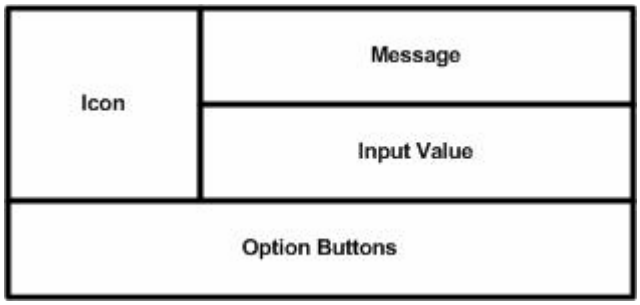
خلف كواليس JOptionPane:

من ضمن أدوات Swing المتوفرة بـ J2SE كلاس مميز لانشاء Panel لتكون على شكل نافذة منبثقة (Popup window). تستطيع استخدام النافذة لإظهار الرسائل للمستخدم و جمع المعلومات المطلوبه منه. الـ Panel ينقسم إلى أربع أقسام رئيسية و هي على الشكل التالي: قسم خاص للايقونة و قسم لاظهار الرسائل و قسم لجمع المعلومات و اخيراً قسم للأزرار. و تستطيع على الأقل استخدام قسم خاص بالرسالة و قسم الأزرار.



قسم الأيقونه خاص لإظهار الـ javax.swing.Icon. يمثل شكل الأيقونة محتوى الرسالة الظاهره للمستخدم و توجد هناك عدة أشكال جاهزه للايقونة من خلال الـ Java Look And Feel و تستطيع التعامل معها من خلال الخصائص التالية:

- `OptionPane.informationIcon`
- `OptionPane.warningIcon`
- `OptionPane.errorIcon`
- `OptionPane.questionIcon`

لا يجب على المستخدم تغيير الخصائص بشكل يدوي عن طريق الـ UIManager.put(property name, new resource). بل تستطيع عمل ذلك بتعريف الجزء المخصص للرسالة من خلال الـ JOptionPane:

- `JOptionPane.PLAIN_MESSAGE`
- `JOptionPane.INFORMATION_MESSAGE`
- `JOptionPane.WARNING_MESSAGE`
- `JOptionPane.ERROR_MESSAGE`
- `JOptionPane.QUESTION_MESSAGE`

الـ Plain عبارة عن مساحة فارغه أما البقية فتستخدم الايقونات الافتراضية للـ Java Look And Feel:

القسم الثاني من الـ JOptionPane هو قسم الرسائل التي تظهر للمستخدم. و تستطيع استعراض اي عدد من الـ Objects بالقسم و سوف يتم عرضه كرسالة للمستخدمين.
القسم التالي هو قسم ادخال المعلومات المطلوبة من المستخدم، و باستخدام هذا القسم تستطيع استخدام الـ JTextField أو JComboBox أو JList لعرض المعلومات للمستخدم و الاختيار من بينها.
القسم الأخير هو قسم الأزرار و هو قسم خاص لعرض الأزرار مثل Yes/No. و بالضغط على أحد الأزرار ينتهي عمل الـ JOptionPane و يوجد خيارات افتراضية لهذا القسم (مثل القسم الخاص بالأيقونات،كـ OptionPane.yesButtonText) تستطيع ايضاً انتاج أزرار اخرى لاستخدامها أو الغاء هذه الخاصية بالـ JOptionPane. و تستطيع كذلك استخدام لغتك الأم لتسمية الأزرار باسماء لغات اخرى مثل "موافق/غير موافق".

الخصائص المتوفرة بشكل افتراضي لهذا القسم كتالي:

- `DEFAULT_OPTION`
- `YES_NO_OPTION`
- `YES_NO_CANCEL_OPTION`
- `OK_CANCEL_OPTION`

الخاصية الافتراضية للـ JOptionPane هو الـ OK button.

استخدام JOptionPane:

الـ JOptionPane يحتوى على سبع من الـ constructors. كل واحد منها يرجع أداه لتستخدمها في كل مكان. أيضاً تستطيع إنشاء أدوات لتستخدمها بشكل افتراضي من خلال الـ JDialog و الـ JInternalFrame. توجد هناك ثمانية طرق لعرض الـ JOptionPane من خلال الـ JDialog أو الـ JInternalFrame:

- `show[Internal]MessageDialog`
- `show[Internal]ConfirmDialog`
- `show[Internal]InputDialog`
- `show[Internal]OptionDialog`

خيار الـ Internal لعرض الـ JOptionPane بداخل الـ JInternalFrame. أما الخيار الاخر فهو للـ JDialog.

الـ Message Dialog لعرض رسالة للمستخدم للموافقة عليها عبر الضغط على الزر – لا تستطيع اعادة اي قيمة –. تستطيع استخدام هذه الخاصية بالكود التالي:

```
JOptionPane.showMessageDialog(component, "Hello, World");
```

النظام سوف يقوم بعرض الـ MessageDialog بوسط الشاشة فوق الـ component.

توجد هناك طريقتين لعرض المعلومات على المستخدم و تستطيع أيضاً تغيير عنوان النافذة و نوع الرسالة و تخصيص شكل الأيقونة التي سوف تظهر.

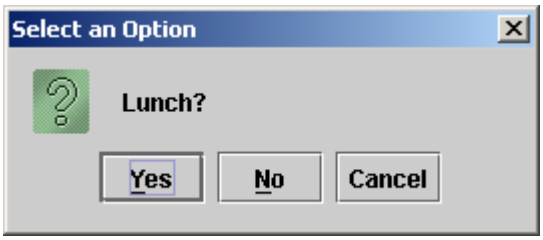
- ```
  showMessageDialog(Component parentComponent,
                    Object message,
                    String title,
                    int messageType)
  ```
- ```
  showMessageDialog(Component parentComponent,
                    Object message,
                    String title,
                    int messageType,
                    Icon icon)
  ```

و توجد هناك اربع طرق لعرض نوافذ الموافقة:

- `showConfirmDialog(Component parentComponent, Object message)`
- `showConfirmDialog(Component parentComponent,`
  `Object message,`
  `String title,`
  `int optionType)`
- `showConfirmDialog(Component parentComponent,`
  `Object message,`
  `String title,`
  `int optionType,`
  `int messageType)`
- `showConfirmDialog(Component parentComponent, Object message,`
  `String title, int optionType,`
  `int messageType, Icon icon)`

أسهل طريقة لعرض نافذة للسؤال (مع عدم تغيير الخصائص) هي:

```
JOptionPane.showConfirmDialog(component, "Lunch?");
```



اذا كانت الخصائص الافتراضية لا تعجبك فتستطيع تغييرها بكل سهولة مع الـ JOptionPane.

بعكس الـ MessageDialog، تستطيع من خلال الـ ConfirmDialog استرجاع القيم المطلوبة من المستخدم و تستطيع معرفة الزر الذي قام المستخدم بالضغط عليه. الـ ConfirmDialog يقوم باعادة قيم عددية (integer) عن طريق أحد الثوابت التالية من الـ JOptionPane:

- `CANCEL_OPTION`
- `CLOSED_OPTION`
- `NO_OPTION`
- `OK_OPTION`
- `YES_OPTION`

بالمثال التالي تستطيع معرفة الزر الذي قام المستخدم بالضغط عليه:

```
int returnValue = JOptionPane.showConfirmDialog(component, "Lunch?");
String message = null;
if(returnValue == JOptionPane.YES_OPTION) {
    message = "Yes";
}
else if(returnValue == JOptionPane.NO_OPTION) {
    message = "No";
}
else if(returnValue == JOptionPane.CANCEL_OPTION) {
    message = "Cancel";
}
else if(returnValue == JOptionPane.CLOSED_OPTION) {
    message = "Closed";
}
System.out.println("User selected: " + message);
```

و توجد هناك ست طرق لعرض نوافذ لادخال البيانات InputDialog (خمسة طرق لاعادة الـ String):

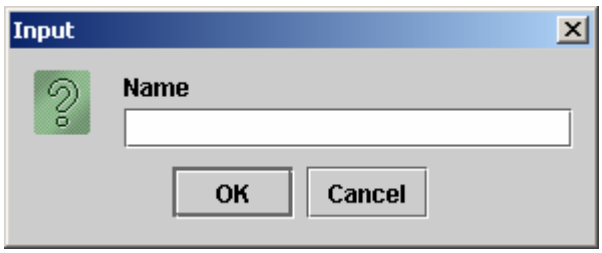
- `showInputDialog(Object message)`
- `showInputDialog(Object message, Object initialSelectionValue)`
- `showInputDialog(Component parentComponent, Object message)`
- `showInputDialog(Component parentComponent, Object message, Object initialSelectionValue)`
- `showInputDialog(Component parentComponent ,Object message, String title,int messageType)`

و اخيراً طريقة لاعادة قيم الـ Object:

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

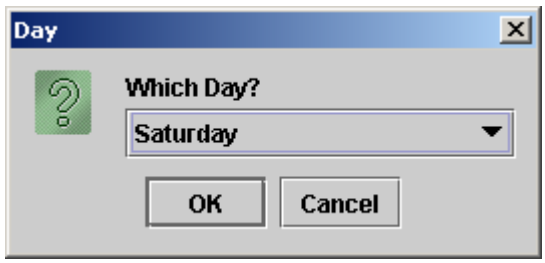
التعديل البسيط الذي حصل هو وجود مكان لادخال البيانات و أزرار للموافقة و الالغاء.

```
String value = JOptionPane.showInputDialog("Name");
```



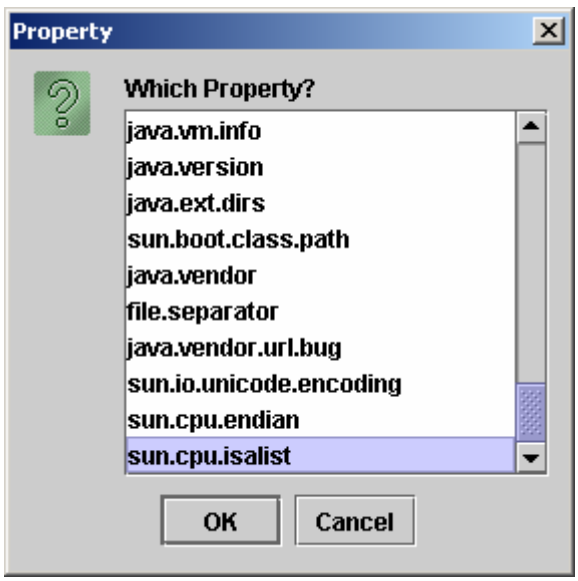
كما فعلنا سابقاً، تستطيع التعديل على الأيقونة و العنوان و محتوى الرسالة. أما الأزرار فهي ثابتة لا تتغير و تستطيع ايضاً اضافة نص جاهزاً بالـ text field.
بالنسبة للطريقة الأخيرة للـ InputDialog فهي مميزة عن الطريق الأخرى. و هذا عن طريق استخدام مصفوفة (Array). بالمثال التالي نستطيع إيضاح ذلك:

```
String smallList[] = {
    "Sunday",
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday"
};
String value =
    (String)JOptionPane.showInputDialog(
        component,
        "Which Day?",
        "Day",
        JOptionPane.QUESTION_MESSAGE,
        null, // Use default icon
        smallList,
        smallList[smallList.length-1]);
System.out.println("Day: " + value);
```
```



لاستخدام مجموعة (List) أكبر – الكود يشبه الكود السابق – ، هنا مجموعة (List) لخصائص النظام:

```
Object largeList[] = System.getProperties().keySet().toArray();
String value =
    (String)JOptionPane.showInputDialog(
        component,
        "Which Property?",
        "Property",
        JOptionPane.QUESTION_MESSAGE,
        null,
        largeList,
        largeList[largeList.length-1]);
System.out.println("Property: " + value);
```



أخيراً، النافذة التي تشمل الجميع، showOptionDialog و التي لها منهج واحد فقط (Method):

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

الأمر الوحيد الجديد هنا هو المصفوفة و تستطيع من خلالها التعديل على مجموعة الأزرار. لاحظ انها تحتوي على Object و ليس على String. اذا الـ Object عبارة عن Component فسوف يعرض بالنافذة ، و تستطيع من خلال ذلك عرض ايقونة بالأزرار.
  ```

الـ showOptionDialog يقوم باعادة قيم عدديه int.

```
String options[] = {"Yes", "Not Now", "Go Away"};
int value = JOptionPane.showOptionDialog(
    component,
    "Lunch?",
    "Lunch Time",
    JOptionPane.YES_NO_OPTION, // Need something
    JOptionPane.QUESTION_MESSAGE,
    null, // Use default icon for message type
    options,
    options[1]);
if (value == JOptionPane.CLOSED_OPTION) {
    System.out.println("Closed window");
}
else {
    System.out.println("Selected: " + options[value]);
}
```

إذا كنت لا تريد أزرار، فأضف مصفوفة (Array) فارغة.

استخدام التواء النصوص (Word warp):

أداة الـ JOptionPane لها خاصية القراءة فقط لـ (MaxCharactersPerLineCount) لحساب أكبر عدد من الحروف لكل سطر، العدد الافتراضي للخاصية هو Integer.MAX_VALUE. نستطيع إعادة كتابة الخاصية من خلال كتابة Class جديد و بذلك نستطيع كتابة النصوص الطويلة بالـ JOptionPane من دون أي مشاكل.

```
public static JOptionPane getNarrowOptionPane(int maxCharactersPerLineCount)
{
    // our inner class definition
    class NarrowOptionPane extends JOptionPane {
        int maxCharactersPerLineCount;

        NarrowOptionPane(int maxCharactersPerLineCount) {
            this.maxCharactersPerLineCount = maxCharactersPerLineCount;
        }
        public int getMaxCharactersPerLineCount() {
            return maxCharactersPerLineCount;
        }
    }
    return new NarrowOptionPane(maxCharactersPerLineCount);
}
```

باستخدام الشفرة البرمجية السابقة، لا تستطيع استخدام showMessageDialog. لكن بالشفرة البرمجية التالية سوف نقوم يدوياً باستخدامها:

```
String msg = "This is a really long message. ...";
JOptionPane pane = getNarrowOptionPane(50);
pane.setMessage(msg);
pane.setMessageType(JOptionPane.INFORMATION_MESSAGE);
JDialog dialog = pane.createDialog(
  component, "Width 50");
dialog.show();
```

بالطبع تستطيع إضافة "\n" للرسالة، و لكن سوف تقوم بحساب عدد الحروف لكل سطر.

رسائل كـ Object:

تتساءل لماذا جميع الرسائل تضاف إلى الـ showXXXXDilog و تكون بهيئة Object ؟. الجواب على هذا السؤال يكمن بأن الرسائل لا تكون بهيئة نصوص (String) و لكن تستطيع إضافة الـ Object. المثال التالي نضيف مصفوفة نصية و إضافتها كرسالة:

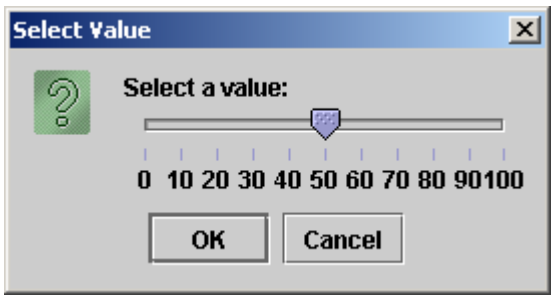
```
String msg[] = {"Welcome", "Home"};
JOptionPane.showMessageDialog(component, msg);
```

تستطيع إضافة Object كالإيقونات و غير ذلك من الأدوات المتوفرة بالـ Swing.
المثال التالي نستطيع استخدام الـ JSlider كـ Object و إضافتها كالتالي:

```
public static JSlider getSlider(final JOptionPane optionPane) {
    JSlider slider = new JSlider();
    slider.setMajorTickSpacing (10);
    slider.setPaintTicks(true);
    slider.setPaintLabels(true);
    ChangeListener changeListener = new ChangeListener() {
        public void stateChanged(ChangeEvent changeEvent) {
            JSlider theSlider = (JSlider)changeEvent.getSource();
            if (!theSlider.getValueIsAdjusting()) {
                optionPane.setInputValue(new Integer(theSlider.getValue()));
            }
        }
    };
    slider.addChangeListener(changeListener);
    return slider;
}
```

لاستخدام الـ Method السابق، نستخدم الشفرة البرمجية التالية:

```
JOptionPane optionPane = new JOptionPane();
JSlider slider = getSlider(optionPane);
Object msg[] = {"Select a value:", slider};
optionPane.setMessage(msg);
optionPane.setMessageType(JOptionPane.QUESTION_MESSAGE);
optionPane.setOptionType(JOptionPane.OK_CANCEL_OPTION);
JDialog dialog = optionPane.createDialog(Options.this, "Select Value");
dialog.show();
Object value = optionPane.getValue();
if (value == null || !(value instanceof Integer)) {
    System.out.println("Closed");
}
else {
    int i = ((Integer)value).intValue();
    if (i == JOptionPane.CLOSED_OPTION) {
        System.out.println("Closed");
    }
    else if (i == JOptionPane.OK_OPTION) {
        System.out.println("OKAY - value is: " +
optionPane.getInputValue());
    }
    else if (i == JOptionPane.CANCEL_OPTION) {
        System.out.println("Cancelled");
    }
}
```



خاصية الأصوات:

توجد هناك أربع أصوات من ضمن الـ JOptionPane:

- OptionPane.errorSound
- OptionPane.informationSound
- OptionPane.questionSound
- OptionPane.warningSound

باستخدام هذه الخصائص، تستطيع استخدام أصوات معينة تحددها كالتالي:

```
UIManager.put("AuditoryCues.playList",
UIManager.get("AuditoryCues.defaultCueList"));
```

خاصية الأصوات غير مفعلة افتراضياً لتفادي لحدوث أي من المشاكل بأحد نظم التشغيل.

مثال متكامل للدرس:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;

public class Options extends JFrame {
    private static class FrameShower

            implements Runnable {
        final Frame frame;

        public FrameShower(Frame frame) {
            this.frame = frame;
        }
        public void run() {
            frame.show();
        }
    }

    public static JOptionPane getNarrowOptionPane(int
maxCharactersPerLineCount) {
        // Our inner class definition
        class NarrowOptionPane extends JOptionPane {
            int maxCharactersPerLineCount;
            NarrowOptionPane(int maxCharactersPerLineCount) {
                this.maxCharactersPerLineCount = maxCharactersPerLineCount;
            }
            public int getMaxCharactersPerLineCount() {
                return maxCharactersPerLineCount;
            }
        }
        return new NarrowOptionPane(maxCharactersPerLineCount);
    }

    public static JSlider getSlider(final JOptionPane optionPane) {
        JSlider slider = new JSlider();
        slider.setMajorTickSpacing (10);
        slider.setPaintTicks(true);
        slider.setPaintLabels(true);
        ChangeListener changeListener = new ChangeListener() {
            public void stateChanged(ChangeEvent changeEvent) {
                JSlider theSlider = (JSlider)changeEvent.getSource();

                if (!theSlider.getValueIsAdjusting()) {
                    optionPane.setInputValue(new
Integer(theSlider.getValue()));
                }
            }
        };
        slider.addChangeListener(changeListener);
        return slider;
    }

    public Options() {
        super("JOptionPane Usage");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        Container contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());
        JButton message = new JButton("Message");
        ActionListener messageListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(Options.this, "Hello, World");
            }
        };
        message.addActionListener(messageListener);
        contentPane.add(message);
```

```
        JButton confirm = new JButton("Confirm");
        ActionListener confirmListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int returnValue =
JOptionPane.showConfirmDialog(Options.this, "Lunch?");
                String message = null;

                if (returnValue == JOptionPane.YES_OPTION) {
                    message = "Yes";
                } else if (returnValue == JOptionPane.NO_OPTION) {
                    message = "No";
                } else if (returnValue == JOptionPane.CANCEL_OPTION) {
                    message = "Cancel";
                } else if (returnValue == JOptionPane.CLOSED_OPTION) {
                    message = "Closed";
                }
                System.out.println("User selected: " + message);
            }
        };
        confirm.addActionListener(confirmListener);
        contentPane.add(confirm);
        JButton inputText = new JButton("Input Text");
        ActionListener inputTextListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String value = JOptionPane.showInputDialog("Name");
                System.out.println("Name: " + value);
            }
        };
        inputText.addActionListener(inputTextListener);
        contentPane.add(inputText);
        JButton inputCombo = new JButton("Input Combo");
        ActionListener inputComboListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String smallList[] = {"Sunday",
                                      "Monday",
                                      "Tuesday",
                                      "Wednesday",
                                      "Thursday",
                                      "Friday",
                                      "Saturday"
                };
                String value = (String)JOptionPane.showInputDialog(
                        Options.this, "Which Day?", "Day",
                        JOptionPane.QUESTION_MESSAGE, null,
                        smallList, smallList[smallList.length-1]);

                System.out.println("Day: " + value);
            }
        };
        inputCombo.addActionListener(inputComboListener);
        contentPane.add(inputCombo);
        JButton inputList = new JButton("Input List");
        ActionListener inputListListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Object largeList[] =
System.getProperties().keySet().toArray();
                String value = (String)JOptionPane.showInputDialog(
                        Options.this, "Which Property?", "Property",
                        JOptionPane.QUESTION_MESSAGE, null,
                        largeList, largeList[largeList.length-1]);

                System.out.println("Property: " + value);
            }
        };
        inputList.addActionListener(inputListListener);
        contentPane.add(inputList);
        JButton all = new JButton("All");
        ActionListener allListener = new ActionListener() {
```

```
            public void actionPerformed(ActionEvent e) {
                String options[] = {"Yes", "Not Now", "Go Away"};
                int value = JOptionPane.showOptionDialog(
                        Options.this,
                        "Lunch?",
                        "Lunch Time",
                        JOptionPane.YES_NO_OPTION,
                        // Message type
                        JOptionPane.QUESTION_MESSAGE,
                        null, // Use default icon for message type
                        options,
                        options[1]);

                if (value == JOptionPane.CLOSED_OPTION) {
                    System.out.println("Closed window");
                } else {
                    System.out.println("Selected: " + options[value]);
                }
            }
        };
        all.addActionListener(allListener);
        contentPane.add(all);
        JButton wide = new JButton("Wide");
        ActionListener wideListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String msg =
                        "This is a really long message. " +
                        "This is a really long message. " +
                        "This is a really long message. " +
                        "This is a really long message. " +
                        "This is a really long message. " +
                        "This is a really long message.";

                JOptionPane pane = getNarrowOptionPane(50);
                pane.setMessage(msg);
                pane.setMessageType(JOptionPane.INFORMATION_MESSAGE);
                JDialog dialog = pane.createDialog(Options.this, "Width
50");
                dialog.show();
            }
        };
        wide.addActionListener(wideListener);
        contentPane.add(wide);
        JButton twoLine = new JButton("Two Line");
        ActionListener twoLineListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String msg[] = {"Welcome", "Home"};
                JOptionPane.showMessageDialog(Options.this, msg);
            }
        };
        twoLine.addActionListener(twoLineListener);
        contentPane.add(twoLine);
        JButton slider = new JButton("Slider");
        ActionListener sliderListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane optionPane = new JOptionPane();
                JSlider slider = getSlider(optionPane);
                Object msg[] = {"Select a value:", slider};
                optionPane.setMessage(msg);
                optionPane.setMessageType(JOptionPane.QUESTION_MESSAGE);
                optionPane.setOptionType(JOptionPane.OK_CANCEL_OPTION);
                JDialog dialog = optionPane.createDialog(Options.this,
"Select Value");
                dialog.show();
                Object value = optionPane.getValue();

                if (value == null || !(value instanceof Integer)) {
                    System.out.println("Closed");
```

```
                } else {
                    int i = ((Integer)value).intValue();
                    if (i == JOptionPane.CLOSED_OPTION) {
                        System.out.println("Closed");
                    } else if (i == JOptionPane.OK_OPTION) {
                        System.out.println("OKAY - value is: " +
optionPane.getInputValue());
                    } else if (i == JOptionPane.CANCEL_OPTION) {
                        System.out.println("Cancelled");
                    }
                }
            }
        };
        slider.addActionListener(sliderListener);
        contentPane.add(slider);
        setSize(300, 200);
    }

    public static void main(String args[]) {

UIManager.put("AuditoryCues.playList",UIManager.get("AuditoryCues.defaultCue
List"));
        JFrame frame = new Options();
        Runnable runner = new FrameShower(frame);
        EventQueue.invokeLater(runner);
    }
}
```

لتحميل الشفرة البرمجية:
http://www.c4arab.com/images/lessons/programming/java/JOptionPane/JOptionPane.zip