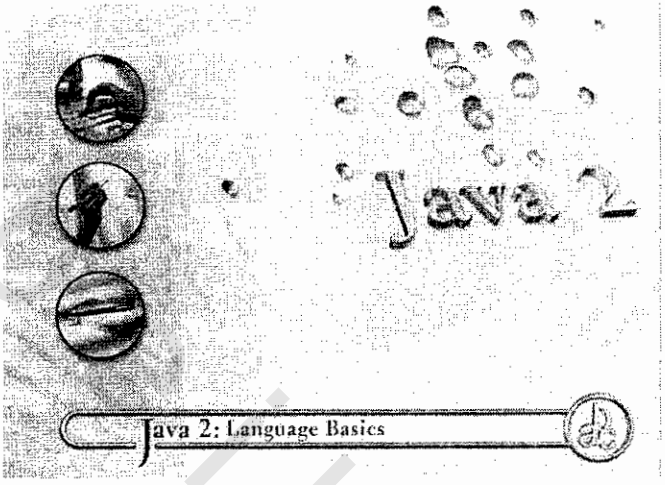
الفصل الخامس عشر



مكتبة أدوات Swing
Swing Components

فى هذا الفصل سوف نتناول مكتبة الأدوات المتقدمة المعروفة باسم (Swing Components) والتي تمكننا من بناء واجهة المستخدم الرسومية. تحتوي هذه المكتبة علي عدد كبير من الأدوات وسوف نتعرض لشرح العديد من هذه الأدوات وهي:

- أداة الإطار JFrame.
- أداة اللوحة JPanel.
- أداة العنوان JLabel.
- أداة الصور ImageIcon.
- أداة النص ذات السطر الواحد JTextField.
- أداة النص متعددة الأسطر JTextArea.
- أداة لوحة التمرير JScrollPane.
- أداة شريط التمرير JScrollBar.
- أداة قائمة الاختيارات JComboBox.
- أداة السرد JList.
- أداة زر الراديو JRadioButton.
- أداة صندوق التحقق JCheckBox.
- أداة صناديق الرسائل JOptionPane.
- أداة اللوحة المبوبة JTabbedPane.
- أداة القائمة JMenu.
- أداة اختيار ملف JFileChooser.
- أداة الزر JButton.

مقدمة:

- تطورت لغات البرمجة بشكل لافت للنظر في الآونة الأخيرة وأصبحت هناك إمكانيات عالية جداً في لغات البرمجة بحيث تقدم للمبرمجين كل ما يحتاجونه في أبسط وأسهل وأسرع شكل ممكن ، وقد انعكس هذا التقدم علي تصميم واجهة أي برنامج ، حيث أصبحت عملية تصميم الواجهة والتحكم في خصائصها عملية غاية في السهولة.
- وكون لغة Java من اللغات الحديثة جداً ، فإنها تقدم للمبرمجين مجموعة كبيرة جداً من الأدوات التي تساعد في بناء الواجهة بطريقة جذابة.
- وفي هذا الفصل نتعرف علي أهم الأدوات المستخدمة في تصميم الواجهة حيث نقوم بإنشاء كل أداة علي حدة ثم نتعرف في الفصل القادم علي كيفية وضع أكثر من أداة في الشاشة ثم نتعرف في الفصل الذي يليه علي كيفية معالجة الأحداث Events Handling التي تتيح للمستخدم التفاعل مع البرنامج كما سنتعلم في الفصول القادمة.
- سوف نتعرف في هذا الفصل علي الأدوات التالية:

1. أداة الإطار JFrame.
2. أداة اللوحة JPanel.
3. أداة العنوان JLabel.
4. أداة الصور ImageIcon.
5. أداة النص ذات السطر الواحد JTextField.
6. أداة النص متعددة الأسطر JTextArea.
7. أداة لوحة التمرير JScrollPane.
8. أداة شريط التمرير JScrollBar.
9. أداة قائمة الاختيارات JComboBox.
10. أداة السرد JList.
11. أداة زر الراديو JRadioButton.
12. أداة صندوق التحقق JCheckBox.
13. أداة صناديق الرسائل JOptionPane.

14. أداة اللوحة المبوبة JTabbedPane.
15. أداة القائمة JMenu.
16. أداة اختيار ملف JFileChooser.
17. أداة الزر JButton.

إنشاء الواجهة الرسومية GUI:

- إن عملية إدخال الأدوات في أي تطبيق تنشئه هي عملية مكررة ومتشابهة ، حيث توفر لنا لغة Java العديد من الفصائل Classes التي تمثل جميع الأدوات المتاحة ، ويجب علي المبرمج اختيار الأداة التي يريدها ثم يقوم بإنشاء هدف Object منها.
- وبعد إدخال الأداة المطلوبة ، فإنه يتم تحديد خصائص الأداة من خلال بعض الدوال Methods المتاحة لكل أداة ، فمثلاً يمكنك تغيير طول وعرض الإطار JFrame ، كما يمكن تغيير شكل ولون وحجم الخط... إلخ.
- تختلف خصائص كل أداة عن الأخري علي الرغم من وجود تشابه بين بعض الخصائص ولكن لن تختلف طريقة التحكم في هذه الخصائص من أداة لأخري.
- إذن نلخص الفقرة السابقة كالآتي:
- لإنشاء أي أداة فإننا ننشئ هدفاً Object من نوع الأداة المطلوبة ثم نقوم بتحديد خصائصها عن طريق استدعاء دالة Method لتحديد قيمة الخاصية ، وبذلك نكون قد انتهينا من إدخال أي أداة نريدها بمنتهي السهولة.
- إذن كل ما نحتاج إلي معرفته هو اسم الأداة وأهميتها ومجال استخدامها مع معرفة أهم خصائصها حتي نستطيع إظهار الأداة بالشكل المطلوب.
- سوف نبدأ بتوضيح أول أداة وهي أداة الإطار JFrame.

أداة الإطار JFrame:

- أداة الإطار JFrame هي المسئولة عن إظهار نافذة البرنامج بكل ما تحتويه من أدوات تتيح للمستخدم التفاعل مع البرنامج.
- يتم إنشاء أداة الإطار JFrame عن طريق استخدام الفصيلة class المعروفة باسم JFrame حيث نقوم بإنشاء فصيلة class عادية ثم نقوم بالوراثة Inheritance من

الفصيلة JFrame وذلك لإضافة جميع الأدوات التي نحتاجها في نافذة البرنامج.

- تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة setIconImage (Image image):

تستخدم في تغيير شكل الأيقونة Icon الموجودة في أعلي يسار الإطار JFrame.

المعاملات Parameters:

Image image: يمثل الصورة التي ستظهر في أعلي يسار الإطار JFrame.

لاستخدام هذه الدالة Method ، فإننا لا بد أن نقوم بإنشاء الصورة التي ستظهر كأيقونة Icon للإطار JFrame كالتالي:

```
Image i=Toolkit.getDefaultToolkit().createImage("1.gif");
```

حيث يتم إنشاء الصورة عن طريق وضع صورة باسم 1.gif في نفس مسار ملف كود البرنامج ويتم تحميلها في البرنامج باستخدام السطر السابق.

الدالة setTitle (String title):

تستخدم في تغيير عنوان النافذة JFrame.

المعاملات Parameters:

String title: يمثل النص الذي سيظهر في أعلي الإطار JFrame.

الدالة setSize(Dimension d):

تستخدم في تغيير حجم النافذة JFrame.

المعاملات Parameters:

Dimension d: يمثل أبعاد الإطار JFrame من خلال تحديد العرض والطول.

باستخدام هذه الدالة Method فيمكننا تحديد حجم الإطار JFrame بأكثر من طريقة كالتالي:

```
Dimension size = new Dimension(200,100);
setSize(size);
```

حيث تم إنشاء هدف Object من نوع Dimension مع تحديد الطول والعرض المطلوبين ثم استخدام الدالة setSize() كما في السطر السابق.

يمكن استخدام طريقة أخري كالتالي:

```
setSize(200,100);
```

حيث يتم تحديد الطول والعرض المطلوبين باستخدام الدالة setSize() مباشرة.

يمكن استخدام طريقة أخري كالتالي:

```
Dimension screensize =
Toolkit.getDefaultToolkit().getScreenSize();
setSize(screensize);
```

حيث تستخدم هذه الطريقة عندما نريد أن تكون شاشة البرنامج بنفس حجم شاشة الكمبيوتر أي سيظهر البرنامج ملء الشاشة Full Screen.

الدالة setCursor(Cursor cursor):

تستخدم في تغيير شكل مؤشر الفارة Mouse.

المعاملات Parameters:

Cursor cursor: يمثل شكل مؤشر الفارة Mouse المطلوب.

تحتوي لغة Java علي عدد كبير من الأشكال الجاهزة لشكل مؤشر الفارة Mouse كما يتضح من الجدول التالي:

اسم المؤشر Cursor	
S_RESIZE_CURSOR	DEFAULT_CURSOR
SE_RESIZE_CURSOR	E_RESIZE_CURSOR
SW_RESIZE_CURSOR	HAND_CURSOR
TEXT_CURSOR	MOVE_CURSOR
W_RESIZE_CURSOR	N_RESIZE_CURSOR
WAIT_CURSOR	NE_RESIZE_CURSOR
	NW_RESIZE_CURSOR

الدالة setDefaultCloseOperation(int operation):

تستخدم في تحديد الإجراء الافتراضي الذي سيحدث عند إغلاق الإطار JFrame.

المعاملات Parameters:

Int operation: يمثل الإجراء الذي سيحدث عند إغلاق الإطار JFrame حيث يمكن أن يكون واحداً من القيم التالية كما يتضح من الجدول التالي.

قيمة المعامل Parameter	المعنى
DO_NOTHING_ON_CLOSE	لا يتم فعل شيء عند إغلاق الإطار JFrame وتظل شاشة البرنامج ظاهرة
HIDE_ON_CLOSE	يتم إخفاء شاشة البرنامج عند إغلاق الإطار JFrame دون إخلاء الذاكرة من الإطار JFrame
DISPOSE_ON_CLOSE	يتم إخفاء شاشة البرنامج عند إغلاق الإطار JFrame وإخلاء الذاكرة من الإطار JFrame
EXIT_ON_CLOSE	يتم إخفاء شاشة البرنامج عند إغلاق الإطار JFrame وإخلاء الذاكرة من الإطار JFrame مع إنهاء البرنامج

الدالة setLocation(int x, int y):

تستخدم في تحديد موقع الإطار JFrame في الشاشة.

المعاملات Parameters:

int x: الإحداثي الأفقي للنقطة العليا اليسري للإطار JFrame.

int y: الإحداثي الرأسي للنقطة العليا اليسري للإطار JFrame.

الدالة setResizable(boolean resizable):

تستخدم في تحديد ما إذا كان يمكن للمستخدم تغيير حجم الإطار JFrame أم لا.

المعاملات Parameters:

boolean resizable: إذا كانت قيمة هذا المعامل Parameter تساوي true فيمكن للمستخدم عندئذ تغيير حجم الإطار JFrame ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فلا يمكن للمستخدم عندئذ أن يقوم بتغيير حجم الإطار JFrame.

الدالة setVisible(boolean b):

تستخدم في تحديد ما إذا كان الإطار JFrame سيكون ظاهراً أم لا.

المعاملات Parameters:

boolean b: إذا كانت قيمة هذا المعامل Parameter تساوي true فسيتم إظهار الإطار JFrame ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فلن يتم إظهار الإطار JFrame.

المثال التالي يوضح استخدام أداة الإطار JFrame.

مثال (1): أداة الإطار JFrame:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة الإطار Jframe مع توضيح كيفية تحديد أهم خصائصها.

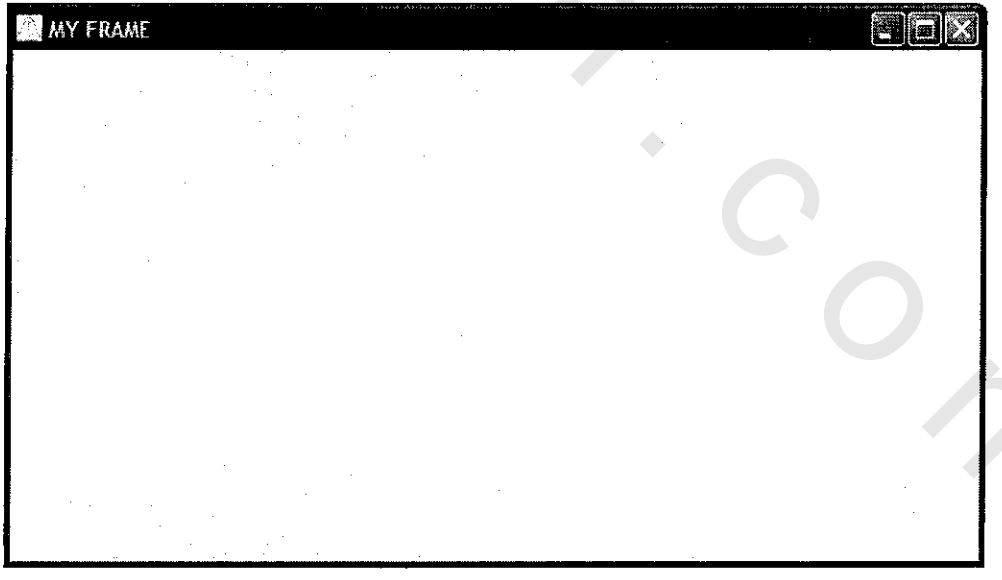
ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyFrame extends JFrame
5.  {
6.   MyFrame()
7.   {
8.      //set the icon of the frame
9.           Image
    i=Toolkit.getDefaultToolkit().createImage("1.gif");
10.          setIconImage(i);
```

```
11.
12.     //set the title
13.     setTitle("MY FRAME");
14.
15.     //set the size
16.     Dimension screensize =
    Toolkit.getDefaultToolkit().getScreenSize();
17.     setSize(screensize);
18.
19.     //set the cursor
20.     setCursor(Cursor.HAND_CURSOR);
21.
22.     //set the frame to close when press the close icon
23.
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE
    );
24.
25.     //set the frame location
26.     setLocation(300,200);
27.
28.     //prevent resizing the window
29.     setResizable(false);
30.
31.     //show the frame
32.     setVisible(true);
33.  }
34.
35.  public static void main(String args[])
36.  {
37.       MyFrame frame = new MyFrame();
38.  }
39.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج حيث نحتاج الحزمة javax.swing لأنها تحتوي علي تعريف أدوات الـ Swing components ، ونحتاج الحزمة java.awt لأنها تحتوي علي تعريف الفصيلة class المعروفة باسم Container وهي فصيلة class سنشرح عملها لاحقاً.
- في السطر رقم 4 يتم إنشاء الفصيلة MyFrame التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء دالة البناء Constructor.
- في السطور من 8 إلي 32 يتم تحديد خصائص أداة الإطار JFrame كما سبق لنا شرحه.
- في السطور من 35 إلي 38 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyFrame ، وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-1).



الشكل (15-1) أداة الإطار JFrame

أداة اللوحة JPanel:

- أداة اللوحة JPanel هي حاوية بسيطة يمكنها أن تحتوى علي بعض الأدوات Controls مثل (أداة الزر JButton) ولكنها لا يمكن أن تحتوى على أداة الإطار JFrame وذلك لأن أداة الإطار JFrame تعتبر حاوية أكبر من أداة اللوحة JPanel ، وبالطبع لا يمكنك وضع شيء كبير في شيء أصغر منه.
- ويمكنك تخيل ذلك بأنه يمكنك وضع لوحة في إطار ولكن لا يوجد معني لوضع إطار في لوحة ، وإذا حاولت فعل ذلك فلن يحدث خطأ في عملية الترجمة Compile ولكن سيحدث خطأ عند تنفيذ البرنامج Run Time Error.
- قد تكون فائدة أداة اللوحة JPanel مختفية وقد تظهر بلا فائدة ، ولكن الحقيقة أن لها فائدة كبرى كما سيتبين لنا في الفصول القادمة عندما نحتاج وضع أكثر من أداة Control في النافذة ، حيث ستظهر هذه الأداة كثيراً في الفصول القادمة.
- يتم إنشاء أداة اللوحة JPanel عن طريق إنشاء هدف Object من نوع الفصيلة JPanel.
- المثال التالي يوضح استخدام أداة اللوحة JPanel.

مثال (2): أداة اللوحة JPanel:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة اللوحة JPanel مع توضيح كيفية تحديد خصائصها.

ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyPanel extends JFrame
5. {
6.   JPanel panel;
7.
```

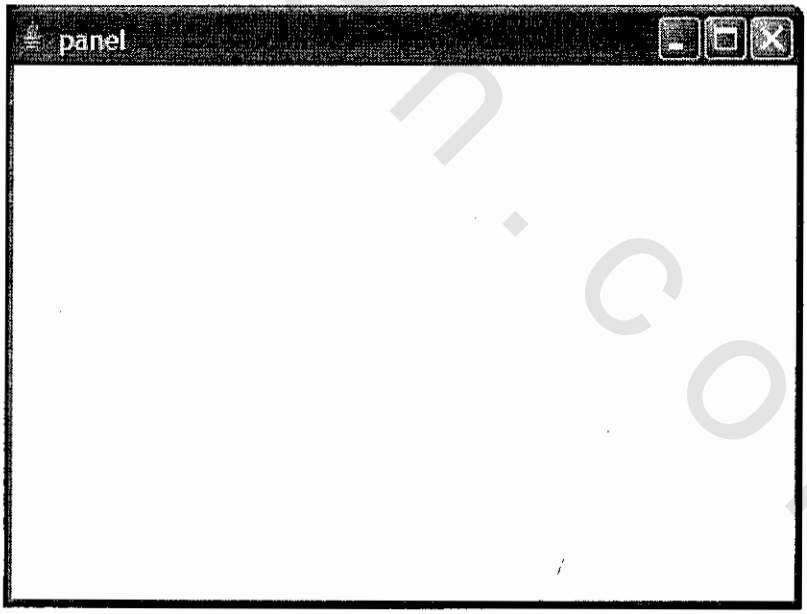
```
8.  MyPanel()
9.  {
10.     Container c = getContentPane();
11.     panel = new JPanel();
12.     c.add(panel);
13.
14.     setTitle("panel");
15.     setSize(400,300);
16.     setVisible(true);
17. }
18.
19. public static void main(String args[])
20. {
21.     MyPanel panel = new MyPanel();
22. }
23.}
```

ثالثاً: شرح الكود:

- في السطر رقم 4 يتم إنشاء الفصيلة MyPanel التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JPanel.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 10 يتم توضيح طريقة إضافة أدوات الـ Swing التي ننشأها إلي نافذة JFrame التطبيق حيث لا يمكننا إضافة الأدوات Controls إلي نافذة التطبيق مباشرة ، ولكننا ننشئ حاوية Container عن طريق استخدام الدالة getContentPane() الموجودة في الفصيلة JFrame حيث تقوم هذه الدالة Method بإرجاع هدف Object من نوع الفصيلة Container لكى نتمكن من إضافة الأدوات Controls إلي هذا المحتوي.
- ولإضافة أي أداة Control إلي الحاوية Container ، فإنك تستخدم الدالة add()

وتمرر لها اسم الأداة Control التي تريد إضافتها إلي نافذة التطبيق كما يتضح في السطر رقم 12.

- في السطر رقم 11 يتم إنشاء هدف Object من نوع JPanel والذي يمثل أداة اللوحة JPanel.
- في السطور من 14 إلي 16 يتم تحديد خصائص أداة الإطار JFrame كما سبق لنا شرحه.
- في السطور من 19 إلي 22 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyPanel وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (2-15).



الشكل (2-15) أداة اللوحة JPanel

أداة العنوان JLabel:

- تستخدم هذه الأداة لوضع صورة أو نص ثابت في التطبيق بحيث لا يكون بإمكان المستخدم التعديل في هذا النص أو التفاعل معه أثناء عمل البرنامج ، أي أن أداة العنوان JLabel هي أداة للقراءة فقط وتستخدم لإعطاء معلومة للمستخدم عن البيانات المطلوب إدخالها.
- فمثلاً إذا أنشأت تطبيقاً يطلب من المستخدم إدخال اسمه ، إذن فأنت تحتاج لوضع كلمة الاسم كنص ثابت في التطبيق لبيان أنك تريد من المستخدم إدخال اسمه ولا يحتاج المستخدم أن يعدل في هذا النص.
- تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء Jlabel():

تقوم بإظهار أداة عنوان JLabel بلا أي نص عليها وبدون ظهور صورة عليها.

دالة البناء JLabel(Icon image):

تقوم بإظهار أداة عنوان JLabel بلا أي نص عليها ولكن تظهر عليها صورة.

المعاملات Parameters:

Icon image: تمثل الصورة التي ستظهر علي أداة العنوان JLabel.

دالة البناء JLabel(Icon image, int horizontalAlignment):

تقوم بإظهار أداة عنوان JLabel بلا أي نص عليها ولكن تظهر عليها صورة مع تحديد المحاذاة الأفقية Horizontal Alignment للنص.

المعاملات Parameters:

Icon image: تمثل الصورة التي ستظهر علي أداة العنوان JLabel.

int horizontalAlignment: المحاذاة الأفقية Horizontal Alignment للنص التي يمكن أن تكون إحدي القيم التالية:

- JLabel.LEFT: تبين أن النص سيتم محاذاته علي اليسار.
- JLabel.RIGHT: تبين أن النص سيتم محاذاته علي اليمين.
- JLabel.CENTER: تبين أن النص سيتم محاذاته في المنتصف.

دالة البناء JLabel(String text):

تقوم بإظهار أداة عنوان JLabel ويظهر عليها نص.

المعاملات Parameters:

String text: النص الذي سيظهر علي أداة العنوان JLabel.

دالة البناء JLabel(String text, Icon icon, int horizontalAlignment):

تقوم بإظهار أداة عنوان JLabel ويظهر عليها نص وصورة مع تحديد وضع النص بالنسبة للصورة ممثله في القيمة الثالثة والتي يمكن أن تكون واحدة من القيم التالية:

- JLabel.LEFT: تبين أن النص سيظهر علي يسار الصورة.
- JLabel.RIGHT: تبين أن النص سيظهر علي يمين الصورة.
- JLabel.CENTER: تبين أن النص سيظهر علي الصورة في منتصفها.

دالة البناء JLabel(String text, int horizontalAlignment):

تقوم بإظهار أداة عنوان JLabel ويظهر عليها نص عليها مع تحديد المحاذاة الأفقية Horizontal Alignment للنص.

المعاملات Parameters:

String text: تمثل الصورة التي ستظهر علي أداة العنوان JLabel.

int horizontalAlignment: المحاذاة الأفقية Horizontal Alignment للنص التي يمكن أن تكون إحدي القيم التالية:

- JLabel.LEFT: تبين أن النص سيتم محاذاته علي اليسار.
- JLabel.RIGHT: تبين أن النص سيتم محاذاته علي اليمين.
- JLabel.CENTER: تبين أن النص سيتم محاذاته في المنتصف.

كما تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة setText(String text):

تستخدم في تغيير النص الذي يظهر علي أداة العنوان JLabel.

المعاملات Parameters:

String text: يمثل النص الذي يظهر علي أداة العنوان JLabel.

الدالة setIcon(Icon icon):

تستخدم في تغيير شكل الأيقونة Icon التي تظهر علي أداة العنوان JLabel.

المعاملات Parameters:

Icon icon: يمثل الصورة التي ستظهر علي أداة العنوان JLabel.

لاستخدام هذه الدالة Method ، فإننا لا بد أن نقوم بإنشاء الصورة التي ستظهر كأيقونة Icon للإطار JFrame كالتالي:

```
Icon icon = new ImageIcon("1.gif");
```

حيث يتم إنشاء الصورة عن طريق وضع صورة باسم gif.1 في نفس مسار ملف كود البرمجة ويتم تحميلها في البرنامج باستخدام السطر السابق.

الدالة setBackground(Color bg):

تستخدم لتغيير لون خلفية أداة العنوان JLabel.

المعاملات Parameters:

Color bg: لون الخلفية المطلوب.

تحتوي لغة Java علي عدد كبير من الألوان الجاهزة كما يتضح من الجدول التالي:

اسم اللون Color	
BLACK	DARK_GRAY
CYAN	BLUE
GREEN	GRAY
MAGENTA	LIGHT_GRAY
PINK	ORANGE
WHITE	RED
	YELLOW

الدالة setForeground(Color fg):

تستخدم لتغيير لون النص الذي يظهر علي أداة العنوان JLabel.

المعاملات Parameters:

Color fg: لون النص المطلوب.

الدالة setFont(Font font):

تستخدم في تغيير شكل الخط للنص الذي يظهر علي أداة العنوان JLabel.

المعاملات Parameters:

Font font: شكل الخط المطلوب.

يمكننا إنشاء خط عن طريق الفصيلة Font التي تستقبل ثلاثة معاملات Parameters كالتالي:

المعامل الأول: يمثل شكل الخط.

المعامل الثاني: يمثل نمط الخط Style كأن يكون عريض Bold أو مائل Italic أو عادي Plain.

المعامل الثالث: يمثل حجم الخط المطلوب.

أي يمكننا إنشاء خط كالتالي:

```
Font font = new Font("arial",Font.BOLD,14);
```

حيث سيكون هذا الخط بالخصائص التالية:

شكل الخط: arial.

نمط الخط Style: عريض Bold.

حجم الخط: 14.

الدالة setToolTipText(String text):

تستخدم في تحديد تلميح Tool Tip للأداة عندما يقف المستخدم بمؤشر الفارة Mouse علي الأداة لمدة ثواني.

المعاملات Parameters:

String text: النص الذي سيظهر كتلميح Tool Tip.

الدالة setOpaque(boolean isOpaque):

تستخدم في تحديد ما إذا كانت الأداة ستظهر شفافة أم لا.

المعاملات Parameters:

boolea is Opaque: إذا كانت قيمة هذا المعامل Parameter تساوي true فإن الأداة ستظهر معتمة ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فإن الأداة ستظهر شفافة.

المثال التالي يوضح استخدام أداة العنوان JLabel.

مثال (3): أداة العنوان JLabel:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة العنوان JLabel مع توضيح كيفية تحديد أهم خصائصها.

ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyLabel extends JFrame
5.  {
6.   JLabel label;
7.
8.   MyLabel()
9.   {
10.      Container container = getContentPane();
11.
12.      label = new JLabel();
13.      label.setText("Name");
14.      Icon icon = new ImageIcon("1.gif");
15.      label.setIcon(icon);
16.      label.setBackground(Color.blue);
17.      label.setForeground(Color.yellow);
```

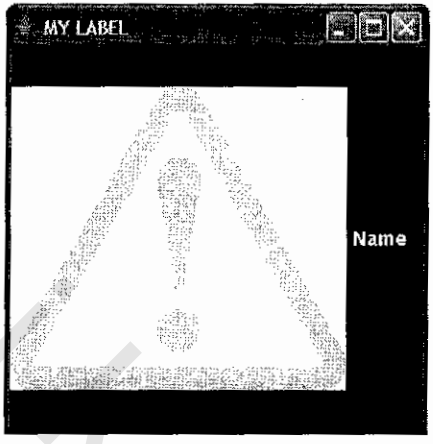
```
18.       Font font = new Font("arial",Font.BOLD,14);
19.       label.setFont(font);
20.       label.setToolTipText("this is a name");
21.       label.setOpaque(true);
22.       container.add(label);
23.
24.       setTitle("MY LABEL");
25.       setSize(300,300);
26.
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE
    );
27.       setVisible(true);
28.  }
29.  public static void main(String args[])
30.  {
31.       MyLabel myLabel = new MyLabel();
32.  }
33.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyLabel التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JLabel.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 12 يتم إنشاء هدف Object من نوع JLabel والذي يمثل أداة العنوان JLabel.
- في السطور من 13 إلي 27 يتم تحديد خصائص أداة العنوان JLabel وأداة الإطار JFrame كما سبق لنا شرحه.
- في السطور من 30 إلي 33 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند

تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyLabel وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.

قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-3).



الشكل (15-3) أداة العنوان JLabel

أداة الصور ImageIcon:

تستخدم هذه الأداة لوضع الصور فى التطبيق أو علي بعض الأدوات مثل أداة الزر JButton وأداة العنوان JLabel.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء ImageIcon(String filename):

تقوم بتحميل الصورة باستخدام اسم ملف الصورة.

المعاملات Parameters:

String filename: اسم ملف الصورة.

المثال التالي يوضح استخدام أداة الصور ImageIcon.

مثال (4): أداة الصور ImageIcon:

أولاً: هدف المثال:

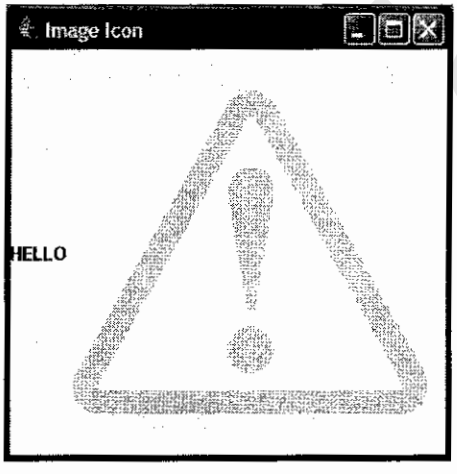
نريد أن نبين كيفية استخدام أداة الصور ImageIcon لإظهار الصور في البرنامج.

ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyImageIcon extends JFrame
5. {
6.  JLabel label;
7.  ImageIcon icon;
8.
9.  MyImageIcon()
10. {
11.      Container c = getContentPane();
12.      label = new JLabel("HELLO");
13.      icon = new ImageIcon("1.gif");
14.      label.setIcon(icon);
15.      label.setHorizontalTextPosition(JLabel.LEFT);
16.      c.add(label);
17.
18.   setTitle("Image Icon");
19.      setSize(300,300);
20.
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE
    );
21.      setVisible(true);
22. }
23.
24. public static void main (String args[])
25. {
26.      MyImageIcon image=new MyImageIcon();
27. }
28.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyImageIcon التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JLabel.
- في السطر رقم 7 يتم إنشاء متغير من نوع ImageIcon.
- في السطر رقم 9 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 12 يتم إنشاء هدف Object من نوع JLabel والذي يمثل أداة العنوان JLabel.
- في السطر رقم 13 يتم إنشاء هدف Object من نوع ImageIcon والذي يمثل أداة الصور ImageIcon.
- في السطرين 14 و 15 يتم تحديد خصائص أداة العنوان JLabel لإظهار الصورة علي الأداة.
- في السطور من 24 إلي 27 يتم إنشاء الدالة الرئيسية main() التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyImageIcon وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-4).



الشكل (15-4) أداة الصور ImageIcon

أداة النص ذات السطر الواحد JTextField:

- تسمح أداة النص JTextField للمستخدم بإدخال حروف من لوحة المفاتيح Keyboard ، ولذلك تعتبر هذه الأداة من الأدوات الرئيسية التي تتيح للمستخدم إدخال معلومات معينة للتطبيق فمثلاً قد تريد من المستخدم إدخال اسمه ، ولذلك نحتاج إلي أداة النص JTextField لكي تسمح للمستخدم بأن يقوم بإدخال البيانات التي يريدها.
- يطلق علي الأداة JTextField أداة النص ذات السطر الواحد لأنها لا تسمح للمستخدم بكتابة أكثر من سطر واحد ، فإذا كان عدد الحروف المكتوبة أكبر من حجم أداة النص JTextField فإن الكتابة تستمر ولكن تختفى الحروف الأولى من النص ويمكن التحرك بالأسهم للرجوع للخلف أو التقدم للأمام في النص.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JTextField():

تقوم بإظهار أداة نص JTextField بلا أي نص افتراضي في الأداة.

دالة البناء JTextField(int columns):

تقوم بإظهار أداة نص JTextField مع تحديد عرض أداة النص JTextField.

المعاملات Parameters:

int columns: عرض أداة النص JTextField.

دالة البناء JTextField(String text):

تقوم بإظهار أداة نص JTextField مع وجود نص افتراضي في الأداة.

المعاملات Parameters:

String text: النص الافتراضي الذي سيظهر في أداة النص JTextField.

دالة البناء JTextField(String text, int columns):

تقوم بإظهار أداة نص JTextField مع وجود نص افتراضي في الأداة ومع تحديد عرض أداة النص JTextField.

المعاملات Parameters:

String text: النص الافتراضي الذي سيظهر في أداة النص JTextField.

int columns: عرض أداة النص JTextField.

المثال التالي يوضح استخدام أداة النص JTextField.

مثال (5): أداة النص ذات السطر الواحد JTextField:

أولاً: هدف المثال:

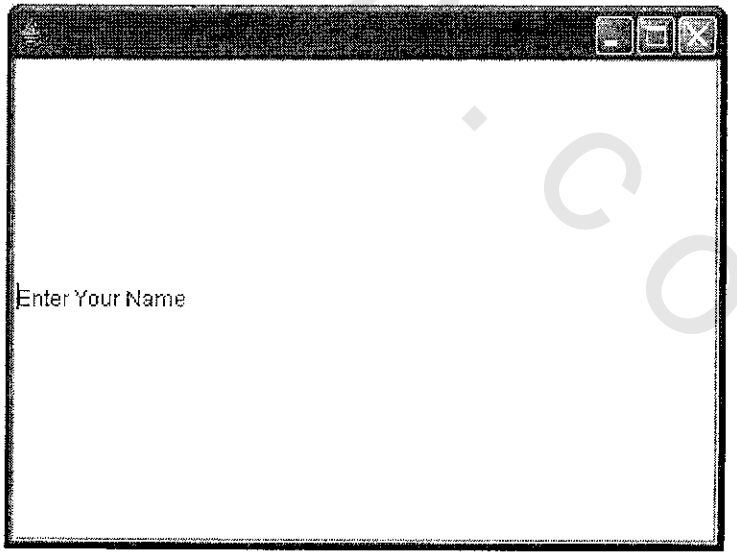
نريد أن نبين كيفية إظهار أداة النص JTextField في البرنامج.

ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyTextField extends JFrame
5.  {
6.   JTextField text;
7.
8.   MyTextField()
9.   {
10.       Container c = getContentPane();
11.       text = new JTextField("Enter Your Name");
12.       c.add(text);
13.
14.       setSize(400,300);
15.       setVisible(true);
16.  }
17.
18.  public static void main (String args[])
19.  {
20.       MyTextField text = new MyTextField();
21.  }
22.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyTextField التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JTextField.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم إنشاء هدف Object من نوع JTextField والذي يمثل أداة النص JTextField.
- في السطور من 18 إلي 21 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyTextField وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (5-15).



الشكل (5-15) أداة النص JTextField

أداة النص متعددة الأسطر JTextArea:

- أداة النص متعددة الأسطر JTextArea هي أداة نص مثل أداة النص ذات السطر الواحد JTextField مع الاختلاف –كما هو واضح من الاسم – أن الأداة JTextArea تتميز بإمكانية كتابة نص يتكون من أكثر من سطر ، وللانتقال إلي سطر جديد يمكنك الضغط علي زر الإدخال (Enter).
- تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JTextArea():

تقوم بإظهار أداة نص JTextArea بلا أي نص افتراضي في الأداة.

دالة البناء JTextArea(int rows, int columns):

تقوم بإظهار أداة نص JTextArea مع تحديد عرض وارتفاع أداة النص JTextArea.

المعاملات Parameters:

int rows: ارتفاع أداة النص JTextArea.

int columns: عرض أداة النص JTextArea.

دالة البناء JTextArea(String text):

تقوم بإظهار أداة نص JTextArea مع وجود نص افتراضي في الأداة.

المعاملات Parameters:

String text: النص الافتراضي الذي سيظهر في أداة النص JTextArea.

دالة البناء JTextField(String text, int rows, int columns):

تقوم بإظهار أداة نص JTextArea مع وجود نص افتراضي في الأداة ومع تحديد عرض وارتفاع أداة النص JTextArea.

المعاملات Parameters:

String text: النص الافتراضي الذي سيظهر في أداة النص JTextArea.

int rows: ارتفاع أداة النص JTextArea.

int columns: عرض أداة النص JTextArea.

المثال التالي يوضح استخدام أداة النص JTextArea.

مثال (6): أداة النص متعددة الأسطر JTextArea:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة النص JTextArea في البرنامج.

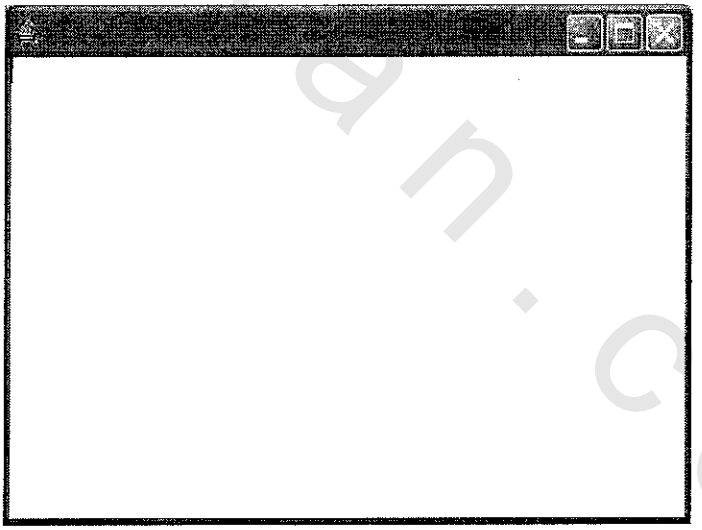
ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyTextArea extends JFrame
5.  {
6.    JTextArea text;
7.
8.    MyTextArea()
9.    {
10.       Container c = getContentPane();
11.       text = new JTextArea();
12.       c.add(text);
13.
14.       setSize(400,300);
15.       setVisible(true);
16.   }
17.
18.  public static void main (String args[])
19.  {
20.       MyTextArea text = new MyTextArea();
21.  }
22.}
```

ثالثاً: شرح الكود:

في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.

- في السطر رقم 4 يتم إنشاء الفصيلة MyTextArea التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JTextArea.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم إنشاء هدف Object من نوع JTextArea والذي يمثل أداة النص JTextArea.
- في السطور من 18 إلي 21 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyTextArea وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (6-15).



الشكل (6-15) أداة النص JTextArea

أداة لوحة التمرير JScrollPane:

- نلاحظ وجود مشكلة في أداة النص متعددة الأسطر JTextArea وأداة النص ذات السطر الواحد JTextField تتمثل في أنه إذا كتبت كلاماً أكبر من حجم أداة النص

فلن يحدث تمرير scrolling أي لن يظهر لك شريط تستطيع من خلاله الرجوع للخلف أو التقدم للأمام في حالة إذا ما كان النص المكتوب أكبر من حجم أداة النص ، ولذلك نستخدم الفصيلة JScrollPane لوضع أداة النص في لوحة تمرير JScrollPane بحيث يظهر شريط تمرير scroll bar في أداة النص.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JScrollPane(Component view):

تقوم بإظهار أداة لوحة تمرير JScrollPane للأداة المطلوبة.

المعاملات Parameters:

Component view: الأداة المطلوب إظهار لوحة التمرير JScrolPane فيها.

كما تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة setHorizontalScrollBarPolicy(int policy):

تستخدم في تحديد وقت ظهور شريط التمرير الأفقي Horizontal Scroll Bar.

المعاملات Parameters:

int policy: يمثل وقت ظهور شريط التمرير الأفقي Horizontal Scroll Bar الذي يمكن أن يكون إحدي القيم التالية:

- ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED: وهو يقوم بإظهار شريط التمرير Scroll Bar الأفقي عند الحاجة فقط ، بمعني أنه إذا كان النص لا يتخطي حدود أداة النص ، فإن شريط التمرير Scroll Bar الأفقي لن يظهر ، أما إذا كان النص يتخطي حدود أداة النص ، فإن شريط التمرير Scroll Bar الأفقي سيظهر.
- ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER: وهو يقوم بإلغاء ظهور شريط التمرير Scroll Bar الأفقي مهما كان حجم النص.

- JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS: وهـــو يقوم بإظهار شريط التمرير Scroll Bar الأفقي مهما كان حجم النص.

الدالة setVerticalScrollBarPolicy(int policy):

تستخدم في تحديد وقت ظهور شريط التمرير الرأسي Vertical Scroll Bar.

المعاملات Parameters:

int policy: يمثل وقت ظهور شـريط التمريـر الرأسـي Vertical Scroll Bar الـذي يمكن أن يكون إحدي القيم التالية:

- ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED : وهو يقوم بإظهار شريط التمرير Scroll Bar الرأسي عند الحاجة فقط ، بمعني أنه إذا كان النص لا يتخطي حـدود أداة الـنص ، فـإن شـريط التمريـر Scroll Bar الرأسي لن يظهر ، أما إذا كان النص يتخطي حدود أداة النص ، فإن شريط التمرير Scroll Bar الرأسي سيظهر.
- ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER: وهو يقوم بإلغاء ظهور شريط التمرير Scroll Bar الرأسي مهما كان حجم النص.
- JScrollPane.VERTICAL_SCROLLBAR_ALWAYS: وهـو يقـوم بإظهار شريط التمرير Scroll Bar الرأسي مهما كان حجم النص.
- المثال التالي يوضح استخدام أداة لوحة التمرير JScrollPane.

مثال (7): أداة لوحة التمرير JScrollPane:

أولاً: هدف المثال:

نريد أن نـبين كيفيـة إظهـار أداة لوحـة التمريـر JScrollPane في البرنـامج لأداة نـص متعددة الأسطر JTextArea.

ثانياً: كود البرمجة:

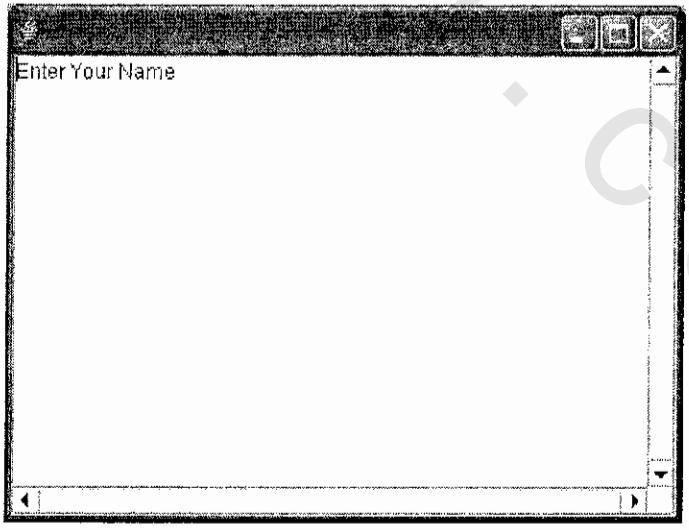
```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyScrollPane extends JFrame
5. {
6.  JTextArea text;
7.  JScrollPane scroll;
8.
9.  MyScrollPane()
10. {
11.     Container c = getContentPane();
12.     text = new JTextArea("Enter Your Name");
13.     scroll = new JScrollPane(text);
14.     scroll.setHorizontalScrollBarPolicy(
    JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
15.     scroll.setVerticalScrollBarPolicy(
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
16.     c.add(scroll);
17.
18.     setSize(400,300);
19.     setVisible(true);
20. }
21.
22. public static void main (String args[])
23. {
24.     MyScrollPane scroll = new MyScrollPane();
25. }
26.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyScrollPane التي ترث من الفصيلة

JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.

- في السطر رقم 6 يتم إنشاء متغير من نوع JTextArea.
- في السطر رقم 7 يتم إنشاء متغير من نوع JScrollPane.
- في السطر رقم 9 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 12 يتم إنشاء هدف Object من نوع JTextArea والذي يمثل أداة النص JTextArea.
- في السطر رقم 13 يتم إنشاء هدف Object من نوع JScrollPane والذي يمثل لوحة التمرير JScrollPane.
- في السطرين 14 و 15 يتم تحديد خصائص أداة لوحة التمرير JScrollPane كما شرحنا سابقاً.
- في السطور من 22 إلي 25 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyScrollPane وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-7).



الشكل (15-7) أداة لوحة التمرير JScrollPane

أداة شريط التمرير JScrollBar:

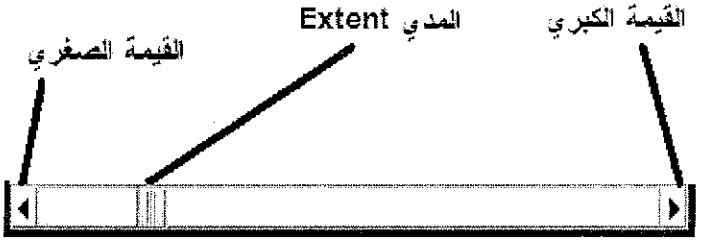
- تتيح لنا أداة شريط التمرير JScrollBar عمل تمرير scrolling للتنقل في الشاشة سواء لأعلي أو لأسفل أو لليمين أو للشمال ، فمثلاً إذا كنت تكتب نصاً وكان حجم الكتابة كبيراً ، فإن السطور الأولي ستختفي وبالتالي ستحتاج إلي أن تتحرك في الصفحة لبدايتها. هنا نحتاج إلي أداة التمرير JScrollBar.
- تذكر أن أداة لوحة التمرير JScrollPane استخدمناها لنفس الغرض ولكن الاختلاف بين لوحة التمرير JScrollPane وشريط التمرير JScrollBar أن عملية التمرير scrolling في لوحة التمرير JScrollPane تتم بدون أن تكتب لها كود برمجة خاص وذلك بعكس عملية التمرير scrolling في شريط التمرير JScrollBar والتي لا بد أن تكتب لها كود برمجة لعملية التمرير scrolling بنفسك. أيضاً يمكننا في شريط التمرير JScrollBar وضع قيمة صغري وعظمي لمتغير ما ، ويمكن تغيير قيمة هذا المتغير عن طريق تحريك شريط التمرير JScrollBar ما بين القيمتين الصغري والعظمي.

مكونات شريط التمرير JScrollBar:

يتكون شريط التمرير JScrollBar من الخصائص التالية:

- القيمة الصغري: وهي تمثل القيمة عندما يكون شريط التمرير JScrollBar في أدني قيمة له.
- القيمة العظمي: وهي تمثل القيمة عندما يكون شريط التمرير JScrollBar في أعلي قيمة له.
- القيمة الافتراضية: وهي تمثل قيمة شريط التمرير JScrollBar عند فتح البرنامج.
- المدي Extent: وهو يمثل حجم المربع الموجود في شريط التمرير JScrollBar الذي يستخدم في تغيير قيمته عند تحريكه.

الشكل (15-8) يوضح هذه المكونات.



الشكل (15-8) شريط التمرير JScrollBar

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JScrollBar():

تقوم بإظهار أداة شريط تمرير JScrollBar بالقيم الافتراضية التالية:

القيمة الصغري = 0

القيمة العظمي = 100

القيمة الافتراضية = 0

المدي = 10

دالة البناء JScrollBar(int orientation):

تقوم بإظهار أداة شريط تمرير JScrollBar بالقيم الافتراضية التالية:

القيمة الصغري = 0

القيمة العظمي = 100

القيمة الافتراضية = 0

المدي = 10

المعاملات Parameters:

int orientation: اتجاه شريط التمرير JScrollBar الذي يمكن أن يكون إحدي القيم التالية:

- JScrollBar.HORIZONTAL: وهو ينشئ شريط تمرير JScrollBar أفقي.
- JScrollBar.VERTICAL: وهو ينشئ شريط تمرير JScrollBar رأسي.

دالة البناء JScrollBar(int orientation, int value, int extent, int min, int max):

تقوم بإظهار أداة شريط تمرير JScrollBar بالقيم المحددة من المعاملات Parameters.

المعاملات Parameters:

int orientation: اتجاه شريط التمرير JScrollBar الذي يمكن أن يكون إحدي القيم التالية:

- JScrollBar.HORIZONTAL: وهو ينشئ شريط تمرير JScrollBar أفقي.
- JScrollBar.VERTICAL: وهو ينشئ شريط تمرير JScrollBar رأسي.

int value: تمثل القيمة الافتراضية لشريط التمرير JScrollBar.

int extent: تمثل حجم المدي Extent لشريط التمرير JScrollBar.

int min: تمثل القيمة الصغري لشريط التمرير JScrollBar.

int max: تمثل القيمة العظمي لشريط التمرير JScrollBar.

المثال التالي يوضح استخدام أداة شريط التمرير JScrollBar.

مثال (8): أداة شريط التمرير JScrollBar:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة شريط التمرير JScrollBar في البرنامج.

ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyScrollBar extends JFrame
5. {
6.  JScrollBar scroll;
7.
8.  MyScrollBar()
```

```
9.  {
10.     Container c = getContentPane();
11.     scroll = new
    JScrollBar(JScrollBar.HORIZONTAL,32,10,0,265);
12.     c.add(scroll);
13.
14.     setTitle("Scroll Bar");
15.     setSize(400,50);
16.     setVisible(true);
17. }
18.
19. public static void main(String args[])
20. {
21.     MyScrollBar scroll = new MyScrollBar();
22. }
23.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyScrollBar التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JScrollBar.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم إنشاء هدف Object من نوع JScrollBar والذي يمثل أداة شريط التمرير JScrollBar.
- في السطور من 19 إلي 22 يتم إنشاء الدالة الرئيسية main() التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyScrollBar وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (9-15).

الشكل (15-9) أداة شريط التمرير JScrollBar

ملحوظة:

إذا أردت أن تنشئ شريط تمرير JScrollBar لاختيار قيمة من صفر إلي 255 وكان حجم المدي extent يساوي 10 فإننا نكتب:

```
JScrollBar myScrollBar = new JScrollBar (JScrollBar.
HORIZONTAL,0,10,0,265);
```

حيث أضفنا قيمة المدي extent على القيمة العظمي التي نريدها للمتغير الذي يمثله شريط التمرير JScrollBar لنحصل على القيمة الخامسة التى نمررها لدالة البناء Constructor.

في هذا المثال فإن القيمة العظمي التي نريدها للمتغير هي 255 وحجم المدي extent حددناه بـ 10 ، إذن تكون القيمة الخامسة التي نمررها لدالة البناء Constructor هي مجموعهما أي 255+10 والتي تساوي 265 كما كتبناها في المثال.

أداة قائمة الاختيارات JComboBox:

تتيح لنا أداة قائمة الاختيارات ComboBox اختيار قيمة واحدة فقط من عدة اختيارات متاحة ، فمثلاً إذا كنت تريد من المستخدم إدخال الدولة التي ولد فيها ، فتستطيع إنشاء قائمة اختيارات تحتوي علي جميع أسماء الدول وعلي المستخدم أن يختار قيمة واحدة فقط من الاختيارات المتاحة وليس من حقه اختيار أي قيمة أخري غير موجودة في القائمة ، أي أن المستخدم محدود بالقيم التي نحددها في قائمة الاختيارات.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JComboBox():

تقوم بإظهار أداة قائمة اختيارات JComboBox خالية.

دالة البناء JComboBox(Object[] items):

تقوم بإظهار أداة قائمة اختيارات JComboBox مكونة من المصفوفة Array الموجودة في المعامل Parameter.

المعاملات Parameters:

- Object[] items: عبارة عن مصفوفة Array تمثل قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

دالة البناء JComboBox(Vector items):

تقوم بإظهار أداة قائمة اختيارات JComboBox مكونة من المتجه Vector الموجود في المعامل Parameter.

المعاملات Parameters:

Vector items: عبارة عن متجه Vector يمثل قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

كما تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة addItem(Object anObject):

تستخدم في إضافة عنصر جديد إلي قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

المعاملات Parameters:

Object anObject: يمثل محتويات العنصر الجديد الذي ستتم إضافته في أداة قائمة الاختيارات JComboBox.

الدالة setSelectedIndex(int anIndex):

تستخدم في تحديد العنصر المختار من قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

المعاملات Parameters:

int anIndex: يمثل ترتيب العنصر المختار من قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

الدالة setSelectedItem(Object anObject):

تستخدم في تحديد العنصر المختار من قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

المعاملات Parameters:

Object anObject: يمثل قيمة العنصر المختار من قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

الدالة setEditable(boolean aFlag):

تستخدم في تحديد ما إذا كان من حق المستخدم كتابة عنصر جديد غير موجود في قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

المعاملات Parameters:

boolean aFlag: إذا كانت قيمة هذا المعامل Parameter تساوي true فيمكن للمستخدم عندئذ كتابة عنصر جديد غير موجود في قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فلا يمكن للمستخدم عندئذ أن يقوم بكتابة عنصر جديد غير موجود في قائمة الاختيارات التي تظهر في أداة قائمة الاختيارات JComboBox.

المثال التالي يوضح استخدام أداة قائمة الاختيارات JComboBox.

مثال (9): أداة قائمة الاختيارات JComboBox:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة قائمة الاختيارات JComboBox في البرنامج.

ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyComboBox extends JFrame
5. {
6.  JComboBox combo;
7.
8.  MyComboBox()
9.  {
10.      Container c = getContentPane();
11.
12.      combo = new JComboBox();
13.      combo.addItem("egypt");
14.      combo.addItem("england");
15.      combo.addItem("lebanon");
16.      combo.addItem("italy");
17.      combo.setSelectedItem("italy");
18.      combo.setEditable(true);
19.      c.add(combo);
20.
21.      setTitle("combo");
22.      setSize(50,50);
23.      setVisible(true);
24. }
25.
26. public static void main(String args[])
27. {
28.      MyComboBox combo = new MyComboBox();
29. }
30.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyComboBox التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JComboBox.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 12 يتم إنشاء هدف Object من نوع JComboBox والذي يمثل أداة قائمة الاختيارات JComboBox.
- في السطور من 13 إلي 16 يتم إضافة عناصر جديدة إلي قائمة الاختيارات باستخدام الدالة addItem( ).
- في السطرين 17 و 18 يتم تحديد خصائص أداة قائمة الاختيارات JComboBox كما سبق لنا شرحه.
- في السطور من 26 إلي 29 يتم إنشاء الدالة الرئيسية main() التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyComboBox وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-10).

الشكل (15-10) أداة قائمة الاختيارات JComboBox

أداة السرد JList:

- تتيح لنا أداة السرد JList اختيار قيمة واحدة فقط من عدة اختيارات متاحة ، فمثلاً إذا كنت تريد من المستخدم إدخال الدولة التي ولد فيها ، فتستطيع إنشاء قائمة اختيارات تحتوي علي جميع أسماء الدول وعلي المستخدم أن يختار قيمة واحدة فقط

من الاختيارات المتاحة وليس من حقه اختيار أي قيمة أخري غير موجودة في القائمة ، أي أن المستخدم محدود بالقيم التي نحددها في قائمة الاختيارات.

- تتشابه هذه الأداة مع أداة قائمة الاختيارات JComboBox ولكنها تختلف في أنها تسمح للمستخدم باختيار قيمة واحدة فقط من قائمة الاختيارات أو عدة قيم متتابعة أو غير متتابعة.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JList():

تقوم بإظهار أداة السرد JList خالية.

دالة البناء JList(Object[] listData):

تقوم بإظهار أداة السرد JList مكونة من المصفوفة Array الموجودة في المعامل Parameter.

المعاملات Parameters:

- Object[] listData: عبارة عن مصفوفة Array تمثل قائمة الاختيارات التي تظهر في أداة السرد JList.

دالة البناء JList(Vector listData):

تقوم بإظهار أداة السرد JList مكونة من المتجه Vector الموجود في المعامل Parameter.

المعاملات Parameters:

Vector listData: عبارة عن متجه Vector يمثل قائمة الاختيارات التي تظهر في أداة السرد JList.

كما تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة setSelectionMode(int selectionMode):

تستخدم في تحديد ما إذا كان بإمكان المستخدم اختيار قيمة واحدة أو أكثر من قائمة الاختيارات التي تظهر في أداة السرد JList.

المعاملات Parameters:

int selectionMode: يمثل متغير يحدد ما إذا كان بإمكان المستخدم اختيار قيمة واحدة أو أكثر من قائمة الاختيارات التي تظهر في أداة السرد JList. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- ListSelectionModel.SINGLE_SELECTION: يستخدم لتحديد أن المستخدم من حقه اختيار قيمة واحدة فقط من قائمة الاختيارات التي تظهر في أداة السرد JList.
- ListSelectionModel.SINGLE_INTERVAL_SELECTION:
  يستخدم لتحديد أن المستخدم من حقه اختيار أكثر من قيمة متتابعة من قائمة الاختيارات التي تظهر في أداة السرد JList.
- ListSelectionModel.MULTIPLE_INTERVAL_SELECTION:
  يستخدم لتحديد أن المستخدم من حقه اختيار أكثر من قيمة متتابعة أو غير متتابعة من قائمة الاختيارات التي تظهر في أداة السرد JList.

الدالة setSelectionBackground(Color selectionBackground):

تستخدم في تحديد لون الخلفية Background للعنصر المختار من قائمة الاختيارات التي تظهر في أداة السرد JList.

المعاملات Parameters:

Color selectionBackground: يمثل لون الخلفية المطلوب.

الدالة setSelectionForeground(Color selectionForeground):

تستخدم في تحديد لون الخط للعنصر المختار من قائمة الاختيارات التي تظهر في أداة السرد JList.

المعاملات Parameters:

Color selectionForeground: يمثل لون الخط المطلوب.

الدالة setBackground(Color bg):

تستخدم في تحديد لون الخلفية Background لقائمة الاختيارات التي تظهر في أداة السرد JList.

المعاملات Parameters:

Color bg: يمثل لون الخلفية المطلوب.

الدالة setForeground(Color fg):

تستخدم في تحديد لون الخط لقائمة الاختيارات التي تظهر في أداة السرد JList.

المعاملات Parameters:

Color fg: يمثل لون الخط المطلوب.

المثال التالي يوضح استخدام أداة السرد JList.

مثال (10): أداة السرد JList:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة السرد JList في البرنامج.

ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyList extends JFrame
5. {
6.  JList list;
7.
8.  MyList()
9.  {
10.      Container c = getContentPane();
11.
12.      String[] names =
   {"ahmed","mohamed","mostafa","mahmoud"};
13.
14.      list = new JList(names);
15.
```

```
        list.setSelectionMode(ListSelectionModel.MULTIPLE_I
    NTERVAL_SELECTION);
16.         list.setSelectionBackground(Color.RED);
17.         list.setSelectionForeground(Color.PINK);
18.         list.setBackground(Color.YELLOW);
19.         list.setForeground(Color.BLUE);
20.         list.setSelectedIndex(1);
21.         c.add(list);
22.
23.         setTitle("List");
24.         setSize(50,50);
25.         setVisible(true);
26.   }
27.
28.   public static void main(String args[])
29.   {
30.         MyList list = new MyList();
31.   }
32.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyList التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JList.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطور رقم 12 يتم إنشاء مصفوفة Array تحتوي علي عناصر قائمة الاختيارات في أداة السرد JList.
- في السطر رقم 14 يتم إنشاء هدف Object من نوع JList والذي يمثل أداة السرد JList.

- في السطور من 15 إلي 20 يتم تحديد خصائص أداة السرد JList كما سبق لنا شرحه.
- 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 MyList وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (11-15).

الشكل (11-15) أداة السرد JList

أداة زر الراديو JRadioButton:

- تستخدم أداة زر الراديو JRadioButton عندما نريد أن نختار قيمة واحدة فقط من عدة اختيارات متاحة ، وبذلك فهي تتشابه مع أداة قائمة الاختيارات JComboBox ولكن الاختلاف يتمثل أساساً في الحجم الذي تحتاجه لكتابة الاختيارات في نافذة التطبيق ، فأداة قائمة الاختيارات JComboBox تحتاج حجم أقل من أداة زر الراديو JRadioButton لإظهار جميع الاختيارات المتاحة للمستخدم ، ولذلك لا تستخدم أداة زر الراديو JRadioButton إلا إذا كانت قائمة الاختيارات محدودة جداً وإلا احتجت إلي مساحة كبيرة جداً لإظهار جميع الاختيارات التي تريدها.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JRadioButton():

تقوم بإظهار أداة زر الراديو JRadioButton غير مختارة وبدون وجود أي نص أو صورة بجانبها.

دالة البناء JRadioButton(Icon icon):

تقوم بإظهار أداة زر الراديو JRadioButton غير مختارة وبدون وجود أي نص بجانبها ولكن مع وجود صورة بجانبها.

المعاملات Parameters:

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة زر الراديو JRadioButton.

دالة البناء JRadioButton(Icon icon, boolean selected):

تقوم بإظهار أداة زر الراديو JRadioButton ويكون اختيار الأداة محدد بالقيمة الموجودة في المعامل Parameter وبدون وجود أي نص بجانبها ولكن مع وجود صورة بجانبها.

المعاملات Parameters:

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة زر الراديو JRadioButton.

boolean selected: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون أداة زر الراديو JRadioButton مختارة ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فستكون أداة زر الراديو JRadioButton غير مختارة.

دالة البناء JRadioButton(String text):

تقوم بإظهار أداة زر الراديو JRadioButton غير مختارة وبوجود نص بجانبها ولكن بدون وجود صورة بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة زر الراديو JRadioButton.

دالة البناء JRadioButton(String text, boolean selected):

تقوم بإظهار أداة زر الراديو JRadioButton ويكون اختيار الأداة محدد بالقيمة الموجودة في المعامل Parameter وبدون وجود أي صورة بجانبها ولكن مع وجود نص بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة زر الراديو JRadioButton.

boolean selected: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون أداة زر الراديو JRadioButton مختارة ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فستكون أداة زر الراديو JRadioButton غير مختارة.

دالة البناء JRadioButton(String text, Icon icon):

تقوم بإظهار أداة زر الراديو JRadioButton غير مختارة وبوجود نص وصورة بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة زر الراديو JRadioButton.

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة زر الراديو JRadioButton.

دالة البناء JRadioButton(String text, Icon icon, boolean selected):

تقوم بإظهار أداة زر الراديو JRadioButton ويكون اختيار الأداة محدد بالقيمة الموجودة في المعامل Parameter وبوجود نص وصورة بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة زر الراديو JRadioButton.

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة زر الراديو JRadioButton.

boolean selected: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون أداة زر الراديو JRadioButton مختارة ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فستكون أداة زر الراديو JRadioButton غير مختارة.

المثال التالي يوضح استخدام أداة زر الراديو JRadioButton.

مثال (11): أداة زر الراديو JRadioButton:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة زر الراديو JRadioButton في البرنامج.

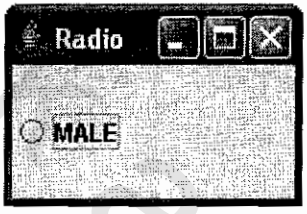
ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyRadioButton extends JFrame
5. {
6.   JRadioButton button;
7.
8.   MyRadioButton()
9.   {
10.       Container c = getContentPane();
11.       button = new JRadioButton("MALE");
12.       c.add(button);
13.
14.       setTitle("Radio");
15.       setSize(150,100);
16.       setVisible(true);
17.  }
18.
19. public static void main(String args[])
20. {
21.       MyRadioButton radio = new MyRadioButton();
22.  }
23.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyRadioButton التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JRadioButton.

- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم إنشاء هدف Object من نوع JRadioButton والذي يمثل أداة زر الراديو JRadioButton.
- في السطور من 19 إلي 22 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyRadioButton وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (12-15).



الشكل (12-15) زر الراديو JRadioButton

أداة صندوق التحقق JCheckBox:

- تمكننا أداة صندوق التحقق JCheckBox من اختيار قيمة من اثنين إما نعم true وإما لا false ، فمثلاً قد تريد من المستخدم بيان ما إذا كان عنده أطفال أم لا ، إذن هنا اختيار المستخدم إما أن عنده أطفال أو لا ، أو بمعني أخر: نعم أو لا ، وهنا تظهر فائدة أداة صندوق التحقق JCheckBox.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JCheckBox():

تقوم بإظهار أداة صندوق التحقق JCheckBox غير مختارة وبدون وجود أي نص أو صورة بجانبها.

دالة البناء JCheckBox(Icon icon):

تقوم بإظهار أداة صندوق التحقق JCheckBox غير مختارة وبدون وجود أي نص بجانبها ولكن مع وجود صورة بجانبها.

المعاملات Parameters:

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة صندوق التحقق JCheckBox.

دالة البناء JCheckBox(Icon icon, boolean selected):

تقوم بإظهار أداة صندوق التحقق JCheckBox ويكون اختيار الأداة محدد بالقيمة الموجودة في المعامل Parameter وبدون وجود أي نص بجانبها ولكن مع وجود صورة بجانبها.

المعاملات Parameters:

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة صندوق التحقق JCheckBox.

boolean selected: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون أداة صندوق التحقق JCheckBox مختارة ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فستكون أداة صندوق التحقق JCheckBox غير مختارة.

دالة البناء JCheckBox(String text):

تقوم بإظهار أداة صندوق التحقق JCheckBox غير مختارة وبوجود نص بجانبها ولكن بدون وجود صورة بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة صندوق التحقق JCheckBox.

دالة البناء JCheckBox(String text, boolean selected):

تقوم بإظهار أداة صندوق التحقق JCheckBox ويكون اختيار الأداة محدد بالقيمة الموجودة في المعامل Parameter وبدون وجود أي صورة بجانبها ولكن مع وجود نص بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة صندوق التحقق JCheckBox.

boolean selected: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون أداة صندوق التحقق JCheckBox مختارة ، أما إذا كانت قيمة هذا المعامل

Parameter تساوي false فستكون أداة صندوق التحقق JCheckBox غير مختارة.

دالة البناء JCheckBox(String text, Icon icon):

تقوم بإظهار أداة صندوق التحقق JCheckBox غير مختارة وبوجود نص وصورة بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة صندوق التحقق JCheckBox.

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة صندوق التحقق JCheckBox.

دالة البناء JCheckBox(String text, Icon icon, boolean selected):

تقوم بإظهار أداة صندوق التحقق JCheckBox ويكون اختيار الأداة محدد بالقيمة الموجودة في المعامل Parameter وبوجود نص وصورة بجانبها.

المعاملات Parameters:

String text: النص الذي يظهر بجانب أداة صندوق التحقق JCheckBox.

Icon icon: عبارة عن الصورة التي تظهر بجانب أداة صندوق التحقق JCheckBox.

boolean selected: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون أداة صندوق التحقق JCheckBox مختارة ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فستكون أداة صندوق التحقق JCheckBox غير مختارة.

المثال التالي يوضح استخدام أداة صندوق التحقق JCheckBox.

مثال (12): أداة صندوق التحقق JCheckBox:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة صندوق التحقق JCheckBox في البرنامج.

ثانياً: كود البرمجة:

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyCheckBox extends JFrame
```

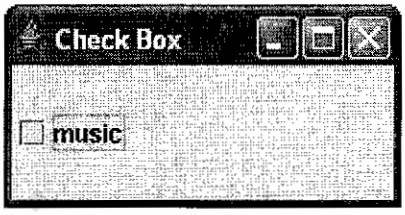
```
5.  {
6.    JCheckBox box;
7.
8.    MyCheckBox()
9.    {
10.       Container c = getContentPane();
11.       box = new JCheckBox("music");
12.       c.add(box);
13.
14.       setTitle("Check Box");
15.       setSize(200,100);
16.       setVisible(true);
17.   }
18.
19.   public static void main(String args[])
20.   {
21.       MyCheckBox check = new MyCheckBox();
22.   }
23.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyCheckBox التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JCheckBox.
- في السطر رقم 8 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 11 يتم إنشاء هدف Object من نوع JCheckBox والذي يمثل أداة صندوق التحقق JCheckBox.
- في السطور من 19 إلي 22 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند

تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyCheckBox وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.

- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (13-15).



الشكل (15-13) أداة صندوق التحقق JCheckBox

أداة صناديق الرسائل JOptionPane:

- أداة صناديق الرسائل JOptionPane تقوم بإظهار مربع حوارى Dialog Box تظهر فيه رسالة نحدد نصها.

تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة showOptionDialog(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon, Object[] options, Object initialValue):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: يمثل عنوان المربع الحوارى Dialog Box.

int optionType: يحدد الاختيارات التي ستظهر في المربع الحوارى Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.YES_NO_CANCEL_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به 3 أزرار تمثل yes وno وcancel.
- JOptionPane.YES_NO_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به زرين يمثلان yes وno.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.
- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.
- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

Icon icon: يحدد الصورة التي ستظهر علي المربع الحواري Dialog Box.

Object[] options: عبارة عن مصفوفة Array تحتوى على النص الذى سيظهر على الأزرار.

Object initialValue: يحدد الزر الافتراضي ، بمعنى أنه عند الضغط على زر Enter مباشرة فذلك سيمثل الضغط على هذا الزر الافتراضي.

الدالة showMessageDialog(Component parentComponent, Object message):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

الدالة showMessageDialog(Component parentComponent, Object message, String title, int messageType):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.

- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.
- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

الدالة showMessageDialog(Component parentComponent, Object message, String title, int messageType, Icon icon):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.
- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.

- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

Icon icon: يحدد الصورة التي ستظهر علي المربع الحواري Dialog Box.

الدالة showInputDialog(Component parentComponent, Object message):

تستخدم في إظهار مربع حواري Dialog Box يظهر رسالة للمستخدم ويطلب منه معلومة مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

الدالة showInputDialog(Component parentComponent, Object message, Object initialSelectionValue):

تستخدم في إظهار مربع حواري Dialog Box يظهر رسالة للمستخدم ويطلب منه معلومة مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

Object initialSelectionValue: يمثل القيمة الافتراضية لإدخال المستخدم.

الدالة showInputDialog(Component parentComponent, Object message, String title, int messageType):

تستخدم في إظهار مربع حواري Dialog Box يظهر رسالة للمستخدم ويطلب منه معلومة مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box.

يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.
- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.
- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

الدالة showInputDialog(Object message):

تستخدم في إظهار مربع حواري Dialog Box يظهر رسالة للمستخدم ويطلب منه معلومة مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

الدالة showInputDialog(Object message, Object initialSelectionValue):

تستخدم في إظهار مربع حواري Dialog Box يظهر رسالة للمستخدم ويطلب منه معلومة مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

Object initialSelectionValue: يمثل القيمة الافتراضية لإدخال المستخدم.

الدالة showInputDialog(Component parentComponent, Object message, String title, int messageType, Icon icon, Object[] selectionValues, Object initialSelectionValue):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.

- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.
- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

Icon icon: يحدد الصورة التي ستظهر علي المربع الحواري Dialog Box.

Object[] selectionValues: عبارة عن مصفوفة Array تحتوى علي جميع الأدوات التي تريدها أن تطلب معلومة من المستخدم. عادة يتم استخدام الأدوات التالية: أداة قائمة الاختيارات JComboBox ، أداة السرد JList ، أداة النص JTextField.

Object initialSelectionValue: يمثل القيمة الافتراضية لإدخال المستخدم.

الدالة showConfirmDialog(Component parentComponent, Object message):

تستخدم في إظهار مربع حواري Dialog Box يحتوي علي الأزرار Yes و No و Cancel وعنوانه Select an Option مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

الدالة showConfirmDialog(Component parentComponent, Object message, String title, int optionType):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int optionType: يحدد الاختيارات التي ستظهر في المربع الحوارى Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.YES_NO_CANCEL_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به 3 أزرار تمثل yes وno وcancel.
- JOptionPane.YES_NO_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به زرين يمثلان yes وno.

الدالة showConfirmDialog(Component parentComponent, Object message, String title, int optionType, int messageType):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int optionType: يحدد الاختيارات التي ستظهر في المربع الحوارى Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.YES_NO_CANCEL_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به 3 أزرار تمثل yes وno وcancel.

- JOptionPane.YES_NO_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به زرين يمثلان yes وno.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.
- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.
- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

الدالة showConfirmDialog(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon):

تستخدم في إظهار مربع حواري Dialog Box مع تحديد خصائص المربع الحواري Dialog Box من خلال المعاملات Parameters.

المعاملات Parameters:

Component parentComponent: يحدد النافذة الأم التي سيظهر المربع الحواري Dialog Box فيها.

Object message: يمثل الرسالة التي تظهر في المربع الحواري Dialog Box.

String title: عنوان المربع الحوارى Dialog Box.

int optionType: يحدد الاختيارات التي ستظهر في المربع الحوارى Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.YES_NO_CANCEL_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به 3 أزرار تمثل yes وno وcancel.
- JOptionPane.YES_NO_OPTION: يستخدم لإظهار مربع حوارى Dialog Box به زرين يمثلان yes وno.

int messageType: يبين نوع الرسالة التي تظهر في المربع الحواري Dialog Box. يمكن أن يكون هذا المتغير واحداً من القيم التالية:

- JOptionPane.ERROR_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة خطأ.
- JOptionPane.INFORMATION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة معلومة.
- JOptionPane.WARNING_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة تحذير.
- JOptionPane.QUESTION_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة سؤال.
- JOptionPane.PLAIN_MESSAGE: يستخدم لتوضيح أن الرسالة التي تظهر في المربع الحواري Dialog Box هي رسالة عادية بدون ظهور أي صورة بجانب الرسالة.

Icon icon: يحدد الصورة التي ستظهر علي المربع الحواري Dialog Box.

- المثال التالي يوضح استخدام أداة صناديق الرسائل JOptionPane.

مثال (13): أداة صناديق الرسائل JOptionPane:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة صناديق الرسائل JOptionPane في البرنامج.

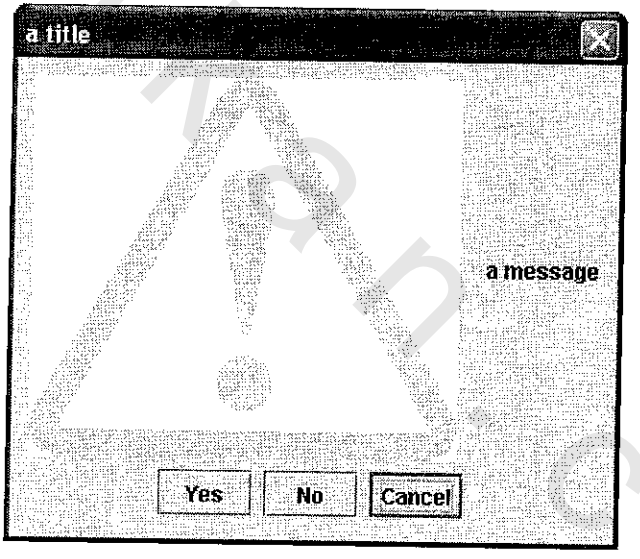
ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyOptionPane extends JFrame
5.  {
6.   MyOptionPane()
7.   {
8.       ImageIcon icon = new ImageIcon("1.gif");
9.       Object options[] = {"Yes","No","Cancel"};
10.      JOptionPane.showOptionDialog(this,
11.      "a message",
12.      "a title",
13.      JOptionPane.YES_NO_CANCEL_OPTION,
14.      JOptionPane.QUESTION_MESSAGE,
15.      icon,
16.      options,
17.      options[2]);
18. }
19.
20. public static void main(String args[])
21. {
22.      MyOptionPane option = new MyOptionPane();
23. }
24.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyOptionPane التي ترث من الفصيلة JFrame حتى يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء دالة البناء Constructor.

- في السطر رقم 8 يتم إنشاء هدف Object من نوع ImageIcon والذي يمثل الصورة التي ستظهر علي أداة صناديق الرسائل JOptionPane.
- في السطور من 9 إلي 17 يتم أظهار أداة صناديق الرسائل JOptionPane كما سبق لنا شرحه.
- في السطور من 20 إلي 23 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyOptionPane وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-15).



الشكل (15-14) أداة صناديق الرسائل JOptionPane

أداة اللوحة المبوبة JTabbedPane:

- تستطيع عن طريق أداة اللوحة المبوبة JTabbedPane إظهار مجموعة كبيرة جداً من البيانات والأدوات فى مساحة صغيرة عن طريق التبويبات Tabs بحيث أن كل تبويب Tab يعتبر نافذة Frame مستقلة بذاتها ، فمثلاً قد تحتاج من المستخدم إدخال مجوعتين من البيانات بحيث أن مجموعة البيانات الأولي تطلب بيانات شخصية مثل

الاسم والسن والعنوان وهكذا ، أما المجموعة الأخري فتطلب بيانات العمل مثل الوظيفة ومدة التعيين وهكذا ، إذن لكل مجموعة بيانات سنضع لها تبويب Tab خاص بها.

تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة addTab(String title, Icon icon, Component component, String tip):

تستخدم في إضافة تبويب Tab إلي التبويبات Tabs التي تظهر في أداة اللوحة المبوبة JTabbedPane.

المعاملات Parameters:

String title: يمثل عنوان التبويب Tab.

Icon icon: تمثل الصورة التي ستظهر علي التبويب Tab.

Component component: يمثل الأداة التي ستظهر عند الضغط علي التبويب Tab.

String tip: يمثل التلميح Tool Tip الذي سيظهر علي التبويب Tab عند الوقوف بمؤشر الفارة Mouse لمدة ثواني عليه.

الدالة addTab(String title, Icon icon, Component component):

تستخدم في إضافة تبويب Tab إلي التبويبات Tabs التي تظهر في أداة اللوحة المبوبة JTabbedPane.

المعاملات Parameters:

String title: يمثل عنوان التبويب Tab.

Icon icon: تمثل الصورة التي ستظهر علي التبويب Tab.

Component component: يمثل الأداة التي ستظهر عند الضغط علي التبويب Tab.

الدالة addTab(String title, Component component):

تستخدم في إضافة تبويب Tab إلي التبويبات Tabs التي تظهر في أداة اللوحة المبوبة JTabbedPane.

المعاملات Parameters:

String title: يمثل عنوان التبويب Tab.

Component component: يمثل الأداة التي ستظهر عند الضغط علي التبويب Tab.

المثال التالي يوضح استخدام أداة اللوحة المبوبة JTabbedPane.

مثال (14): أداة اللوحة المبوبة JTabbedPane:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة اللوحة المبوبة JTabbedPane في البرنامج.

ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyTab extends JFrame
5.  {
6.    JButton submit, cancel;
7.    JPanel p1, p2;
8.    JTabbedPane tab;
9.
10. MyTab()
11. {
12.       Container c = getContentPane();
13.
14.       submit = new JButton("Submit");
15.       cancel = new JButton("Cancel");
16.
17.       p1 = new JPanel();
18.       p2 = new JPanel();
19.
20.       tab = new JTabbedPane();
```

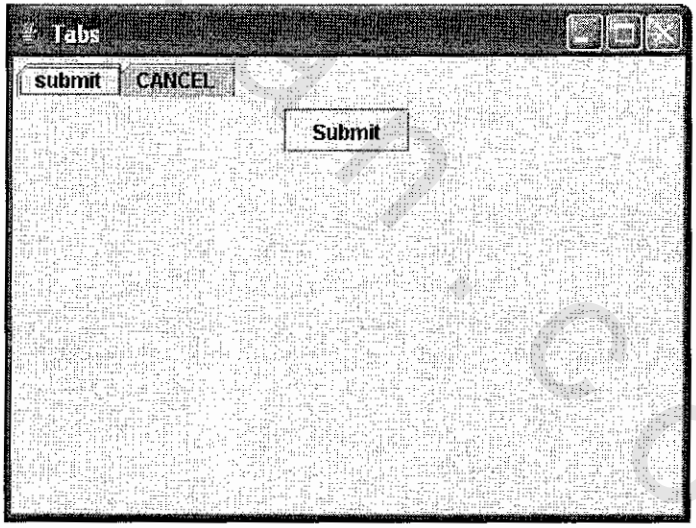
```
21.
22.       createPanel1();
23.       createPanel2();
24.
25.       c.add(tab);
26.
27.       tab.addTab("submit",null,p1,"this is panel1");
28.       tab.addTab("Cancel",null,p2,"this is panel2");
29.
30.       setTitle("Tabs");
31.       setSize(400,300);
32.       setVisible(true);
33.  }
34.
35.  void createPanel1()
36.  {
37.       p1.add(submit);
38.  }
39.
40.  void createPanel2()
41.  {
42.       p2.add(cancel);
43.  }
44.
45.  public static void main(String args[])
46.  {
47.       MyTab tab=new MyTab();
48.  }
49.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyTab التي ترث من الفصيلة JFrame حتى يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغيرين من نوع JButton.
- في السطر رقم 7 يتم إنشاء متغيرين من نوع JPanel.
- في السطر رقم 8 يتم إنشاء متغير من نوع JTabbedPane.
- في السطر رقم 10 يتم إنشاء دالة البناء Constructor.
- في السطرين 14 و 15 يتم إنشاء هدف Object من نوع JButton والذي يمثل أداة الزر JButton.
- في السطرين 17 و 18 يتم إنشاء هدف Object من نوع JPanel والذي يمثل أداة اللوحة JPanel.
- في السطر رقم 20 يتم إنشاء هدف Object من نوع JTabbedPane والذي يمثل أداة اللوحة المبوبة JTabbedPane.
- في السطر رقم 22 نقوم باستدعاء الدالة createPanel1() التي عرفناها في السطر رقم 35 والتي تقوم بوضع الأدوات Controls فى التبويب Tab الأول وهذا ما سنشرحه لاحقاً.
- في السطر رقم 23 نقوم باستدعاء الدالة createPanel2() التي عرفناها في السطر رقم 40 والتي تقوم بوضع الأدوات Controls فى التبويب Tab الثاني وهذا ما سنشرحه لاحقاً.
- في السطر رقم 27 يتم إضافة تبويب Tab للهدف Object المسمي tab والذي يمثل أداة لوح التبويب JTabbedPane عن طريق الدالة addTab() كما سبق لنا شرحه.
- في السطر رقم 28 يتم إضافة تبويب Tab ثاني للهدف Object المسمي tab والذي يمثل أداة لوح التبويب JTabbedPane عن طريق الدالة addTab() كما سبق لنا شرحه.
- في السطر رقم 35 يتم تعريف الدالة createPanel1() التى تضيف الزر Submit

لأداة اللوحة JPanel المسماة p1 كما هو واضح فى السطر رقم 37.

- في السطر رقم 40 يتم تعريف الدالة ()createPanel2 التى تضيف الزر Cancel لأداة اللوحة JPanel المسماة p2 كما هو واضح فى السطر رقم 42.
- في السطور من 45 إلي 48 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyTab وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-15) حيث يتم ظهور التبويب Tab الأول تلقائياً عند تنفيذ البرنامج ويمكنك تجربة الضغط التبويب Tab الثاني لإظهار محتوياته ، كما يمكنك إعادة الضغط علي التبويب Tab الأول لإظهار محتوياته مرة اخري. تأكد من ظهور التلميحات Tool Tips عند الوقوف بمؤشر الفارة Mouse لمدة ثواني علي أي من التبويبين Tabs.



الشكل (15-15) أداة اللوحة المبوبة JTabbedPane

أداة القائمة JMenu:

- بإمكاننا إنشاء قوائم اختيار Menus فى لغة Java عن طريق إنشاء شريط قوائم JMenuBar لكى يحتوى على القوائم Menus المطلوبة بحيث أن كل قائمة رئيسية

يتم إنشاءها عن طريق الفصيلة JMenu ، أما كل قائمة فرعية من القائمة الرئيسية فننشئها باستخدام الفصيلة JMenuItem.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JMenu():

تقوم بإظهار قائمة JMenu بلا أي نص عليها.

دالة البناء JMenu(String s):

تقوم بإظهار قائمة JMenu مع ظهور نص عليها.

المعاملات Parameters:

String s: يمثل النص الذي يظهر في القائمة JMenu.

خطوات إنشاء القوائم Menus:

- نقوم بإنشاء هدف Object من نوع JMenuBar وليكن اسمه bar.
- نقوم باستخدام الدالة setJMenuBar(bar) لتحديد أن bar هو شريط القوائم Menu Bar في شاشة البرنامج.
- لكل قائمة رئيسية (أي التي تظهر أسماؤها مباشرة بدون الضغط عليها) نقوم بإنشاء هدف Object من نوع JMenu.
- نقوم باستخدام الدالة ( )add المتاحة من الهدف bar (راجع الخطوة الأولي) لإضافة القوائم Menus الرئيسية إلي شريط القوائم Menus.
- لكل قائمة فرعية (أي التي تظهر أسماؤها بعد الضغط علي قائمتها الرئيسية) نقوم بإنشاء هدف Object من نوع JMenuItem.
- نقوم بإضافة كل هدف Object من نوع JMenuItem إلي القائمة Menu الرئيسية التي تحتويه عن طريق الدالة ( )add.
- كخطوة اختيارية ، يمكنك استخدام الدالة ( )addSeparator لإضافة فاصل أفقي بعد القائمة Menu.

المثال التالي يوضح استخدام أداة القوائم JMenu.

مثال (15): أداة القائمة JMenu:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة القائمة JMenu في البرنامج.

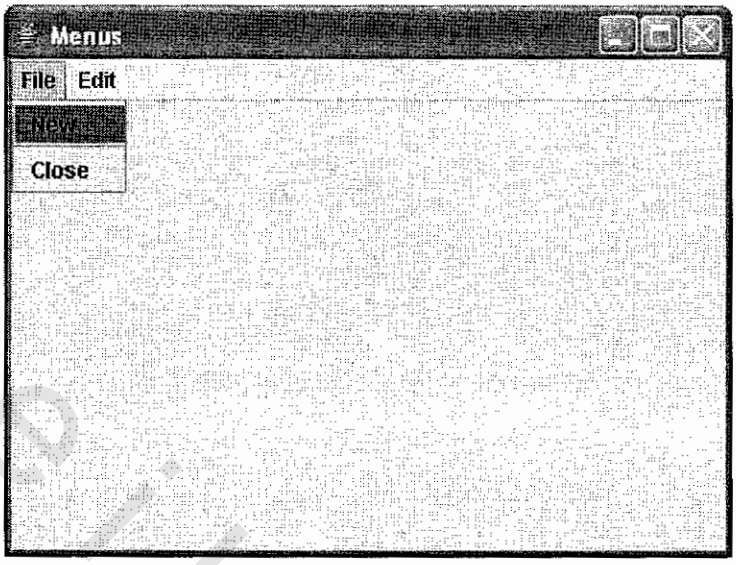
ثانياً: كود البرمجة:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyMenu extends JFrame
5.  {
6.    JMenuBar bar;
7.    JMenu file, edit;
8.    JMenuItem mnew, close, copy, paste;
9.
10.  MyMenu()
11.  {
12.       bar = new JMenuBar();
13.       setJMenuBar(bar);
14.
15.       file = new JMenu("File");
16.       edit = new JMenu("Edit");
17.
18.       bar.add(file);
19.       bar.add(edit);
20.
21.       mnew = new JMenuItem("New");
22.       mnew.setBackground(Color.red);
23.       close = new JMenuItem("Close");
24.       copy = new JMenuItem("Copy");
25.       paste = new JMenuItem("Paste");
```

```
26.
27.         file.add(mnew);
28.         file.addSeparator();
29.         file.add(close);
30.         edit.add(copy);
31.         edit.addSeparator();
32.         edit.add(paste);
33.
34.         setTitle("Menus");
35.         setSize(400,300);
36.         setVisible(true);
37.   }
38.
39.   public static void main(String args[])
40.   {
41.         MyMenu menus = new MyMenu();
42.   }
43.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyMenu التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطور من 6 إلي 32 يتم إنشاء القوائم Menus كما سبق لنا شرح خطواته.
- 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 MyMenu وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (16-15).



الشكل (15-16) أداة القائمة JMenu

أداة اختيار ملف JFileChooser:

- أحياناً نحتاج أن نظهر نافذة صغيرة لتسمح لنا باختيار ملف لفتحه أو حفظه. تمكننا أداة اختيار ملف JFileChooser من تحديد اسم ملف ليتم فتحه أو حفظه ، ولكن لاحظ أنك كمبرمج تكتب الكود الخاص بحفظ أو فتح الملف ، فذلك ليس وظيفة أداة اختيار الملف JFileChooser بل وظيفتك أنت.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JFileChooser():

تقوم بإظهار أداة اختيار الملف JFileChooser بحيث يكون المسار Directory الافتراضي عند فتح الأداة هو المسار الافتراضي للمستخدم والذي عادة ما يكون المجلد My Documents.

دالة البناء JFileChooser(String currentDirectoryPath):

تقوم بإظهار أداة اختيار الملف JFileChooser بحيث يكون المسار Directory الافتراضي عند فتح الأداة هو المسار الموجود في المعامل Parameter.

المعاملات Parameters:

String currentDirectoryPath: يمثل المسار Directory الافتراضي عند فتح الأداة.

كما تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة showOpenDialog(Component parent):

تستخدم في إظهار أداة اختيار ملف JFileChooser في وضع فتح ملف.

المعاملات Parameters:

Component parent: يحدد النافذة الأم التي ستظهر أداة اختيار ملف JFileChooser فيها.

الدالة showSaveDialog(Component parent):

تستخدم في إظهار أداة اختيار ملف JFileChooser في وضع حفظ ملف.

المعاملات Parameters:

Component parent: يحدد النافذة الأم التي ستظهر أداة اختيار ملف JFileChooser فيها.

المثال التالي يوضح استخدام أداة اختيار ملف JFileChooser.

مثال (16): أداة اختيار ملف JFileChooser:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة اختيار ملف JFileChooser في البرنامج.

ثانياً: كود البرمجة:

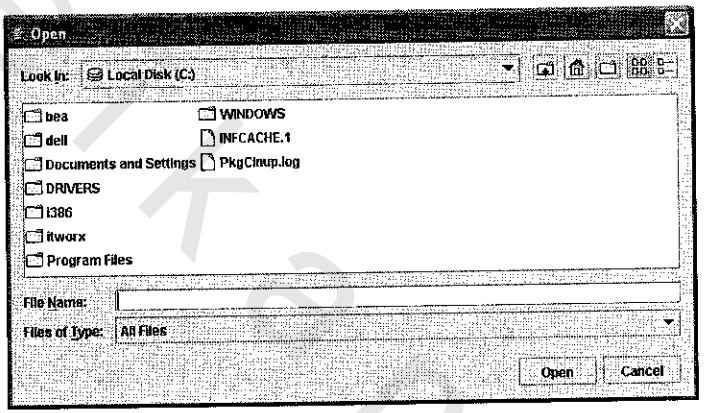
```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class MyFileChooser extends JFrame
```

```
5.  {
6.    JFileChooser choose;
7.    String name;
8.    String path;
9.
10.   MyFileChooser()
11.   {
12.       Container c = getContentPane();
13.
14.       choose = new JFileChooser("c:\\") ;
15.
16.       int i = choose.showOpenDialog(this);
17.
18.       if ( i == JFileChooser.APPROVE_OPTION )
19.       {
20.           name = choose.getSelectedFile().getName();
21.           path =
    choose.getSelectedFile().getAbsolutePath();
22.           System.out.println("File Name: " + name);
23.           System.out.println("File Path: " + path);
24.       }
25.
26.       else
27.       {
28.           System.out.println("You Pressed Cancel");
29.       }
30.   }
31.
32.   public static void main (String args[])
33.   {
34.       MyFileChooser file = new MyFileChooser();
35.   }
36.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyFileChooser التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع JFileChooser.
- في السطر رقم 7 يتم إنشاء متغير من نوع String يمثل اسم الملف.
- في السطر رقم 8 يتم إنشاء متغير من نوع String يمثل مسار الملف.
- في السطر رقم 10 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 14 يتم إنشاء هدف Object من نوع JFileChooser والذي يمثل أداة فتح ملف JFileChooser بحيث يكون المسار الابتدائى للملف الذي سيتم اختياره هو المسار :C مع ملاحظة أنه عند تحديد المسار نستخدم \\ وليس \ واحدة حيث أن \ لها معنى خاص فى لغة Java.
- في السطر رقم 16 يتم إظهار مربع اختيار ملف JFileChooser لفتح الملف الذي تختاره. لاحظ أن كلمة this ضرورية فى السطر رقم 16 بسبب أن أى أداة فتح ملف JFileChooser تكون مقترنة بنافذة أم تظهر فيها ، وفى هذه الحالة فإن this تشير إلى هدف Object من نوع الفصيلة JFrame لأن الفصيلة class المسماة MyFileChooser - والتي تمثل الفصيلة class الحالية - ترث من الفصيلة JFrame.
- في السطر رقم 18 يتم التحقق من القيمة المرتجعة Return Value من إظهار أداة فتح ملف JFileChooser ، فإذا كنت ضغطت على الزر Open (والمحدد بالقيمة JFileChooser.APPROVE_OPTION) من أداة اختيار ملف JFileChooser ، فتقوم الدالة getSelectedFile() فى السطر رقم 20 بمعرفة الملف الذى اخترته وتقوم الدالة getName() بمعرفة اسم الملف المختار ، أما الدالة getAbsolutePath() فتقوم بمعرفة مسار الملف المختار كما هو واضح فى السطر رقم 21.
- في السطرين 22 و 23 يتم طباعة اسم الملف المختار ومساره.

- في السطر رقم 28 يتم طباعة رسالة على شاشة الدوس Dos إذا كنت ضغطت على الزر Cancel من أداة اختيار ملف JFileChooser.
- في السطور من 32 إلي 35 يتم إنشاء الدالة الرئيسية ()main التي يتم تنفيذها عند تنفيذ البرنامج وفيها يتم إنشاء هدف Object من نوع الفصيلة MyFileChooser وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-17).



الشكل (15-17) أداة اختيار ملف JFileChooser

يمكنك تجربة اختيار ملف مرة والضغط علي الزر Cancel مرة أخري وتأكد من طباعة الرسائل علي شاشة الدوس Dos.

أداة الزر JButton:

- تستخدم أداة الزر JButton لإظهار أزرار فى نافذة التطبيق حيث يستخدم الزر في تنفيذ أوامر معينة عندما يضغط المستخدم علي الزر JButton.

تتوافر العديد من دوال البناء Constructors لهذه الأداة كما يلي.

دالة البناء JButton():

تقوم بإظهار أداة زر JButton بلا أي نص عليه وبدون ظهور صورة عليه.

دالة البناء JButton(Icon icon):

تقوم بإظهار أداة زر JButton بلا أي نص عليه ولكن تظهر عليه صورة.

المعاملات Parameters:

Icon icon: تمثل الصورة التي ستظهر علي أداة الزر JButton.

دالة البناء JButton(String text):

تقوم بإظهار أداة زر JButton ويظهر عليه نص.

المعاملات Parameters:

String text: تمثل النص الذي سيظهر علي أداة الزر JButton.

دالة البناء JButton(String text, Icon icon):

تقوم بإظهار أداة زر JButton ويظهر عليه نص وصورة.

المعاملات Parameters:

String text: تمثل النص الذي سيظهر علي أداة الزر JButton.

Icon icon: تمثل الصورة التي ستظهر علي أداة الزر JButton.

كما تتوافر العديد من الدوال Methods التي تستخدم في تحديد الخصائص لهذه الأداة كما سيلي شرحهم في النقاط التالية.

الدالة setText(String text):

تستخدم في تغيير النص الذي يظهر علي أداة الزر JButton.

المعاملات Parameters:

String text: يمثل النص الذي يظهر علي أداة الزر JButton.

الدالة setIcon(Icon icon):

تستخدم في تغيير شكل الأيقونة Icon التي تظهر علي أداة الزر JButton.

المعاملات Parameters:

Icon icon: يمثل الصورة التي ستظهر علي أداة الزر JButton.

لاستخدام هذه الدالة Method فإننا لا بد أن نقوم بإنشاء الصورة التي ستظهر كأيقونة Icon للإطار JFrame كالتالي:

```
Icon icon = new ImageIcon("1.gif");
```

حيث يتم إنشاء الصورة عن طريق وضع صورة باسم gif.1 في نفس مسار ملف كود البرمجة ويتم تحميلها في البرنامج باستخدام السطر السابق.

الدالة setBackground(Color bg):

تستخدم لتغيير لون خلفية أداة الزر JButton.

المعاملات Parameters:

Color bg: لون الخلفية المطلوب.

تحتوي لغة Java علي عدد كبير من الألوان الجاهزة كما يتضح من الجدول التالي:

اسم اللون Color	
BLACK	DARK_GRAY
CYAN	BLUE
GREEN	GRAY
MAGENTA	LIGHT_GRAY
PINK	ORANGE
WHITE	RED
	YELLOW

الدالة setForeground(Color fg):

تستخدم لتغيير لون النص الذي يظهر علي أداة الزر JButton.

المعاملات Parameters:

Color fg: لون النص المطلوب.

الدالة setFont(Font font):

تستخدم في تغيير شكل الخط للنص الذي يظهر علي أداة الزر JButton.

المعاملات Parameters:

Font font: شكل الخط المطلوب.

يمكننا إنشاء خط عن طريق الفصيلة Font التي تستقبل ثلاثة معاملات Parameters كالتالي:

المعامل الأول: يمثل شكل الخط.

المعامل الثاني: يمثل نمط الخط Style كأن يكون عريض Bold أو مائل Italic أو عادي Plain.

المعامل الثالث: يمثل حجم الخط المطلوب.

أي يمكننا إنشاء خط كالتالي:

```
Font font = new Font("helvetica",Font.BOLD,16);
```

حيث سيكون هذا الخط بالخصائص التالية:

شكل الخط: helvetica.

نمط الخط Style: عريض Bold.

حجم الخط: 16.

الدالة setToolTipText(String text):

تستخدم في تحديد تلميح Tool Tip للأداة عندما يقف المستخدم بمؤشر الفارة Mouse علي الأداة لمدة ثواني.

المعاملات Parameters:

String text: يمثل النص الذي سيظهر كتلميح Tool Tip.

الدالة setEnabled(boolean b):

تستخدم في تحديد ما إذا كانت الأداة فعالة Enabled أم لا.

المعاملات Parameters:

boolean b: إذا كانت قيمة هذا المعامل Parameter تساوي true فستكون الأداة

فعالة Enabled ، أما إذا كانت قيمة هذا المعامل Parameter تساوي false فستكون الأداة غير فعالة Disabled.

الدالة setMnemonic(char mnemonic):

تستخدم في تحديد حرف من لوحة المفاتيح Keyboard بحيث إذا ضغطنا (الحرف + Alt) فذلك يقوم بنفس وظيفة الضغط على الزر.

المعاملات Parameters:

char mnemonic: الحرف الذي إذا ضغطنا عليه مع الزر Alt (الحرف + Alt) فذلك يقوم بنفس وظيفة الضغط على الزر.

المثال التالي يوضح استخدام أداة الزر JButton.

مثال (17): أداة الزر JButton:

أولاً: هدف المثال:

نريد أن نبين كيفية إظهار أداة الزر JButton مع توضيح كيفية تحديد أهم خصائصها.

ثانياً: كود البرمجة:

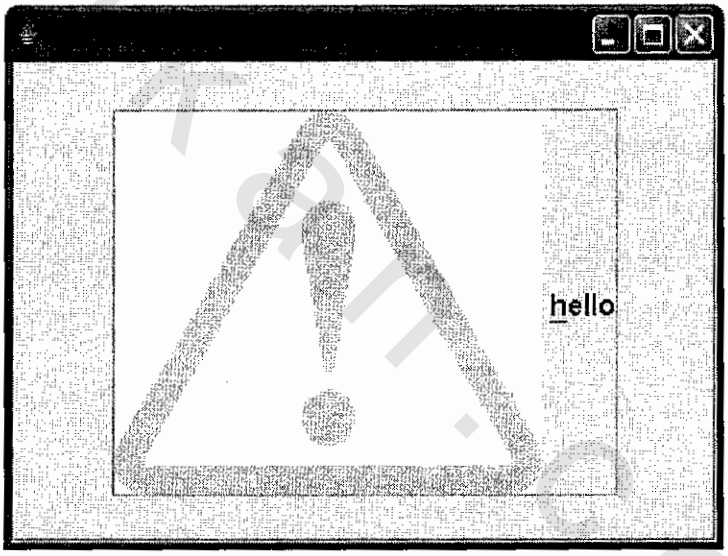
```
1.  import javax.swing.*;
2.  import java.awt.*;
3.
4.  public class MyButton extends JFrame
5.  {
6.    ImageIcon icon;
7.    JButton button;
8.
9.    MyButton()
10.   {
11.       Container c = getContentPane();
12.
13.       icon = new ImageIcon("1.gif");
```

```
14.
15.       button = new JButton ("hello",icon);
16.       button.setForeground(Color.red);
17.       button.setBackground(Color.yellow);
18.       button.setFont(new
    Font("helvetica",Font.BOLD,16));
19.       button.setToolTipText("this is a button");
20.       button.setEnabled(true);
21.       button.setMnemonic('H');
22.       c.add(button);
23.
24.       setSize(400,300);
25.       setVisible(true);
26.  }
27.
28.  public static void main (String args[])
29.  {
30.       MyButton button = new MyButton();
31.  }
32.}
```

ثالثاً: شرح الكود:

- في السطرين 1 و 2 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 4 يتم إنشاء الفصيلة MyButton التي ترث من الفصيلة JFrame حتي يمكنها إنشاء شاشة البرنامج كما شرحنا سابقاً.
- في السطر رقم 6 يتم إنشاء متغير من نوع ImageIcon.
- في السطر رقم 7 يتم إنشاء متغير من نوع JButton.
- في السطر رقم 9 يتم إنشاء دالة البناء Constructor.
- في السطر رقم 13 يتم إنشاء هدف Object من نوع ImageIcon والذي يمثل الصورة علي أداة الزر JButton.

- في السطر رقم 15 يتم إنشاء هدف Object من نوع JButton والذي يمثل أداة الزر JButton.
- في السطور من 16 إلي 21 يتم تحديد خصائص أداة الزر JButton كما سبق لنا شرحه.
- 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 MyButton وينتج عن ذلك استدعاء دالة البناء Constructor وإظهار نافذة البرنامج.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (15-18).



الشكل (15-18) أداة الزر JButton

ملحوظات:

- عند استخدام الدالة ( )setMnemonic ، فإذا كان الحرف المختار ليس من حروف الكلمة التى على الزر مثل:

```
JButton b1 = new JButton ("Button 1");
b1.setMnemonic ('y');
```

فهنا الحرف y ليس من حروف الكلمة Button 1 ، فما سيحدث أن الضغط على الزرين Alt+y يكافئ الضغط على الزر ولكن لن يكون هناك خط تحت أى حرف من حروف الكلمة Button 1 وذلك بسبب عدم وجود الحرف y في النص ، وذلك على عكس اختيار الحرف b ، ففى هذه الحالة ستلاحظ وجود خط تحت الحرف B.

- الدالة ()setMnemonic تستقبل معامل من نوع حرف char أى أنه من الخطأ كتابة الدالة كالأتى:

```
b1.setMnemonic("b");
```

حيث أنه فى هذه الحالة يعامل المعامل b كنص String ولذلك فإنه عند ترجمة compile البرنامج تظهر رسالة خطأ.

ملخص الفصل:

- تعلمنا في هذا الفصل كيفية استخدام العديد من أدوات مكتبة Swing وكيفية تحديد خصائص كل أداة علي حدة.
- في الفصل القادم نتعلم – بإذن الله – كيفية تصميم الشاشة بحيث تحتوي علي أكثر من أداة ، فتابع معنا الفصل القادم.