

# Mobile and Wireless Systems Programming

Graphical User Interfaces  
High level API



# Mobile UI

- MIDP user interface : LCDUI
- Adapted for small screens, performance and memory constraints, input user mode...
- Package : `javax.microedition.lcdui`
- High level (Form, TextBox...)
- `javax.microedition.lcdui.Display` = the screen
- `javax.microedition.lcdui.Displayable` = item that can be displayed

# Ready-made screens

- 4 ready-made screens
  - TextBox : a title and a text area that fills the whole screen.
  - Alert : suitable for showing short messages to the user, it can show an icon and make a sound.
  - List : shows a list of items that can have associated icons. It can be used as a menu.
  - Form : can contain a combination of items, like checkboxes, text fields, ../images, and others.

# TextBox

- TextBox(String title, String text, int maxSize, int constraints)
- The TextBox class is a Screen that allows the user to enter and edit text.
- Input constraints :
  - TextField.ANY
  - TextField.EMAILADDR
  - TextField.NUMERIC
  - TextField.PHONENUMBER
  - TextField.URL
  - TextField.DECIMAL
- More constraints : SENSITIVE, UNEDITABLE...

# Alert

- Alert(String title)
- Alert(String title, String alertText, Image alertImage, AlertType alertType)
- AlertType :
  - AlertType.ALARM
  - AlertType.CONFIRMATION
  - AlertType.ERROR
  - AlertType.INFO
  - AlertType.WARNING

# List

- Class : `javax.microedition.lcdui.List`
- Screen containing list of choices
- `List(String title, int listType, String[] stringElements, Image[] imageElements)`
- `listType` :
  - IMPLICIT
  - EXCLUSIVE
  - MULTIPLE

# Form

- Class `:javax.microedition.lcdui.Form`
- Screen that contains items : Image, TextField... (Any Item's subclass)
- `Form(String title, Item[] items)`

## Item direct Subclasses

- ChoiceGroup : a group of selectable elements
- CustomItem : subclassing CustomItem to define a custom item
- DateField : Editable component for presenting date and time
- Gauge : a graphical display, such as a bar graph, of an integer value
- ImageItem : an item that can contain an image
- Spacer : blank, non-interactive item that has a settable minimum size
- StringItem : an item that can contain a string
- TextField : an editable text component



# Image

```
try {  
    Image i = Image.createImage("/res/img.png");  
} catch (IOException ex) {  
    ex.printStackTrace();  
}
```

# Command

Command(String shortLabel, String longLabel, int commandType, int priority)

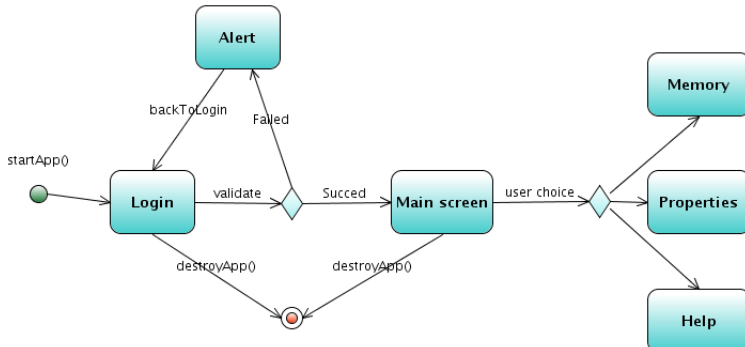
- BACK : returns the user to the previous screen
- CANCEL : standard negative answer to a dialog
- EXIT : exiting from the application
- HELP : specifies a request for on-line help
- ITEM : items of the Screen or the elements of a Choice
- OK : answer to a dialog implemented by current screen
- SCREEN : application-defined command
- STOP : stop some currently running process, operation...

# CommandListener

- Interface
- `void commandAction(Command c, Displayable d)`
- Indicates that a command event has occurred on Displayable d

## Example

Workshop : an application to explore and display mobile phone properties.



## Application's screens

- Login :
  - 2 TextFields (Login, passwd)
  - Command : Exit, Clear, Validate
- Main Screen :
  - List (Memory, Properties, Help)
  - Command : Exit, Ok
- Alert : If the login/password are wrong (Image + Command)
- Memory : StringItem + Back Command
- Properties : StringItem + Back Command
- Help : TextBox with information (Back command)

## Mobile information

- `System.getProperty("...");`
  - `microedition.encoding`
  - `microedition.configuration`
  - `microedition.profiles`
  - `microedition.pim.version`
  - `microedition.hostname`
  - ...
- `Runtime.getRuntime().totalMemory();`
- `Runtime.getRuntime().freeMemory();`

## Drawbacks

- Applications do not define the visual appearance (shape, color, font...) of the components
- Applications are not aware of the interactions (scrolling...)
- Applications cannot access concrete input devices like specific individual keys

## Conclusion

- User-friendly
- Avoid multiple classes
- Not used for professional applications
- Javadoc : <http://java.sun.com/javame/reference/apis/jsr118/>