الفصل الثامن

التعامل مع القوائم List في J2ME، ضمن High Level APIs

سنتعرف في هذا الجزء على كيفية التعامل مع القوائم List.

يرث هذا الصنف عدد من الأعضاء Members من الصنف Screen ومن هذه الطرق setTicker, setTitle كما أنه يستخدم الواجهة Choice على الشكل Choice implements.

List: عبارة عن شاشة تحتوي على مجموعة خيارات Choices يستطيع المستخدم اختيار أي واحد منها.

وهناك ثلاثة أشكال لهذه القائمة:

IMPLICIT: قائمة من الأزرار يستطيع المستخدم الضغط على أي زر من القائمة ويتم معرفة الزر الذي تم كبسه باستخدام الطريقة getSelectedIndex التابعة للكائن المنشأ من الصنف List.

أما SELECT_COMMAND: فيعبر عن الزر اختيار select الذي يأتي مترافقاً مع هذا النوع من القوائم.

EXCLUSIVE: تمثل قائمة خيارات يستطيع المستخدم اختيار خيار واحد فقط من هذه القائمة وتكون قيمة الخيار الذي تم اختياره true وبقية الخيارات false وهذه القائمة شبيهة بال Option في الفيجوال بيسك ويمكن معرفة قيمة كل خيار هل تم اختياره أم لا بواسطة الطريقة isSelected حيث نعطيها رقم الخيار في القائمة وتعيد لنا قيمة true أو false أي هل تم اختياره أم لا.

MULTIPLE: قائمة خيارات من النوع المتعدد وتشبه إلى حد ما CheckBox في الفيجوال بيسك ويتم التعامل معها بنفس التعامل مع النوع EXCLUSIVE.

لنأخذ مثالاً أنه يوجد لدينا قائمة من النوع الأول IMPLICIT فإننا سنعرف أن المستخدم قام بالضغط على زر معين كما في هذا المثال:

obeikandl.com

CODE

```
public void commandAction (Command c,
Displayable d) {
   if (d == myList) {
      if (List.SELECT_COMMAND ==  ) {
         // do Something
      }
   }
}
```

يجب ملاحظة أن كل هذه القوائم يتم إنشاؤها بنفس الطريقة ولكن فقط يختلف النوع.

هذا الكائن يحتوي على مشيدين Constructor تستطيع اختيار أحدهما.

CODE

```
List(String title, int listType)
List(String title, int listType, String[]
stringElements, Image[] imageElements)
```

title: العنوان الذي سيظهر أعلى الصندوق.

listType: نوع هذه القائمة وسيترتب على النوع شكل العرض.

وستكتب على الشكل List TYPE حيث أن TYPE ستمثل إما EXCLUSIVE, IMPLICIT, MULTIPLE.

stringElements: مصفوفة نصوص الاختيارات التي ستظهر كعناصر على القائمة.

imageElements: مصفوفة الأيقونات أو الصور التي ستظهر مترافقة مع العناصر.

وإذا لم نرد استخدام أي أيقونه سنضع القيمة null بدلاً عن المصفوفة.

obeikandl.com

وبعد أن تعرفنا على بنية المشيد سنذكر الآن بعض الطرق التابعة للصنف List وكيفية استخدامها.

CODE

```
public void addCommand(Command cmd)
```

يرث الصنف List هذه الطريقة من الصنف Displayable.

CODE

```
public int append(String stringPart, Image
imagePart)
```

تقوم هذه الطريقة بإضافة عنصر ما إلى القائمة وتعيد رقم العنصر في القائمة.

stringPart: نص العنصر الذي نقوم بإضافته.

imagePart: الأيقونة التي ستظهر مترافقة مع العنصر المضاف.

وإذا أردنا عدم إظهار أي أيقونة سنستخدم القيمة null.

CODE

```
public int getSelectedIndex()
```

تعيد هذه الدالة رقم العنصر الذي تم اختياره حالياً.

CODE

```
public boolean isSelected(int elementNum)
```

تعيد هذه الدالة قيمة true إذا كان هذا العنصر قد تم اختياره أو false إذا لم يتم اختياره.

obeikandi.com

elementNum: تعبر عن رقم العنصر.

والآن بعد شرحنا كيفية التعامل مع القائمة List بقي علينا أن نقوم بوضع مثال يوضح لنا كل هذا.

CODE

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

public class MyList extends MIDlet implements
CommandListener
{
 private Display display;

Command cmdExit = new Command("Exit",
Command.EXIT, 0);
Command cmdBack = new Command("Back",
Command.BACK, 0);
Command cmdInfo = new Command("Info",
Command.SCREEN, 0);

  List mainList;
  List subList;

  String[] choices;

  Ticker ticker;

 public MyList  ()
  {
   display =Display.getDisplay(this);
   mainList = new List("Main List",List.IMPLICIT);
```

obeikandl.com

```
  ticker = new Ticker (" www.arabteam2000.com
");
  choices = new String[3];
  choices[0] = "choice1";
  choices[1] = "choice2";
  choices[2] = "choice3";
 }

 public void startApp()
  {
  mainList.append("EXCLUSIVE",null);
  mainList.append("MULTIPLE",null);
  mainList.addCommand(cmdExit);
  mainList.setTicker(ticker);
  mainList.setCommandListener(this);
  display.setCurrent(mainList);
 }

 private void whatSelected( List list )
  {
   String selected = new String();
   for ( int i = 0; i < list.size(); i++ )
    if ( list.isSelected( i ) )
      selected = selected + " , " + list.getString( i
);
    Alert alert = new Alert("choices");
    if ( selected.length() > 0 )
      alert.setString(" you selected : " + selected);
    else
      alert.setString("you didn`t select");
    display.setCurrent(alert);
  }
```

obeikandl.com

```
 public void pauseApp(){ }
 public void destroyApp(boolean unconditional){}

/************** implements CommandListener
*****************/
public void commandAction(Command c,
Displayable s)
{
  if ( s == mainList )
    {
     if (c == cmdExit)
       {
        destroyApp(false);
        notifyDestroyed();
       }
     else if ( c == List.SELECT_COMMAND )
      if ( mainList.getSelectedIndex() == 0 )
       {
        subList = new List("EXCLUSIVE List",
List.EXCLUSIVE, choices, null);
        subList.addCommand(cmdBack);
        subList.addCommand(cmdInfo);
        subList.setCommandListener(this);
        display.setCurrent(subList);
       }
      else if (mainList.getSelectedIndex() == 1)
       {
        subList = new List("MULTIPLE
List",List.MULTIPLE, choices, null);
        subList.addCommand(cmdBack);
        subList.addCommand(cmdInfo);
```

onelkandi.com

```
            subList.setCommandListener(this);
            display.setCurrent(subList);
          }
      }
    else if ( c == cmdInfo )
      whatSelected( (List) s );
    else if ( c == cmdBack )
      {
       mainList.setCommandListener(this);
       display.setCurrent(mainList);
      }
  }  // end commandAction
}  // end Class
```

obeikandl.com

obeikandl.com