



## Interfície gràfica d'usuari. LCDUI



# Java ME API docs

## ♦ És molt útil disposar de la documentació (javadocs) de Java ME

- ♦ Les podeu trobar online a:

<http://java.sun.com/javame/reference/apis.jsp#api>

- ♦ Amb Netbeans trobareu la documentació a:

`~/netbeans-6.5/mobility8/WTk2.5.2/docs/api`

- ♦ LCDUI forma part del perfil MIDP. Dues versions

- MIDP 1.0 (JSR 37)
- MIDP 2.0 (JSR 118)

- ♦ Consulteu

- Wiki del curs



# Display

- ♦ **La interfície gràfica amb CLDC es realitza amb l'API LCDUI**
  - ♦ LCDUI = Limited Connected Devices User Interface
  - ♦ El codi es troba al paquet **javax.microedition.lcdui**
- ♦ **L'objecte Display representa una pantalla de mòbil**
  - ♦ Permet accedir a la pantalla i al teclat
  - ♦ Cada MIDlet té associat un sol *Display*
- ♦ **Es pot obtenir el display amb:**

```
Display display = Display.getDisplay(midlet);
```

- ♦ Normalment s'executa des de un MIDlet i pot ser un Bean Pattern:

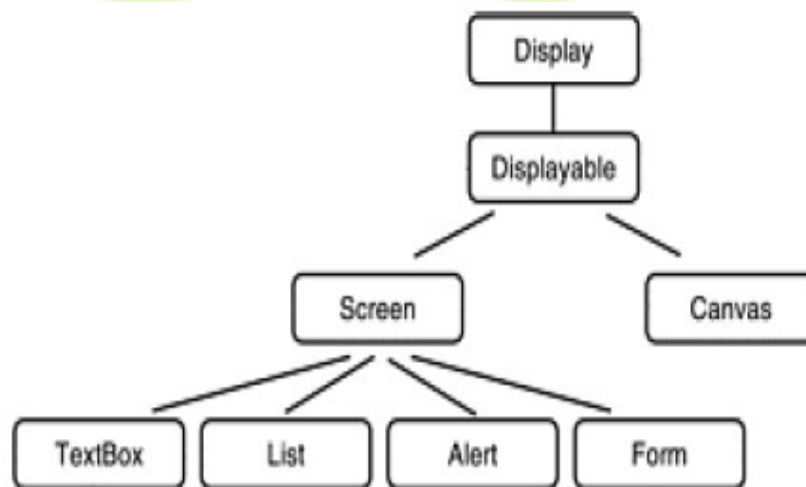
```
public Display getDisplay () {
    return Display.getDisplay(this);
}
```



# Components gràfics

## ♦ Són els objectes que es poden mostrar per pantalla

- ♦ Tots són extensions (hereten) de la classe abstracta Displayable
- ♦ Només es pot mostrar un component gràfic a l'hora.
- ♦ **Canvas (baix nivell)**
  - Dibuix lliure
- ♦ **Screen (alt nivell)**
  - Pantalles de la interfície gràfica





# Components gràfics

## ♦ Alt nivell (Screen)

- ♦ Components predefinitos (hereten d'*Screen*)
  - Llistes, formularis, alertes, camps de text...
- ♦ Aplicacions més portables. Natiu de Java ME.
- ♦ Adequats per a interfícies gràfiques d'aplicacions

## ♦ Baix nivell

- ♦ Components a mida: *Canvas*
- ♦ Cal especificar per codi com mostrar el contingut de la pantalla.
- ♦ Tenim control dels esdeveniments de teclat
- ♦ Es redueix la portabilitat
- ♦ Adequat per a jocs



# Components gràfics

- ♦ **Es determina quin és el component gràfic a mostrar per pantalla amb:**

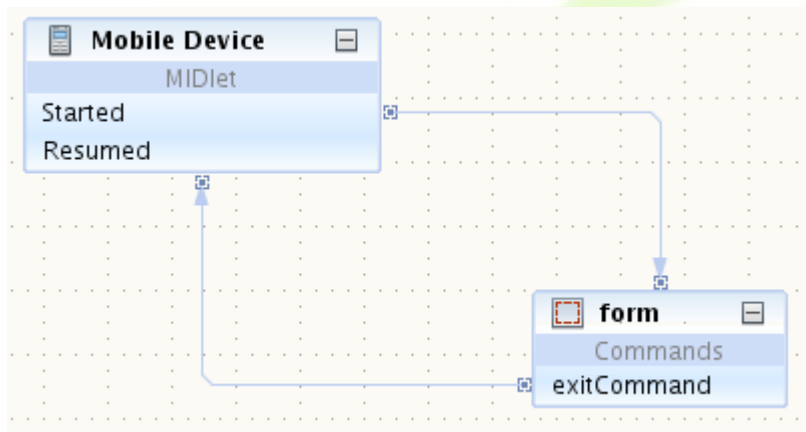
```
Display.setCurrent ( displayable);
```

- ♦ Els components d'alt nivell hereten de la classe abstracta Screen (buida) que hereta de Displayable:
  - Poden tenir un títol
  - Poden tenir un ticker
  - Poden tenir zero o més comandes i els corresponents *Listeners* associats a aquestes comandes.



# Components gràfics i Netbeans

- La pestanya Flow permet dissenyar la interfície gràfica



- Codi font

```

public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
    // write pre-switch user code here
    Display display = getDisplay();

    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
    // write post-switch user code here
}
    
```



# Camps de text

```
TextBox tb = new TextBox("Contraseña",
    "", 8, TextField.ANY |
    TextField.PASSWORD);
Display d = Display.getDisplay(this);
d.setCurrent(tb);
```







# Llistes

```
List l = new List("Menu",
    Choice.IMPLICIT);
l.append("Nuevo juego", null);
l.append("Continuar", null);
l.append("Instrucciones", null);
l.append("Hi-score", null);
l.append("Salir", null);
Display d =
    Display.getDisplay(this);
d.setCurrent(l);
```



Implícita



Múltiple



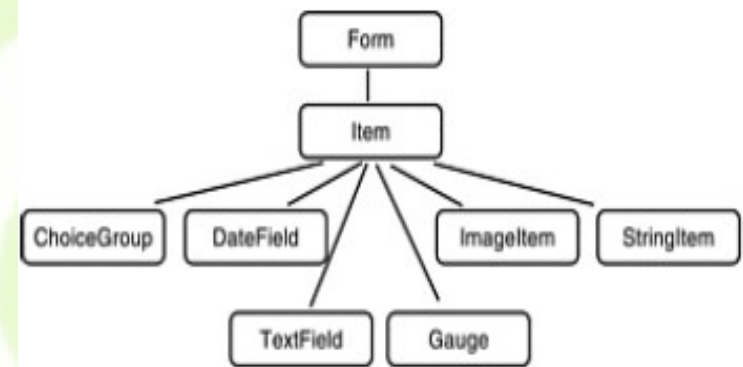
Exclusiva



# Formularis

```

Form f = new Form("Formulario");
Item itemEtiqueta = new StringItem(
    "Etiqueta:",
    "Texto de la etiqueta");
Item itemTexto = new TextField(
    "Telefono:", "", 8,
    TextField.PHONENUMBER);
Item itemFecha = new DateField(
    "Fecha",
    DateField.DATE_TIME);
Item itemBarra = new Gauge("Volumen",
    true, 10, 8);
f.append(itemEtiqueta);
f.append(itemTexto);
f.append(itemFecha);
f.append(itemBarra);
Display d = Display.getDisplay(this);
d.setCurrent(f);
  
```





# Alertes

```
Alert a = new Alert("Error",  
    "No hay ninguna nota seleccionada",  
    null, AlertType.ERROR);  
Display d = Display.getDisplay(midlet);  
d.setCurrent(a, d.getCurrent());
```





# Imatges amb MIDP

- ♦ **Es poden incloure imatges en molt components gràfics**
  - ♦ Les imatges s'encapsulen a la classe *Image*
  - ♦ L'únic format reconegut per l'estàndard és PNG
- ♦ **Es poden obtenir a partir de fitxer continguts dins del fitxer JAR**

```
Image img = Image.createImage("/logo.png");
```



# Comandes d'entrada

- ♦ **L'entrada dels usuaris es realitza mitjançant comandes (classe Command)**
  - ♦ L'única interfície és el teclat?





# Creació de comandes

- **Per afegir una comanda a un component gràfic**

```
TextBox tb = new TextBox("Login", "", 8, TextField.ANY);  
Command cmdOK = new Command("OK", Command.OK, 1);  
Command cmdAyuda = new Command("Ajuda", Command.HELP, 1);  
Command cmdSalir = new Command("Sortir", Command.EXIT, 1);  
Command cmdBorrar = new Command("Esborrar", Command.SCREEN, 1);  
tb.addCommand(cmdOK);  
tb.addCommand(cmdAyuda);  
tb.addCommand(cmdSalir);  
tb.addCommand(cmdBorrar);  
Display d = Display.getDisplay(this);  
d.setCurrent(tb);
```



# Model d'esdeveniments

## ♦ En Java per tractar els esdeveniments s'utilitzen *Listeners*

- ♦ Component que escolta un determinat esdeveniment
- ♦ Quan succeïx l'esdeveniment s'executa el codi del *Listener*.

## ♦ Tres tipus de *Listeners* a Java ME

- ♦ **CommandListener**
- ♦ **ItemCommandListener**
- ♦ **ItemStateListener**

## ♦ El MIDlet ha d'implementar la classe *Listener* desitjada

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

- ♦ Cal implementar els mètodes de la interfície i
- ♦ Registrar el Listener al component gràfic el qual ha de rebre els esdeveniments



# Listener de comandes

## ♦ Exemple

```
class ListenerLogin implements CommandListener {  
    public void commandAction(Command c, Displayable d) {  
        if(c == cmdOK) {  
            // Aceptar  
        } else if(c == cmdCancelar) {  
            // Cancelar  
        } else if(c == cmdSalir) {  
            // Salir  
        } else if(c == cmdAyuda) {  
            // Ayuda  
        } else if(c == cmdBorrar) {  
            // Borra r  
        }  
    }  
}
```

```
tb.setCommandListener(new ListenerLogin());
```





# Listener de comandes

## ♦ Exemple Hola Mon! de Netbeans

```
public class HelloMIDlet extends MIDlet implements CommandListener {
    ...
    public void commandAction(Command command, Displayable displayable) {
        // write pre-action user code here
        if (displayable == form) {
            if (command == exitCommand) {
                // write pre-action user code here
                exitMIDlet();
                // write post-action user code here
            }
        }
        // write post-action user code here
    }

    public Form getForm() {
        if (form == null) {
            // write pre-init user code here
            form = new Form("Welcome", new Item[] { getStringItem() });
            form.addCommand(getExitCommand());
            form.setCommandListener(this);
            // write post-init user code here
        }
        return form;
    }
}
```



## Reconeixement 3.0 Unported

### Sou lliure de:



copiar, distribuir i comunicar públicament l'obra



fer-ne obres derivades

### Amb les condicions següents:



**Reconeixement.** Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

Advertiment

Els drets derivats d'usos legítims o altres limitacions reconegudes per llei no queden afectats per l'anterior  
Això és un resum fàcilment llegible del text legal (la llicència completa).

<http://creativecommons.org/licenses/by/3.0/deed.ca>