من اعداد مصعب غيلان- جمهورية اليمن العربية

اضافة الازرار

Commands كيفية بناء التطبيق مع استخدام

بعد ان تعرفنا على كيفية بناء التطبيق الاول وكيفية تحويل هذا الكودالى تطبيق يعمل على الجوال سنقوم الان بشرح لكيفية اضافة الازار والتعامل معها بعد هذا الشرح يفترض بك ان تكون قادر على بناء تطبيق صغير من تلقاء نفسك يجب عليك اولا انشاء زر جديد ثم إضافة هذا الزر الى الواجهة التي تريد عرضها بعد ذلك ستضع الكود الذي تود تنفيذة عند الضغط على هذا الزر

1 - تعريف الازرار في البرنامج
Command اولا سنقوم بتعريف كائن جديد من نوع

```
Command command;
```

2 - ونقوم باعطاء الزر بعض الخصائص مثل الاسم الذي سيظهر للمستخدم new ثم نقوم بانشاء هذا الكائن باستخدام الامر وكذلك نوع هذا الزر واولوية الظهور

```
command = new Command(String label, int commandType, int priority)
```

ويمكن اختصار الطريقتين السابقتين على الشكل

```
Command command = new Command(String label, int commandType, int priority)
```

انشأنا كائن من نوع زر ثم اعطيناه بعض الخصائص

3 - ربط هذا الزر مع واجهة محددة
الذي نريد عرض الزر معها باستخدام الطريقة UI بعد ذلك نقوم بإضافة هذا الزر الى الواجهة

```
UI.addCommand(command);
```

سنكتب الامر على الشكل TextBox box مثلا لو كانت الواجهة

```
box.addCommand(command);
```

4 - اضافة الكود الذي سينفذ عند الضغط على الزر
implements Interface من اجل اضافة كود لزر امر معين يجب علينا استخدام
في الكلاس الذي سنستخدم فيه الازرار بهذا الشكل CommandListener اي اننا سنضمن الواجهة

```
public class HelloMidlet extends MIDlet implements CommandListener {
```

commandAction(Command c, Displayable s) الطريقة المجردة CommandListener يوجد في الواجهة

implements CommandListener لهذا يجب تضمين هذة الطريقة في تطبيقنا لاننا استخدمنا

تعيد لنا هذة الواجهة كائنان

وهو الزر الذي قمنا بضغطه Command c الاول
الذي تحوي هذا الزر UI وهي الواجهة Displayable s الثاني

محددة UI لكي يتصنت على كبسة الزر يجب اولا توجيه هذا المتنصت الى Listener تعمل هذة الواجهة على تشغيل المتنصت setCommandListener من خلال الامر
UI اي اننا سنخبر المتنصت ان ينفذ الكود عند حدوث كبسة زر ما من تلك الواجهة

والان سنضع هذا المثال المطور عن مثالنا السابق ونعود بعد ذلك لمناقشة الاشياء الجديدة فيه

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

import javax.microedition.lcdui.*;


public class HelloMidlet extends MIDlet implements CommandListener
{

private Display display;
Command command = new Command("Exit", Command.EXIT, 0);

TextBox box;

public void HelloMidlet(){}

public void startApp(){


display =Display.getDisplay(this);

box =new TextBox("Simple Example ","Hello ",20,0);
box.addCommand(command);
box.setCommandListener(this);
display.setCurrent(box);
}

public void pauseApp(){ }

public void destroyApp(boolean unconditional){}

/************* implements CommandListener ***************/

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

```
}

}
    public class HelloMidlet extends MIDlet implements CommandListener
```

يعني ان هذ ا الكلاس سيستخدم واجهة او اكثر من واجهة تفصل بينهما فاصلة صغيرة : implements
Interface1,Interface2
عبارة عن واجهة تستخدم لاضافة رد فعل معين تجاه كبسة زر معين : CommandListener

```
    Command command = new Command("Exit", Command.EXIT, 0);
```

Exit والذي سيظهر للمستخدم بالاسم command قمنا بانشاء الكائن

```
    box =new TextBox("Simple Example ","Hello ",20,0);
```

Simple Example عنوان هذا الصندوق TextBox انشاء كائن من نوع صندوق نص
وعدد الحروف التي يسمح بها 20 حزف
وهذا يعني ان صندوق النص سيتقبل اي حرف او رقم TextField.ANY اما القيمة 0 فتمثل

```
    box.addCommand(command);
```

box قمنا باضافة الزر الذي انشأناه سابقا الى
UI الموجودة في جميع الواجهات addCommand باستخدام الطريقة

```
    box.setCommandListener(this);
```

box اخبرنا المتنصت بان يقوم بالتنصت على الواجهة
UI الموجود في جميع الواجهات setCommandListener باستخدام الامر
فتعني الكلاس الحالي الذي نحن فيه this اما الكلمة
وهذا يعني اننا سوف نقوم بمعالجة كبسة الزر في هذا الكلاس

```
    public void commandAction(Command c, Displayable s)
```

CommandListener هذة هي الطريقة التي تعمل مع الواجهة
التي تحوي هذا الزر UI وتعيد بارامتران الاول هو الزر الذي تم كبسه والثاني الواجهة

```
    if ( c == command && s == box )
```

التحقق من ان الزر الذي تم كبسه والتأكد من الواجهة التي تحوي هذا الزر

```
    destroyApp(false); notifyDestroyed();
```

destroyApp هذان الامران يقومان باغلاق التطبيق الاول يقوم بتشغيل عملية الاغلاق باستدعاء الدالة

والثاني يقوم بالتحسس لعملية الاغلاق

6- التعامل مع الواجهات High Level APIs في J2ME,
مقدمة نظرية

بعد ان تعرفنا على كيفية بناء التطبيق
الى التطبيق Commands وكذلك تعرفنا على كيفية اضافة الازرار

سنتعرف الان على بقية الواجهات التي تستخدم للعرض مثل
TextBox , List , Form ,Canvas , Graphics

ويتم تقسيم هذة الواجهات الى قسمين

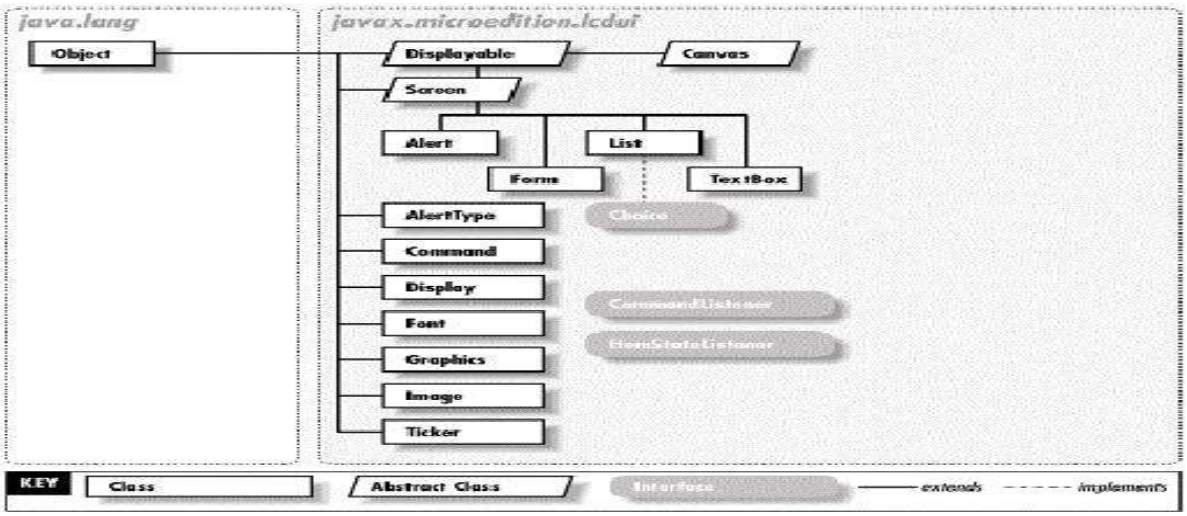
1 - high-level API
2 - low-level API

تستخدم الاولى النمط الحرفي للعرض
Graphic بينما تستخدم الثانية النمط الرسومي

وهي J2ME وكل هذة الاصناف موجودة في حزمة واحدة لل

Package javax.microedition.lcdui

ويمكنك التعرف على هيكلية بناء هذة المكتبة بالنظر الى الصورة



Interfaces والواجهات Classes بالنظر الى الهيكلية سنتعرف على الاصناف
توجد ثلاث واجهات في هذة المكتبة
فقط List ويستخدمها الصنف : Choice
للتطبيق Commands وقد تحدثنا عنه سابقا عند اضافة الازرار : CommandListener
تستخدم للتنصت على كبسة زر معين
فقط Form ويستخدمها الصنف : ItemStateListener

بالاضافة الى وجود العديد من الاصناف الذي سنتعرف على بعض منها لاحقا