```
191:        gbc.gridx = 1;
192:        gbc.gridy = 34;
193:        grid.setConstraints(labelphone,gbc);
194:        c.add(labelphone);
195:
196:        gbc.gridx = 4;
197:        gbc.gridy = 34;
198:        grid.setConstraints(textphone,gbc);
199:        c.add(textphone);
200:
201:        gbc.gridx = 1;
202:        gbc.gridy = 37;
203:        grid.setConstraints(labelcreditno,gbc);
204:        c.add(labelcreditno);
205:
206:        gbc.gridx = 4;
207:        gbc.gridy = 37;
208:        grid.setConstraints(textcreditno,gbc);
209:        c.add(textcreditno);
210:
211:        gbc.gridx = 1;
212:        gbc.gridy = 43;
213:        grid.setConstraints(labelcredittype,gbc);
214:        c.add(labelcredittype);
215:
216:        gbc.gridx = 4;
217:        gbc.gridy = 43;
218:        grid.setConstraints(combocredittype,gbc);
219:        c.add(combocredittype);
220:
221:        gbc.gridx = 1;
222:        gbc.gridy = 40;
223:        grid.setConstraints(labelexpiredate,gbc);
224:        c.add(labelexpiredate);
```

```
225:
226:            gbc.gridx = 4;
227:            gbc.gridy = 40;
228:            grid.setConstraints(textexpiredate,gbc);
229:            c.add(textexpiredate);
230:
231:            gbc.gridx = 1;
232:            gbc.gridy = 46;
233:            grid.setConstraints(submit,gbc);
234:            c.add(submit);
235:
236:            gbc.gridx = 4;
237:            gbc.gridy = 46;
238:            grid.setConstraints(cancel,gbc);
239:            c.add(cancel);
240:            // end of add controls to container
241:
242:            //frame properties
243:            setTitle("Customer Data");
244:            setSize(300,400);
245:
      setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOS
    E);
246:            setVisible(true);
247:            //end of frame properties
248:        }
249:
250:        public static void main(String args[])
251:        {
252:            MyGridBagLayout        gridbag        =        new
    MyGridBagLayout();
253:        }
254:}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyGridBagLayout التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطور من 7 إلي 9 يتم إنشاء متغيرات من نوع JLabel.
- في السطور من 13 إلي 15 يتم إنشاء متغيرات من نوع JTextField.
- في السطر رقم 16 يتم استخدام أداة كلمة السر JPasswordField وهي مثل أداة النص JTextField ولكن الفارق يتمثل في أنه عندما تكتب أى حروف فإنها تظهر كنجوم ولا تبين الكتابة الأصلية ، ولذلك نستخدمها عندما نريد كتابة كلمة سر.
- في السطر رقم 20 يتم إنشاء متغيرات من نوع JButton.
- في السطر رقم 24 يتم إنشاء متغير من نوع JComboBox.
- في السطر رقم 28 يتم إنشاء متغير من نوع GridBagLayout.
- في السطر رقم 29 يتم إنشاء متغير من نوع GridBagConstraints.
- في السطر رقم 32 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 39 يتم إنشاء هدف Object من نوع GridBagLayout.
- في السطر رقم 40 يتم إنشاء هدف Object من نوع GridBagConstraints.
- في السطر رقم 44 يتم تحديد GridBagLayout ليصبح هو مدير التخطيط Layout Manager لنافذة البرنامج عن طريق الدالة ()setLayout.
- في السطور من 48 إلي 239 يتم إنشاء نافذة البرنامج.
- في السطور من 250 إلي 253 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyGridBagLayout وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (6-16).



الشكل (16-6) مدير التخطيط GridBagLayout

تركيب أكثر من تخطيط Combining Layouts:

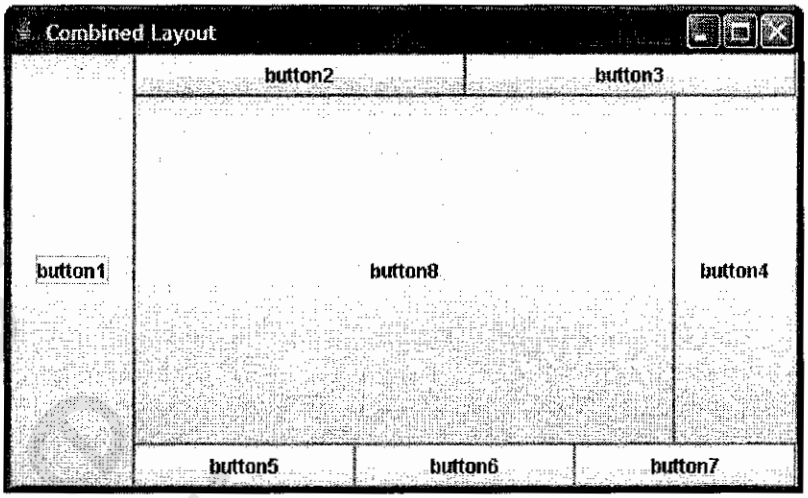
لاحظنا أن كل مدير تخطيط Layout Manager به ميزة عن أي مدير تخطيط Layout Manager أخر ، وعادة نريد أن ننشئ نافذة البرنامج وبها العديد من الأدوات التي تتطلب في كثير من الأحيان استخدام أكثر من مدير تخطيط Layout Manager ليوصلنا لبناء نافذة البرنامج.

الأمثلة التالية توضح هذه النقطة.

مثال (5): تركيب أكثر من تخطيط Combining Layouts:

أولاً: هدف المثال:

توضيح كيفية تركيب أكثر من تخطيط Combining Layuts لإنشاء نافذة البرنامج كما هو واضح في الشكل (16-7).

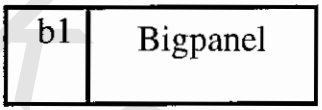


الشكل (16-7) تركيب أكثر من تخطيط Combining Layouts

ثانياً: التخطيط للحل:

1. يتشابه الشكل المبين مع مدير التخطيط BorderLayout مع اختلاف أنه يوجد أكثر من أداة Control فى المنطقة الواحدة ، ولذلك نبدأ بتحديد التخطيط ليكون BorderLayout للـ Container c.
2. نضيف الزر b1 إلى المنطقة الغربية West للـ Container c.
3. ننشئ هدفاً Object من نوع JPanel ونسميه northPanel ونحدد لها مدير التخطيط GridLayout ونضع فيها صف واحد وعمودين.
4. نضيف الزرين b2 و b3 للوحة northPanel وبذلك أصبحت northPanel تحمل زرين Buttons.
5. ننشئ هدفاً Object من نوع JPanel ونسميه southPanel ونحدد لها مدير التخطيط GridLayout ونضع فيها صف واحد و3 أعمدة.
6. نضيف الأزرار b5 و b6 و b7 للوحة southPanel ، وبذلك أصبحت southPanel تحمل 3 أزرار Buttons.
7. ننشئ هدفاً Object من نوع JPanel ونسميه bigpanel ونحدد لها مدير التخطيط BorderLayout.

8. نضيف اللوحة northPanel للمنطقة الشمالية North للوحة bigpanel ونضيف اللوحة southPanel للمنطقة الجنوبية South للوحة bigpanel.
9. نضيف الزر b4 للمنطقة الشرقية East للوحة bigpanel.
10. نضيف الزر b8 لمنطقة الوسط Center للوحة bigpanel.
11. أخيراً نضيف اللوحة bigpanel إلى منطقة الوسط Center للـ Container c.
12. لاحظ أنه بالنسبة للـ Container c الذى له مدير تخطيط BorderLayout ، فإننا لم نضف أي أداة Control إلى المناطق South وNorth و East ولكننا أضفنا الزر b1 إلى المنطقة West وأضفنا اللوح BigEastPanel لمنطقة الوسط Center ليصبح الشكل كما يلى.



ثم بدأنا إضافة الأدوات Controls إلي اللوح bigpanel ثم تابعنا إضافة الأدوات حتي ننتهي من إنشاء نافذة البرنامج بالطريقة التي نريدها.

ثالثاً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class CombinedLayout1 extends JFrame
5:  {
6:   JButton b1,b2,b3,b4,b5,b6,b7,b8;
7:   JPanel northpanel,southpanel,bigpanel;
8:
9:   CombinedLayout1()
10:  {
11:       Container c = getContentPane();
12:
13:       b1 = new JButton("button1");
```

```
14:       b2 = new JButton("button2");
15:       b3 = new JButton("button3");
16:       b4 = new JButton("button4");
17:       b5 = new JButton("button5");
18:       b6 = new JButton("button6");
19:       b7 = new JButton("button7");
20:       b8 = new JButton("button8");
21:
22:       northpanel = new JPanel();
23:       southpanel = new JPanel();
24:       bigpanel = new JPanel();
25:
26:       c.setLayout(new BorderLayout());
27:       c.add("West",b1);
28:
29:       northpanel.setLayout(new GridLayout(1,2));
30:       northpanel.add(b2);
31:       northpanel.add(b3);
32:
33:       southpanel.setLayout(new GridLayout(1,3));
34:       southpanel.add(b5);
35:       southpanel.add(b6);
36:       southpanel.add(b7);
37:
38:    bigpanel.setLayout(new BorderLayout());
39:    bigpanel.add("North",northpanel);
40:       bigpanel.add("South",southpanel);
41:    bigpanel.add("East",b4);
42:       bigpanel.add("Center",b8);
43:    c.add("Center",bigpanel);
44:
45:       setTitle("Combined Layout");
46:       setSize(500,300);
47:       setVisible(true);
48: }
```

```
49:
50:  public static void main(String args[])
51:  {
52:      CombinedLayout1 layout = new
     CombinedLayout1();
53:  }
54:}
```

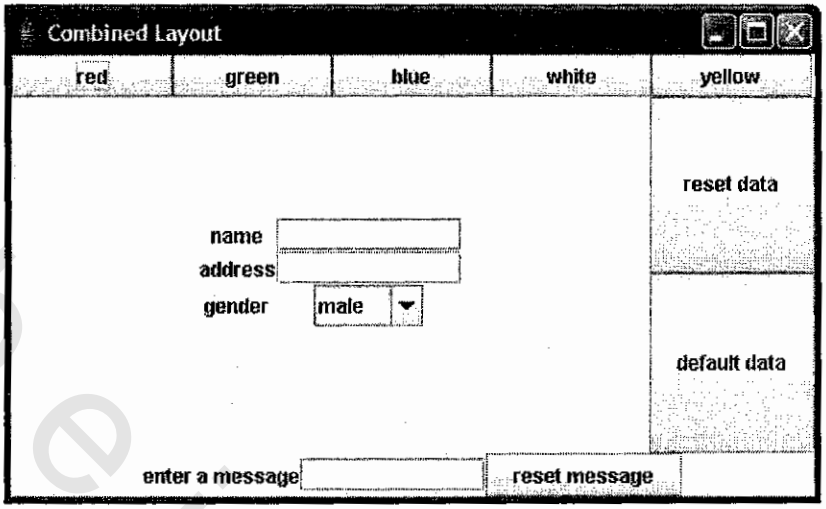
رابعاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة CombinedLayout1 التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء 8 متغيرات من نوع JButton.
- في السطر رقم 7 يتم إنشاء 3 متغيرات من نوع JPanel.
- في السطر رقم 9 يتم إنشاء دالة البناء Constructor.
- في السطور من 13 إلي 43 يتم تنفيذ خطوات إنشاء نافذة البرنامج كما سبق لنا شرحه في نقطة التخطيط للحل.
- في السطور من 50 إلي 53 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة CombinedLayout1 وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-7) السابق.

مثال (6): تركيب أكثر من تخطيط Combining Layouts:

أولاً: هدف المثال:

توضيح كيفية تركيب أكثر من تخطيط Combining Layuts لإنشاء نافذة البرنامج كما هو واضح في الشكل (16-8).



الشكل (16-8) تركيب أكثر من تخطيط Combining Layouts

ثانياً: التخطيط للحل:

1. ننشئ 8 أزرار JButton.
2. ننشئ 4 أداة عنوان JLabel.
3. ننشئ 3 أداة نص JTextField.
4. ننشئ 1 أداة قائمة اختيارات JComboBox.
5. ننشئ 4 ألواح JPanel.
6. نجعل مدير التخطيط للـ Container c هو مدير التخطيط BorderLayout.
7. نجعل مدير التخطيط للوحة northPanel هو GridLayout ونضيف فيه 5 أزرار.
8. نجعل مدير التخطيط للوحة eastPanel هو GridLayout ونضيف فيه زرين.
9. نجعل مدير التخطيط للوحة centerPanel هو GridBagLayout ونضيف فيها 3 أداة عنوان JLabel و2 أداة نص JTextField و1 أداة قائمة اختيارات JComboBox.
10. نجعل مدير التخطيط للوحة southPanel هو GridBagLayout ونضيف فيه أداة عنوان JLabel وأداة نص JTextField وزر JButton.
11. أخيراً نضيف الألواح JPanel الأربعة إلى الـ Container c.

ثالثاً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class CombinedLayout2 extends JFrame
5:  {
6:   JButton
    redbutton,greenbutton,bluebutton,yellowbutton,whitebutton,
7:   resetbutton,messagebutton,defaultbutton;
8:   JLabel name,address,gender,message;
9:   JTextField nametext,addresstext,messagetext;
10:  JComboBox gendercombo;
11:  JPanel northpanel,southpanel,eastpanel,centerpanel;
12:  GridBagLayout centergrid,southgrid;
13:  GridBagConstraints centergbc,southgbc;
14:
15:  CombinedLayout2()
16:  {
17:       Container c = getContentPane();
18:       c.setLayout(new BorderLayout());
19:
20:       northpanel = new JPanel();
21:       southpanel = new JPanel();
22:       eastpanel = new JPanel();
23:       centerpanel = new JPanel();
24:
25:       redbutton = new JButton("red");
26:       greenbutton = new JButton("green");
27:       bluebutton = new JButton("blue");
28:       yellowbutton = new JButton("yellow");
29:       whitebutton = new JButton("white");
30:
31:       resetbutton = new JButton("reset data");
32:       defaultbutton = new JButton("default data");
33:
```

```
34:       messagebutton = new JButton("reset message");
35:
36:       name = new JLabel("name");
37:       address = new JLabel("address");
38:       gender = new JLabel("gender");
39:       message = new JLabel("enter a message");
40:
41:       nametext = new JTextField(10);
42:
43:       nametext.requestFocus();
44:
45:       addresstext = new JTextField(10);
46:       messagetext = new JTextField(10);
47:       String kind[] = {"male","female"};
48:       gendercombo = new JComboBox(kind);
49:
50:       northpanel.setLayout(new GridLayout(1,5));
51:    northpanel.add(redbutton);
52:       northpanel.add(greenbutton);
53:       northpanel.add(bluebutton);
54:       northpanel.add(whitebutton);
55:       northpanel.add(yellowbutton);
56:
57:       eastpanel.setLayout(new GridLayout(2,1));
58:    eastpanel.add(resetbutton);
59:       eastpanel.add(defaultbutton);
60:
61:       centergrid = new GridBagLayout();
62:       centergbc = new GridBagConstraints();
63:       centerpanel.setLayout(centergrid);
64:
65:       centergbc.gridx = 1;
66:       centergbc.gridy = 1;
67:       centergrid.setConstraints(name,centergbc);
68:       centerpanel.add(name);
```

```
69:
70:       centergbc.gridx = 2;
71:       centergbc.gridy = 1;
72:       centergrid.setConstraints(nametext,centergbc);
73:       centerpanel.add(nametext);
74:
75:       centergbc.gridx = 1;
76:       centergbc.gridy = 2;
77:       centergrid.setConstraints(address,centergbc);
78:       centerpanel.add(address);
79:
80:       centergbc.gridx = 2;
81:       centergbc.gridy = 2;
82:       centergrid.setConstraints(addresstext,centergbc);
83:       centerpanel.add(addresstext);
84:
85:       centergbc.gridx = 1;
86:       centergbc.gridy = 3;
87:       centergrid.setConstraints(gender,centergbc);
88:       centerpanel.add(gender);
89:
90:       centergbc.gridx = 2;
91:       centergbc.gridy = 3;
92:       centergrid.setConstraints(gendercombo,centergbc);
93:       centerpanel.add(gendercombo);
94:
95:    southgrid = new GridBagLayout();
96:       southgbc = new GridBagConstraints();
97:       southpanel.setLayout(southgrid);
98:
99:       southgbc.gridx = 1;
100:          southgbc.gridy = 1;
101:          southgrid.setConstraints(message,southgbc);
102:          southpanel.add(message);
103:
```

```
104:            southgbc.gridx = 3;
105:            southgbc.gridy = 1;
106:            southgrid.setConstraints(messagetext,southgbc);
107:            southpanel.add(messagetext);
108:
109:            southgbc.gridx = 5;
110:            southgbc.gridy = 1;
111:
      southgrid.setConstraints(messagebutton,southgbc);
112:            southpanel.add(messagebutton);
113:
114:    c.add("North",northpanel);
115:            c.add("South",southpanel);
116:            c.add("East",eastpanel);
117:            c.add("Center",centerpanel);
118:
119:            setTitle("Combined Layout");
120:            setSize(500,300);
121:            setVisible(true);
122:        }
123:
124:        public static void main(String args[])
125:        {
126:            CombinedLayout2        layout        =        new
    CombinedLayout2();
127:        }
128:}
```

رابعاً: شرح الكود:

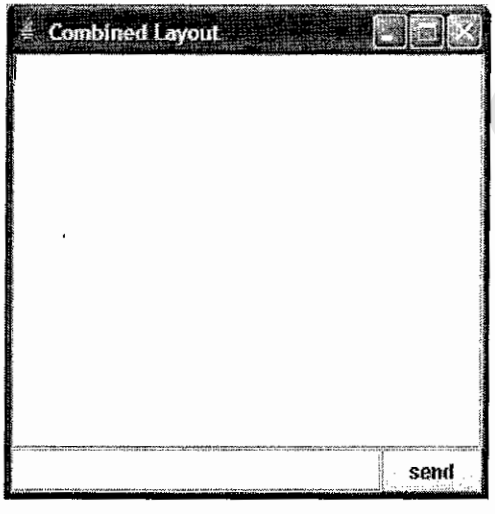
- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة CombinedLayout2 التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطرين رقم 6 و 7 يتم إنشاء 8 متغيرات من نوع JButton.
- في السطر رقم 8 يتم إنشاء 4 متغيرات من نوع JLabel.

- في السطر رقم 9 يتم إنشاء 3 متغيرات من نوع JTextField.
- في السطر رقم 10 يتم إنشاء متغير من نوع JComboBox.
- في السطر رقم 11 يتم إنشاء 4 متغيرات من نوع JPanel.
- في السطر رقم 15 يتم إنشاء دالة البناء Constructor.
- في السطور من 20 إلي 117 يتم تنفيذ خطوات إنشاء نافذة البرنامج كما سبق لنا شرحه في نقطة التخطيط للحل.
- في السطور من 124 إلي 127 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة CombinedLayout2 وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-8) السابق.

مثال (7): تركيب أكثر من تخطيط Combining Layouts:

أولاً: هدف المثال:

توضيح كيفية تركيب أكثر من تخطيط Combining Layuts لإنشاء نافذة البرنامج كما هو واضح في الشكل (16-9).



الشكل (16-9) تركيب أكثر من تخطيط Combining Layouts

ثانياً: التخطيط للحل:

1. ننشئ 1 لوحة JPanel.
2. ننشئ 1 أداة نص JTextArea.
3. ننشئ 1 أداة نص JTextField.
4. ننشئ 1 زر JButton.
5. ننشئ 1 أداة لوحة التمرير JScrollPane.
6. نجعل مدير التخطيط للـ Container c هو مدير التخطيط BorderLayout.
7. نقوم بإضافة أداة لوحة التمرير JScrollPane إلي المنطقة الوسطي للـ Container c.
8. نجعل مدير التخطيط للوحة southPanel هو BorderLayout ونضيف فيها أداة نص JTextField وزر JButton في المنطقتين الوسطي والشرقية علي الترتيب.
9. نضيف اللوحة southpanel إلى الـ Container c في المنطقة الجنوبية.

ثالثاً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class CombinedLayout3 extends JFrame
5:  {
6:    JPanel southpanel;
7:    JTextArea area;
8:    JTextField text;
9:    JButton send;
10:  JScrollPane scroll;
11:
12:  CombinedLayout3()
13:  {
14:       Container c = getContentPane();
15:
16:       southpanel = new JPanel();
17:       area = new JTextArea();
```

```
18:       text = new JTextField();
19:       send = new JButton("send");
20:       scroll = new JScrollPane(area);
21:
22:       c.setLayout(new BorderLayout());
23:       c.add("Center",scroll);
24:
25:       southpanel.setLayout(new BorderLayout());
26:       southpanel.add("Center",text);
27:       southpanel.add("East",send);
28:
29:    c.add("South",southpanel);
30:
31:       setTitle("Combined Layout");
32:       setSize(300,300);
33:       setVisible(true);
34:  }
35:
36:  public static void main(String args[])
37:  {
38:       CombinedLayout3 layout = new CombinedLayout3();
39:  }
40:}
```

رابعاً: شرح الكود:

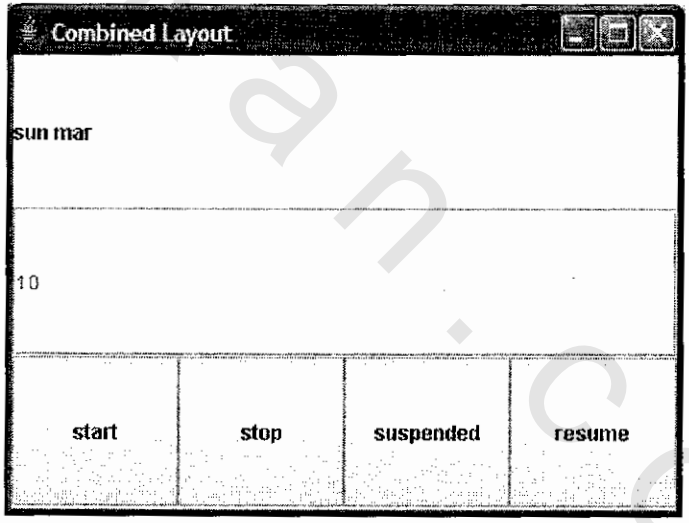
- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة CombinedLayout3 التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطور من 6 إلي 10 يتم إنشاء جميع المتغيرات التي نحتاجها في البرنامج.
- في السطر رقم 12 يتم إنشاء دالة البناء Constructor.
- في السطور من 22 إلي 29 يتم تنفيذ خطوات إنشاء نافذة البرنامج كما سبق لنا شرحه في نقطة التخطيط للحل.

- في السطور من 36 إلي 39 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة CombinedLayout3 وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-9) السابق.

مثال (8): تركيب أكثر من تخطيط Combining Layouts:

أولاً: هدف المثال:

توضيح كيفية تركيب أكثر من تخطيط Combining Layouts لإنشاء نافذة البرنامج كما هو واضح في الشكل (16-10).



الشكل (16-10) تركيب أكثر من تخطيط Combining Layouts

ثانياً: التخطيط للحل:

1. ننشئ 1 لوحة JPanel.
2. ننشئ 4 أزرار JButton.
3. ننشئ 1 أداة نص JTextField.

4. ننشئ 1 أداة عنوان JLabel.
5. نجعل مدير التخطيط للـ Container c هو مدير التخطيط GridLayout.
6. نجعل مدير التخطيط للوحة downpanel هـو GridLayout ونضـيف فيهـا 4 أزرار JButton.
7. نضيف أداة العنوان JLabel وأداة الـنص JTextField واللوحـة southpanel إلى الـ Container c.

ثالثاً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class CombinedLayout4 extends JFrame
5:  {
6:   JPanel downpanel;
7:   JButton start,stop,suspended,resume;
8:   JTextField text;
9:   JLabel sun;
10:
11:  CombinedLayout4()
12:  {
13:      Container c = getContentPane();
14:      c.setLayout(new GridLayout(3,1));
15:
16:      downpanel = new JPanel();
17:      start = new JButton("start");
18:      stop = new JButton("stop");
19:      suspended = new JButton("suspended");
20:      resume = new JButton("resume");
21:      text = new JTextField("10");
22:      sun = new JLabel("sun mar");
23:
24:      downpanel.setLayout(new GridLayout(1,4));
25:      downpanel.add(start);
```

```
26:       downpanel.add(stop);
27:       downpanel.add(suspended);
28:       downpanel.add(resume);
29:
30:    c.add(sun);
31:       c.add(text);
32:       c.add(downpanel);
33:
34:       setTitle("Combined Layout");
35:       setSize(400,300);
36:       setVisible(true);
37:  }
38:
39:  public static void main(String args[])
40:  {
41:       CombinedLayout4        layout        =        new
     CombinedLayout4();
42:  }
43:}
```

رابعاً: شرح الكود:

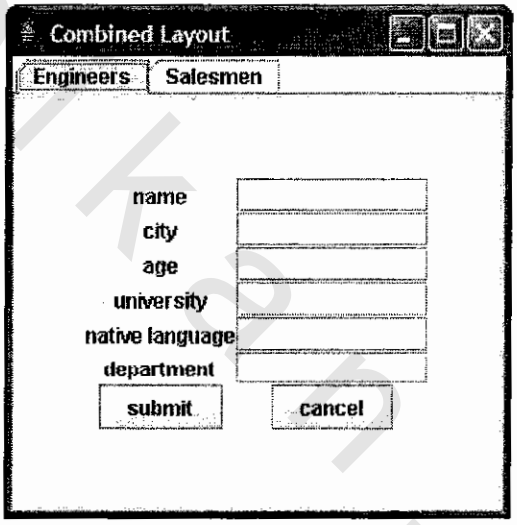
- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة CombinedLayout4 التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطور من 6 إلي 9 يتم إنشاء جميع المتغيرات التي نحتاجها في البرنامج.
- في السطر رقم 11 يتم إنشاء دالة البناء Constructor.
- في السطور من 24 إلي 32 يتم تنفيذ خطوات إنشاء نافذة البرنامج كما سبق لنا شرحه في نقطة التخطيط للحل.
- في السطور من 39 إلي 42 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة CombinedLayout4 وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.

قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-10) السابق.

مثال (9): تركيب أكثر من تخطيط Combining Layouts:

أولاً: هدف المثال:

توضيح كيفية تركيب أكثر من تخطيط Combining Layuts لإنشاء نافذة البرنامج كما هو واضح في الشكل (16-11).



الشكل (16-11) تركيب أكثر من تخطيط Combining Layouts

ثانياً: التخطيط للحل:

1. هذا المثال يحتوي علي أداة اللوحة المبوبة JTabbedPane وسنفترض وجود تبويبين Tabs يحتويان علي نفس البيانات ولكن يمكنك بالطبع إنشاء تبويب Tab يختلف كلية عن التبويب Tab الأول ، ولذلك ستجد تكراراً في الأدوات المستخدمة.
2. ننشئ 4 أزرار JButton.
3. ننشئ 2 لوحة JPanel.
4. ننشئ 1 أداة اللوحة المبوبة JTabbedPane.

5. ننشئ 12 أداة عنوان JLabel.
6. ننشئ 12 أداة نص JTextField.
7. نجعل مدير التخطيط لكل من اللوحتين p1 و p2 هو مدير التخطيط GridLayout.
8. نضيف الأدوات المطلوبة في كل لوحة JPanel.
9. نضيف اللوحتين JPanel لأداة اللوحة المبوبة JTabbedPane.
10. نضيف أداة اللوحة المبوبة JTabbedPane إلى الـ Container c.

ثالثاً: كود البرمجة:

```
1:  import javax.swing.*;
2:  import java.awt.*;
3:
4:  public class CombinedLayout5 extends JFrame
5:  {
6:    JButton cancel,submit;
7:    JButton cancel1,submit1;
8:    JPanel p1,p2;
9:    JTabbedPane tab;
10:  JLabel
     name,city,age,university,nativelanguage,department;
11:  JLabel
     name1,city1,age1,university1,nativelanguage1,department1;
12:  JTextField nametext,citytext,agetext,universitytext,
13:  nativelanguagetext,departmenttext;
14:  JTextField
     nametext1,citytext1,agetext1,universitytext1,
15:  nativelanguagetext1,departmenttext1;
16:  GridBagLayout grid;
17:  GridBagConstraints gbc;
18:
19:  CombinedLayout5()
20:  {
21:       Container c = getContentPane();
22:
```

```
23:       cancel = new JButton("cancel");
24:       submit = new JButton("submit");
25:
26:    cancel1 = new JButton("cancel");
27:       submit1 = new JButton("submit");
28:
29:       tab = new JTabbedPane();
30:       p1 = new JPanel();
31:       p2 = new JPanel();
32:       grid = new GridBagLayout();
33:       gbc = new GridBagConstraints();
34:
35:       name = new JLabel("name");
36:       city = new JLabel("city");
37:       university = new JLabel("university");
38:       age = new JLabel("age");
39:       nativelanguage = new JLabel("native language");
40:       department = new JLabel("department");
41:
42:    name1 = new JLabel("name");
43:       city1 = new JLabel("city");
44:       university1 = new JLabel("university");
45:       age1 = new JLabel("age");
46:       nativelanguage1 = new JLabel("native language");
47:       department1 = new JLabel("department");
48:
49:       nametext = new JTextField(10);
50:       citytext = new JTextField(10);
51:       agetext = new JTextField(10);
52:       universitytext = new JTextField(10);
53:       nativelanguagetext = new JTextField(10);
54:       departmenttext = new JTextField(10);
55:
56:    nametext1 = new JTextField(10);
57:       citytext1 = new JTextField(10);
```

```
58:     agetext1 = new JTextField(10);
59:     universitytext1 = new JTextField(10);
60:     nativelanguagetext1 = new JTextField(10);
61:     departmenttext1 = new JTextField(10);
62:
63:     p1.setLayout(grid);
64:
65:     gbc.gridx = 1;
66:     gbc.gridy = 1;
67:     grid.setConstraints(name,gbc);
68:     p1.add(name);
69:
70:     gbc.gridx = 2;
71:     gbc.gridy = 1;
72:     grid.setConstraints(nametext,gbc);
73:     p1.add(nametext);
74:
75:     gbc.gridx = 1;
76:     gbc.gridy = 2;
77:     grid.setConstraints(city,gbc);
78:     p1.add(city);
79:
80:     gbc.gridx = 2;
81:     gbc.gridy = 2;
82:     grid.setConstraints(citytext,gbc);
83:     p1.add(citytext);
84:
85:     gbc.gridx = 1;
86:     gbc.gridy = 3;
87:     grid.setConstraints(age,gbc);
88:     p1.add(age);
89:
90:     gbc.gridx = 2;
91:     gbc.gridy = 3;
92:     grid.setConstraints(agetext,gbc);
```

```
93:     p1.add(agetext);
94:
95:     gbc.gridx = 1;
96:     gbc.gridy = 4;
97:     grid.setConstraints(university,gbc);
98:     p1.add(university);
99:
100:        gbc.gridx = 2;
101:        gbc.gridy = 4;
102:        grid.setConstraints(universitytext,gbc);
103:        p1.add(universitytext);
104:
105:        gbc.gridx = 1;
106:        gbc.gridy = 5;
107:        grid.setConstraints(nativelanguage,gbc);
108:        p1.add(nativelanguage);
109:
110:        gbc.gridx = 2;
111:        gbc.gridy = 5;
112:        grid.setConstraints(nativelanguagetext,gbc);
113:        p1.add(nativelanguagetext);
114:
115:        gbc.gridx = 1;
116:        gbc.gridy = 6;
117:        grid.setConstraints(department,gbc);
118:        p1.add(department);
119:
120:        gbc.gridx = 2;
121:        gbc.gridy = 6;
122:        grid.setConstraints(departmenttext,gbc);
123:        p1.add(departmenttext);
124:
125:        gbc.gridx = 1;
126:        gbc.gridy = 7;
127:        grid.setConstraints(submit,gbc);
```

```
128:        p1.add(submit);
129:
130:        gbc.gridx = 2;
131:        gbc.gridy = 7;
132:        grid.setConstraints(cancel,gbc);
133:        p1.add(cancel);
134:
135:    p2.setLayout(grid);
136:
137:        gbc.gridx = 1;
138:        gbc.gridy = 1;
139:        grid.setConstraints(name1,gbc);
140:        p2.add(name1);
141:
142:        gbc.gridx = 2;
143:        gbc.gridy = 1;
144:        grid.setConstraints(nametext1,gbc);
145:        p2.add(nametext1);
146:
147:        gbc.gridx = 1;
148:        gbc.gridy = 2;
149:        grid.setConstraints(city1,gbc);
150:        p2.add(city1);
151:
152:        gbc.gridx = 2;
153:        gbc.gridy = 2;
154:        grid.setConstraints(citytext1,gbc);
155:        p2.add(citytext1);
156:
157:        gbc.gridx = 1;
158:        gbc.gridy = 3;
159:        grid.setConstraints(age1,gbc);
160:        p2.add(age1);
161:
162:        gbc.gridx = 2;
```

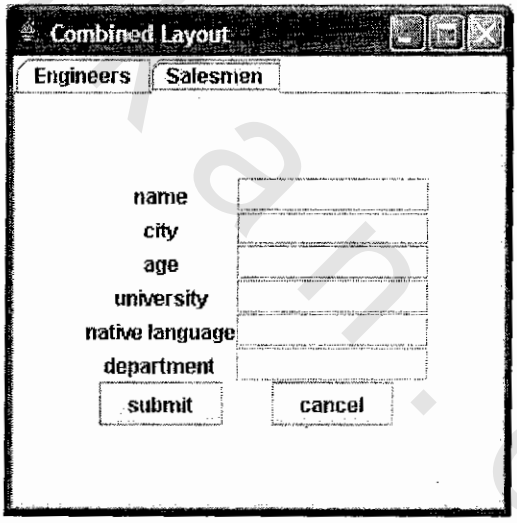
```
163:        gbc.gridy = 3;
164:        grid.setConstraints(agetext1,gbc);
165:        p2.add(agetext1);
166:
167:        gbc.gridx = 1;
168:        gbc.gridy = 4;
169:        grid.setConstraints(university1,gbc);
170:        p2.add(university1);
171:
172:        gbc.gridx = 2;
173:        gbc.gridy = 4;
174:        grid.setConstraints(universitytext1,gbc);
175:        p2.add(universitytext1);
176:
177:        gbc.gridx = 1;
178:        gbc.gridy = 5;
179:        grid.setConstraints(nativelanguage1,gbc);
180:        p2.add(nativelanguage1);
181:
182:        gbc.gridx = 2;
183:        gbc.gridy = 5;
184:        grid.setConstraints(nativelanguagetext1,gbc);
185:        p2.add(nativelanguagetext1);
186:
187:        gbc.gridx = 1;
188:        gbc.gridy = 6;
189:        grid.setConstraints(department1,gbc);
190:        p2.add(department1);
191:
192:        gbc.gridx = 2;
193:        gbc.gridy = 6;
194:        grid.setConstraints(departmenttext1,gbc);
195:        p2.add(departmenttext1);
196:
197:        gbc.gridx = 1;
```

```
198:            gbc.gridy = 7;
199:            grid.setConstraints(submit1,gbc);
200:            p2.add(submit1);
201:
202:            gbc.gridx = 2;
203:            gbc.gridy = 7;
204:            grid.setConstraints(cancel1,gbc);
205:            p2.add(cancel1);
206:
207:    tab.addTab("Engineers",null,p1,"this is panel1");
208:            tab.addTab("Salesmen",null,p2,"this          is
    panel2");
209:    c.add(tab);
210:
211:            setTitle("Combined Layout");
212:            setSize(300,300);
213:            setVisible(true);
214:        }
215:
216:        public static void main(String args[])
217:        {
218:            CombinedLayout5      layout      =      new
    CombinedLayout5();
219:        }
220:}
```

رابعاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة CombinedLayout5 التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطور من 6 إلي 17 يتم إنشاء جميع المتغيرات التي نحتاجها في البرنامج.
- في السطر رقم 19 يتم إنشاء دالة البناء Constructor.

- في السطور من 23 إلي 209 يتم تنفيذ خطوات إنشاء نافذة البرنامج كما سبق لنا شرحه في نقطة التخطيط للحل.
- في السطور من 216 إلي 219 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة CombinedLayout5 وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-11) السابق.
- لاحظ أنه يمكنك الضغط علي التبويب Tab الثاني لإظهار أدواته كما هو واضح في الشكل (16-12).



الشكل (16-11) تركيب أكثر من تخطيط Combining Layouts

ملخص الفصل:

- تعلمنا في هذا الفصل كيفية تصميم نافذة البرنامج بحيث تحتوي علي أكثر من أداة باستخدام مدير التخطيط Layout Manager.
- في الفصل القادم نتعلم - بإذن الله - كيفية معالجة أحداث الأدوات Event Handling بحيث يستجيب البرنامج لأفعال المستخدم ، فتابع معنا الفصل القادم.