منتديات الفريق العربي للبرمجة---- القسم المتخصص بتقنية J2ME لبرمجة الموبايلات

مصعب غيلان - جمهورية اليمن العربية

تعرفنا سابقا على كيفية التعامل مع الفورم **Form**

وكيفية اضافة كائن من النوع **TextField** الى الفورم

سنتعرف الان على بعض الاصناف الاخرى الشبيهة بالصنف **TextField** من حيث اضافتها الى الفورم

Items : هي الاصناف الابناء المشتقة من الصنف الاب **Item** وهذة الاصناف يمكن اضافتها الى كائن من نوع فورم **Form** او من نوع تنبيه **Alert**

ونحن في هذا الشرح سنوضح كيفية اضافتها الى الكائن **Form** فقط وسنهمل اضافتها الى التنبيه **Alert** وذلك لان العمليتين متشابهتين

الاصناف المشتقة من الصنف **Item** هي **ChoiceGroup, DateField, Gauge,**

ImageItem, StringItem, TextField

ترث هذة الاصناف طريقتين **Methods** من الصنف **Item** هما **getLabel , setLabel ,**

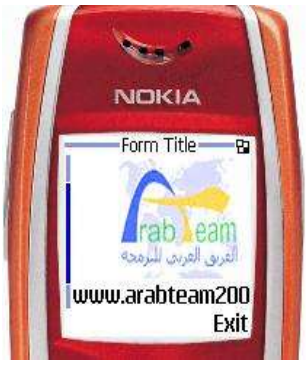
ChoiceGroup : عبارة عن قائمة تحتوي على عدد من الاختيارات **(** التي سنقوم باضافتها **)** يستطيع المستخدم اختيار احد الخيارات او بعض منها حسب نوعية **ChoiceGroup** التي سنقوم بتحديدها

وهذا الصنف شبيه بالقائمة **List** التي شرحناها في الدرس السابق

DateField : حقل الوقت والتاريخ وهو عبارة عن صندوق يحتوي على التاريخ اوالوقت او الاثنين معا حسب ما تريده

Gauge : المقياس او المعيار وهو شبيه الى حد ما بشريط التقدم **Progress Bar** او **Slider**
يستطيع المستخدم بزيادة القيمة او انقاصها

ImageItem : هذالصنف لا يمثل الصورة نفسها وانما يمثل الموضع الذي ستوضع فيه الصورة ويمكن تشبيهه بالبرواز او الاطار للصورة حيث يتحكم هذا الصنف بطريقة عرض الصورة مثلا
محاذاة لليمين او لليسار او الوسط
LAYOUT_CENTER و **LAYOUT_LEFT** و **LAYOUT_RIGHT**



StringItem : عنصر يقوم بعرض نص معين للقراءة فقط غير قابل للتعديل يشبه تماما عملية الحاق نص الى الفورم عبر الطريقة

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

المشروحة في التعامل مع الفورم

TextField : عنصر صندوق نص يقوم بعرض جزء من النص واخفاء بقية النص يشبه الى حد ما عنصر صندوق النص **TextBox**

وقد تعاملنا معه سابقا في درس التعامل مع الفورم

-: ChoiceGroup

المشيد **Constructor** الخاص بهذا الصنف يكون على احد الشكلين

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

label : العنوان الذي سيكتب في الاعلى 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 ان **TYPE** يمثل احد الانواع التالية

EXCLUSIVE, IMPLICIT, MULTIPLE

وقد تم توضيح هذة الانواع عند شرح القائمة **List**

stringElements : مصفوفة تحتوي على الخيارات التي تود اضافتها الى **ChoiceGroup**

imageElements : مصفوفة الايقونات التي نرغب باضافتها مع الخيارات وسيظهر الخيار مع صورة الايقونة

واذا اردنا عدم استخدام اي ايقونة نقوم بوضع القيمة **null** بدلا عن المصفوفة

معظم هذه طرق هذا الصنف شبيهة للطرق التابعة للصنف **List** لذا لن نقوم بشرحها وسنقوم فيما بعد باضافة مثال يوضح كيفية اضافة كائن من هذا الصنف الى كائن من نوع **Form**

```
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 الى الفورم
frm.addCommand(exitCommand);
// تشغيل المتصنت للاستجابة لاوامر 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 الى 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 الى الفورم
frm.append(choice);
}

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

-:
المشيد Constructor الخاص بهذا الصنف يكون على الشكل

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

label : العنوان او النص الذي يظهر اعلى Gauge
interactive : يحدد اذا كان المستخدم يستطيع تغيير القيمة ام لا وذلك بوضع القيمة true او false
maxValue : اعلى قيمة ممكن يصل اليها قيمة Gauge وتبدأ القيم من صفر الى اعلى قيمة
initialValue : القيمة الابتدائية التي سيظهر بها Gauge ويجب ان تكون بين الصفر واعلى قيمة maxValue

الان سنوضح بعض الطرق الخاصة بالصنف Gauge

public int getMaxValue()

تعيد هذة الدالة اعلى قيمة ممكن ان تصل اليها قيمة Gauge
او بمعنى اخر فهي تعيد القيمة maxValue التي قمت بضبطها سابقا

public int getValue()

تعيد هذة الدالة القيمة الحالية للـ Gauge

public boolean isInteractive()

من اسم هذة الدالة وطريقة تركيبها نستطيع القول ان هذة الدالة تعيد القيمة true اذا كان المستخدم يستطيع تغيير قيمة Gauge وتعيد false اذا كان لا يستطيع تغيير القيمة

public void setMaxValue(int maxValue)

نستطيع ضبط اعلى قيمة للـ Gauge من خلال هذة الدالة

public void setValue(int value)

كذلك نستطيع تغيير قيمة Gauge من خلال هذة الطريقة

الان سنضع كود يوضح عمل Gauge

```
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) { }

// اذا كنت تستخدم
// Wireless Toolkits
// لن يظهر لديك عمل هذا الاجراء
```

```
// يقوم هذا الاجراء باظهار شريط التقدم يسير من الصفر حتى اعلى قيمة
// يظهر عمل هذا الاجراء عند استخدام محاكي الموبايل لنوكيا
// او 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

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

title : عنوان النص وسيظهر في اعلى صندوق النص
ونستطيع تغييره باستخدام الطريقة الموجودة في نفس الصنف setTitle وهذة الطريقة ورثها هذا الصنف TextBox من الصنف Screen
text : وهو النص الذي سيكون موجودا في داخل صندوق النص
ونستطيع التعامل معه من خلال الامرين setString , getString
maxSize : عدد الاحرف التي سيتقبلها صندوق النص بحيث لا يمكن للمستخدم اضافة اكثر من العدد المطلوب

constraints : القيد او الهيئة للنص المدخل وهناك العديد من القيود
هذة القيود موجودة في الصنف TextField على هيئة حقول ساكنه static
ويتم استخدامها على الشكل TextField.Field حيث Field تعبر عن اسم القيد
من هذة القيود :
ANY : ويعبر عن جميع الاحرف او الارقام
NUMERIC : يسمح للمستخدم باستخدام الارقام فقط
PASSWORD : تظهر للمستخدم نجمة عن كل حرف يكتبه

وبعد ان تعرفنا على بنية المشيد سنذكر الان بعض الطرق التابعة للصنف TextBox

public void delete(int offset, int length)

تقوم هذة الدالة بحذف جزء من النص الموجود في صندوق النص
offset : بداية النص المراد حذفه
length : طول النص

public int getConstraints()

تعيد نوع القيد المستخدم على شكل رقم

public void setConstraints(int constraints)

تستطيع تغيير القيد المستخدم في الصندوق باستخدام هذا المنهج

public int getMaxSize()

تعيد الحد الاقصى لعدد الحروف الممكن كتابتها في صندوق النص

public int setMaxSize(int maxSize)

ايظا نستطيع تغيير الحد الاقصى لعدد الحروف باستخدام هذا المنهج

public String getString()

تعيد هذة الدالة النص المكتوب في الصندوق

public void setString(String text)

يقوم هذا المنهج بوضع النص التي تريده على الصندوق

public int size()

تعيد هذة الدالة عدد الاحرف المكتوبة حاليا في الصندوق

هذا مثال ل Gauge بحيث يستطيع المستخدم زيادتها او تنقيصها حتى يشتغل على Wireless Toolkit

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

public class GaugeTracker extends MIDlet implements ItemStateListener

{
private Gauge mGauge;
private StringItem mStringItem;


public GaugeTracker()
{

int iValue = 5;
mGauge = new Gauge("Gauge Title",true,20,iValue);
mStringItem = new StringItem(null,"[value]");
itemStateChanged(mGauge);

}

public void itemStateChanged(Item item)
{
if(item == mGauge)
mStringItem.setText("Value = "+mGauge.getValue());

}
public void startApp()

{
Form form = new Form("GaugeTracker");

//Now add selected items
form.append(mGauge);
form.append(mStringItem);
form.setItemStateListener(this);
Display.getDisplay(this).setCurrent(form);

}


public void pauseApp(){}
```

public void destroyApp(boolean unconditional){}

}