J2ME: a MIDlet example

Contributed by Administrator
lunedì, 08 settembre 2008
Last Updated lunedì, 21 dicembre 2015

A MIDlet is a Java application for embedded devices installing a J2ME Virtual Machine.

As you can see from the following example in fact the main class mandatory extends MIDlet class with public class NomeClasse extends MIDlet instruction.

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

import javax.microedition.lcdui.*;


public class Hello extends MIDlet

{

  public Hello() {}

  public void pauseApp() {}

  public void destroyApp(boolean ignore) {}

  public void startApp()

  {

    Display.getDisplay(this).setCurrent(

      new TextBox("Hello", "World", 9, 0));

  }

}
```

The first and second line of our code show the package required for the MIDlet to run: the abstract MIDlet class and the graphic management library (lcdui) useful to draw elements on the device display.

(adsbygoogle = window.adsbygoogle || []).push({});

We can also remark another typical aspect of J2ME applications: that is the compulsory definition of startApp, pauseApp e destroyApp methods.

They underline in fact what is usually called the MIDlet life cycle.

The startApp method initializes all the necessary resources and contains the base instructions for the MIDlet to run and is called when the application is started or re-started.

PauseApp manages instead a MIDlet pause whose concrete behaviour may depend on the specific device (such as closing the flip for a mobile phone).

DestroyApp manages finally the application closure and its behaviour depends on the boolean parameter passed: true to release and de-allocate all the resources; false to keep these resources for a little time more (to continue playing an audio track till the end for example).

The real MIDlet ending notification is however managed with notifyDestroyed() method as we will se in the next example.

We can now add some more elements to our MIDlet. That will be useful to explain the Display and Form notions.

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

import javax.microedition.lcdui.*;



public class Hello extends MIDlet

{

  private Display mDisplay;

  private Form mForm;



  public Hello() {}

  public void pauseApp() {}

  public void destroyApp(boolean ignore) {

    notifyDestroyed();

  }

  public void startApp()

  {

    mDisplay = Display.getDisplay(this);

    mForm = new Form ("Form 1");

    StringItem si = new StringItem("hello"," by alex");

    mForm.append(si);
    mDisplay.setCurrent(mForm);


  }

}
```

As you can see also from the first and simple example the Display is one of the first instruction we have called. That is in fact the device display abstraction where we can put our graphics elements.

A Form is instead a container (at a level just below than our Display) for a specific list of graphic elements having a particular function in our MIDlet (just a StringItem in our example).

You can easily compare a Form to a window PC application gathering the different graphic elements we want to show time by time in our display. A MIDlet application could in fact shows the main options in a Form and the other preferences in another, exchanging the visualization depending on user actions and by the setCurrent(myForm) Display method.

Finally as you can see we have added also the notifyDestroyed() method, according to the explanation just above.