الفصل العاشر

إضافة العناصر Items إلى الفورم Form في J2ME، آخر درس ضمن High Level APIs APIs

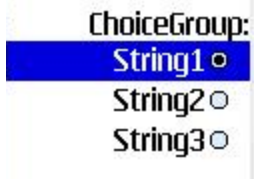
تعرفنا سابقاً على كيفية التعامل مع الفورم Form وكيفية إضافة كائن من النوع TextField إلى الفورم.

سنتعرف الآن على بعض الأصناف الأخرى الشبيهة بالصنف TextField من حيث إضافتها إلى الفورم:

Items: هي الأصناف الأبناء المشتقة من الصنف الأب Item وهذه الأصناف يمكن إضافتها إلى كائن من نوع فورم Form أو من نوع تنبيه Alert ونحن في هذا الشرح سنوضح كيفية إضافتها إلى الكائن Form فقط وسنهمل إضافتها إلى التنبيه Alert وذلك لأن العمليتين متشابهتين.

الأصناف المشتقة من الصنف Item هي ,DateField, ChoiceGroup Gauge, ImageItem, StringItem, TextField . ترث هذه الأصناف طريقتين Methods من الصنف Item هما , setLabel , getLabel.

ChoiceGroup: عبارة عن قائمة تحتوي على عدد من الاختيارات (التي سنقوم بإضافتها) يستطيع المستخدم اختيار أحد الخيارات أو بعض منها حسب نوعية ChoiceGroup التي سنقوم بتحديدها، وهذا الصنف شبيه بالقائمة List التي شرحناها في سابقاً:



DateField: حقل الوقت والتاريخ وهو عبارة عن صندوق يحتوي على التاريخ أو الوقت أو الاثنين معاً حسب ما تريده.

Gauge: المقياس أو المعيار وهو شبيه إلى حد ما بشريط التقدم Progress Bar أو Slider يستطيع المستخدم بزيادة القيمة أو إنقاصها.

ImageItem: هذا الصنف لا يمثل الصورة نفسها وإنما يمثل الموضع الذي ستوضع فيه الصورة ويمكن تشبيهه بالبرواز أو الإطار للصورة حيث يتحكم هذا الصنف بطريقة عرض الصورة مثلاً محاذاة لليمين أو لليسار أو الوسط LAYOUT_CENTER و LAYOUT_LEFT و LAYOUT_RIGHT.

StringItem : عنصر يقوم بعرض نص معين للقراءة فقط غير قابل للتعديل يشبه تماماً عملية إلحاق نص إلى الفورم عبر الطريقة.

CODE

```
public int append(String str)
```

المشروحة في التعامل مع الفورم.

TextField: عنصر صندوق نص يقوم بعرض جزء من النص وإخفاء بقية النص يشبه إلى حد ما عنصر صندوق النص TextBox وقد تعاملنا معه سابقاً في فصل التعامل مع الفورم.

ChoiceGroup :-

المشيد Constructor الخاص بهذا الصنف يكون على أحد الشكلين

CODE

```
ChoiceGroup(String label, int choiceType)
ChoiceGroup(String label, int choiceType, String[]
stringElements, Image[] imageElements)
```

label: العنوان الذي سيكتب في الأعلى قبل كتابة الخيارات.

choiceType: النوع ويتم استخدامه بنفس الطريقة المستخدمة في القائمة List بكتابة اسم الصنف ChoiceGroup ثم نقطة ونوع القائمة كالتالي:

ChoiceGroup.TYPE حيث أن TYPE يمثل أحد الأنواع التالية:

IMPLICIT, MULTIPLE EXCLUSIVE

وقد تم توضيح هذه الأنواع عند شرح القائمة List.

stringElements: مصفوفة تحتوي على الخيارات التي تود إضافتها إلى ChoiceGroup.

imageElements: مصفوفة الأيقونات التي نرغب بإضافتها مع الخيارات وسيظهر الخيار مع صورة الأيقونة وإذا أردنا عدم استخدام أي أيقونة نقوم بوضع القيمة null بدلاً عن المصفوفة.

معظم طرق هذا الصنف شبيهة بالطرق التابعة للصنف List لذا لن نقوم بشرحها وسنقوم فيما بعد بإضافة مثال يوضح كيفية إضافة كائن من هذا الصنف إلى كائن من نوع Form

CODE

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class MyForm extends MIDlet  implements
```

```
CommandListener
 {
  Display display;  // The display for this MIDlet
  Form frm;                        // The Window
  Command exitCommand; // The exit command

 public MyForm()
  {
    frm = new Form("Form Title");
    display = Display.getDisplay(this);
    exitCommand = new Command("Exit",
Command.EXIT, 1);
  }
 public void startApp()
  {
   addChoiceGroupToForm();
إضافة زر الخروج إلى الفورم //
    frm.addCommand(exitCommand);
تشغيل المتصنت للاستجابة لأوامر الفورم //
    frm.setCommandListener(this);
عرض النافذة على شاشة التطبيق //
    display.setCurrent(frm);
  }

  public void  pauseApp() { }

  public void destroyApp(boolean unconditional)  {
}

 public void addChoiceGroupToForm()
  {
```

```
إضافة نص إلى بداية الفورم          //
        frm.append("This Text in Form");
        ChoiceGroup choice = new
ChoiceGroup("Choice Title",
                              Choice.EXCLUSIVE);  //
اختيار واحد من عدة خيارات
كما يوجد النوعان          //
    // Choice.MULTIPLE , Choice.IMPLICIT
إضافة الاختيارات للقائمة          //
        choice.append("String1",null);
        choice.append("String2",null);
        choice.append("String3",null);
إضافة القائمة إلى الفورم          //
        frm.append(choice);
  }

   public void commandAction(Command c,
Displayable s)
       {
if (c == exitCommand)
           {
  destroyApp(false);
  notifyDestroyed();
}
       }
}
```

Gauge:

المشيد Constructor الخاص بهذا الصنف يكون على الشكل:

CODE

```
Gauge(String label, boolean interactive, int maxValue, int initialValue)
```

label: العنوان أو النص الذي يظهر أعلى Gauge.

interactive: يحدد إذا كان المستخدم يستطيع تغيير القيمة أم لا وذلك بوضع القيمة true أو false.

maxValue: أعلى قيمة ممكن أن يصل إليها قيمة Gauge وتبدأ القيم من صفر إلى أعلى قيمة.

initialValue: القيمة الابتدائية التي سيظهر بها Gauge ويجب أن تكون بين الصفر وأعلى قيمة maxValue.

الآن سنوضح بعض الطرق الخاصة بالصنف Gauge.

CODE

```
public int getMaxValue()
```

تعيد هذه الدالة أعلى قيمة ممكن أن تصل إليها قيمة Gauge أو بمعنى آخر فهي تعيد القيمة maxValue التي قمت بضبطها سابقاً

CODE

```
public int getValue()
```

تعيد هذه الدالة القيمة الحالية لـ Gauge.

CODE

```
public boolean isInteractive()
```

من اسم هذه الدالة وطريقة تركيبها نستطيع القول أن هذه الدالة تعيد القيمة true إذا كان المستخدم يستطيع تغيير قيمة Gauge وتعيد false إذا كان لا يستطيع تغيير القيمة

CODE

```
public void setMaxValue(int maxValue)
```

نستطيع ضبط أعلى قيمة للـ Gauge من خلال هذه الدالة.

CODE

```
public void setValue(int value)
```

كذلك نستطيع تغيير قيمة Gauge من خلال هذه الطريقة.

الآن سنضع كود يوضح عمل Gauge.

CODE

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class MyGauge extends MIDlet  implements
CommandListener
 {
  Display display;
```

```
 Form frm;
 Gauge gauge;
 Command cmdExit =  new Command("Exit",
Command.EXIT, 0);
 Command cmdGetValue =  new Command("get
value", Command.SCREEN, 0);

public MyGauge()
 {
  frm = new Form("Form Title");
  display = Display.getDisplay(this);
  gauge = new Gauge("Progress", false, 10, 0);
 }
public void startApp()
 {
  frm.append("Please wait...");
  frm.addCommand(cmdExit);
  frm.setCommandListener(this);
  display.setCurrent(frm);
  addGaugeToForm();
  gauge = new Gauge("Gauge", true, 30, 10);
  frm.append(".................\n");
  frm.append( gauge );
  frm.addCommand(cmdGetValue);
 }

 public void  pauseApp() { }

 public void destroyApp(boolean unconditional)  {
}

//إذا كنت تستخدم
```

```
// Wireless Toolkits
//لن يظهر لديك عمل هذا الإجراء
//يقوم هذا الإجراء بإظهار شريط التقدم يسير من الصفر حتى أعلى قيمة
//يظهر عمل هذا الإجراء عند استخدام محاكي الموبايل لنوكيا
//أو عند تحميل هذا البرنامج مباشرة على الموبايل
 public void addGaugeToForm()
  {
   Thread thread = new Thread();
   frm.append( gauge );
   for (int i=0; i<gauge.getMaxValue(); i++)
    {
     int value = gauge.getValue();
     try  { thread.sleep(300); }
     catch(InterruptedException e){}
     gauge.setValue( value+1 );
    }
  }

  public void commandAction(Command c,
Displayable s)
    {
     if ( s == frm)
        {
         if (c == cmdExit)
          {
 destroyApp(false);
 notifyDestroyed();
}
         else if ( c == cmdGetValue )
          {
           Alert alert = new Alert ( "My Gauge" );
```

```
            alert.setString( "gauge value is : " +
Integer.toString( gauge.getValue() ) );
            display.setCurrent(alert);
           }
        }
    }
}
```

TextField:

المشيد Constructor الخاص بهذا الكائن على الشكل:

CODE

```
TextBox(String title, String text, int maxSize, int
constraints)
```

title: عنوان النص وسيظهر في أعلى صندوق النص.

ونستطيع تغييره باستخدام الطريقة الموجودة في نفس الصنف setTitle وهذه الطريقة ورثها هذا الصنف TextBox من الصنف Screen.

text: وهو النص الذي سيكون موجوداً في داخل صندوق النص ونستطيع التعامل معه من خلال الأمرين getString , setString .

maxSize: عدد الأحرف التي سيتقبلها صندوق النص بحيث لا يمكن للمستخدم إضافة أكثر من العدد المطلوب.

constraints: القيد أو الهيئة للنص المدخل وهناك العديد من القيود. هذه القيود موجودة في الصنف TextField على هيئة حقول ساكنه static ويتم استخدامها على الشكل TextField.Field حيث Field تعبر عن اسم القيد من هذه القيود:

ANY: ويعبر عن جميع الأحرف أو الأرقام.

NUMERIC: يسمح للمستخدم باستخدام الأرقام فقط.

PASSWORD: تظهر للمستخدم نجمة عن كل حرف يكتبه.

وبعـد أن تعرفنـا علـى بنيـة المـشيد سـنذكر الآن بعـض الطـرق التابعـة للـصنف TextBox:

CODE

```
Public void delete(int offset, int length)
```

تقوم هذه الدالة بحذف جزء من النص الموجود في صندوق النص.

offset: بداية النص المراد حذفه.

length: طول النص.

CODE

```
public int getConstraints()
```

تعيد نوع القيد المستخدم على شكل رقم.

CODE

```
public void setConstraints(int constraints)
```

تستطيع تغيير القيد المستخدم في الصندوق باستخدام هذا المنهج.

CODE

```
public int getMaxSize()
```

تعيد الحد الأقصى لعدد الحروف الممكن كتابتها في صندوق النص.

CODE

```
public int setMaxSize(int maxSize)
```

أيضاً نستطيع تغيير الحد الأقصى لعدد الحروف باستخدام هذا المنهج.

CODE

```
public String getString()
```

تعيد هذه الدالة النص المكتوب في الصندوق.

CODE

```
public void setString(String text)
```

يقوم هذا المنهج بوضع النص الذي تريده على الصندوق.

CODE

```
public int size()
```

تعيد هذه الدالة عدد الأحرف المكتوبة حالياً في الصندوق.