

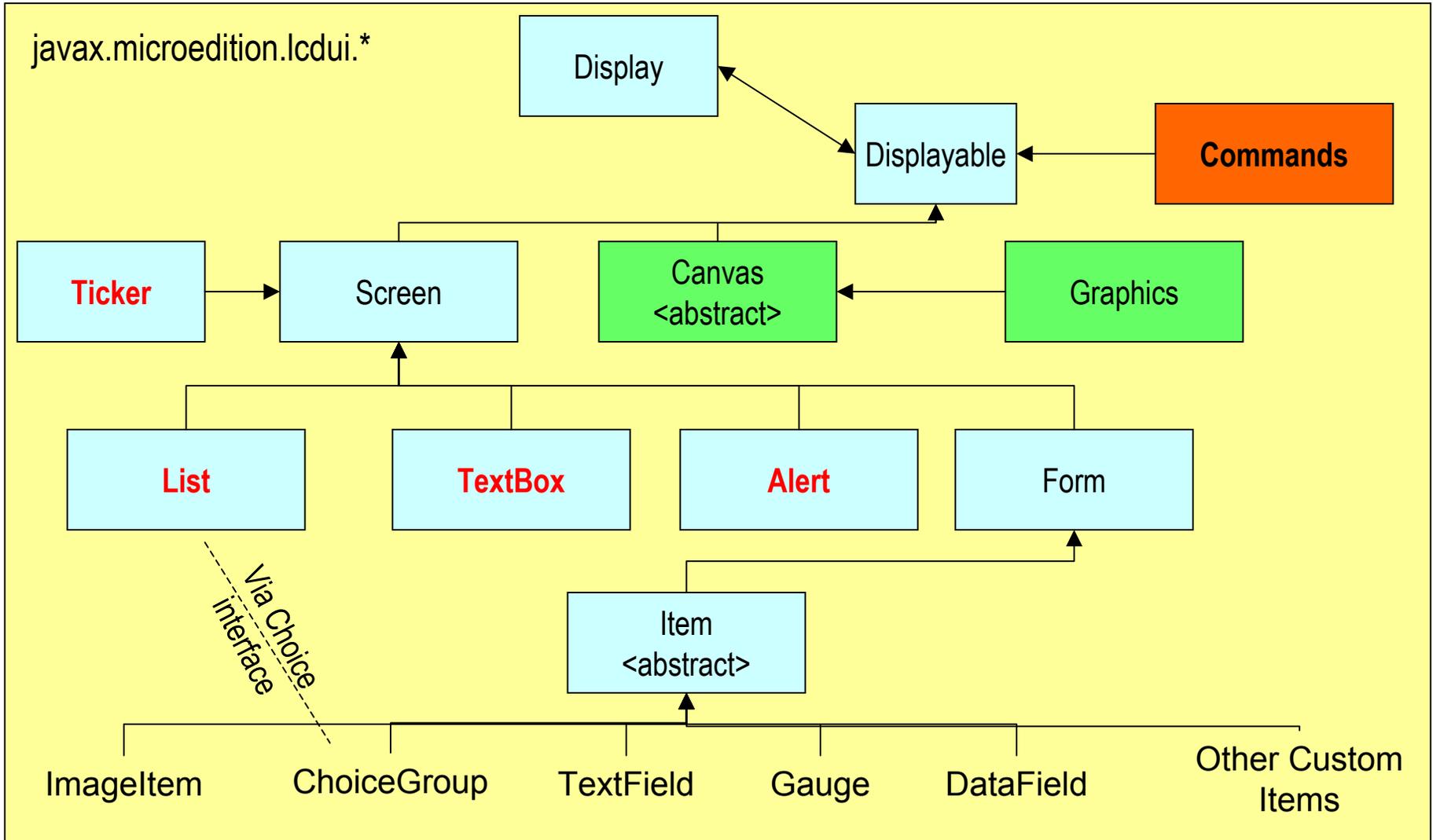


Mobile Information Device Programming (10)

Lecturer: Alireza Mousavi
School of Engineering & Design
www.brunel.ac.uk/~emstaam



Higher-Level UI





List

There are three types Lists:

- Multiple (check boxes)
- Exclusive (radio buttons)
- Implicit (simple list)



ChoiceGroup vs. List

see [[slide 2](#)]

- The only similarity between **ChoiceGroup** and **List** is in their appearance
- The event handling for each is different
- Reminder on **ChoiceGroup** event handling:
 - Add ChoiceGroup to the Form
 - Specify ItemStateListener
 - Listen to event and use ItemStateChanged() method
 - Add Commands to the Form containing the ChoiceGroup and use commandAction () method



List Event Handling

- 1. Exclusive and Multiple Lists:** You need to associate a command with your event. A selection of a choice does not necessarily trigger an event. For example "Save" or "View"
- 2. Implicit List:** With this type of List an event is triggered once an element has been selected. `commandAction()` method will be called. Example: to register a Listener for an implicit List:

```
lsMyList = new List("SMS Options:", Choice.IMPLICIT);  
lsMyList.setCommandListener(this);
```

- 3. Further reading on List API :** List Class: "javax.microedition.lcdui.List"



Example for Implicit List

- Create an implicit list using array of "Image" objects and "String" objects
- The first 3 members of the list are "Next", "Previous" and "New"
- In the same Midlet provide user with an *"Add"* command to be able to add new member to the list called "Delete"
- Exercise E 10.1



Result

```
Command Prompt - midp -classpath . ImplList
Next
Previous
New
New Task

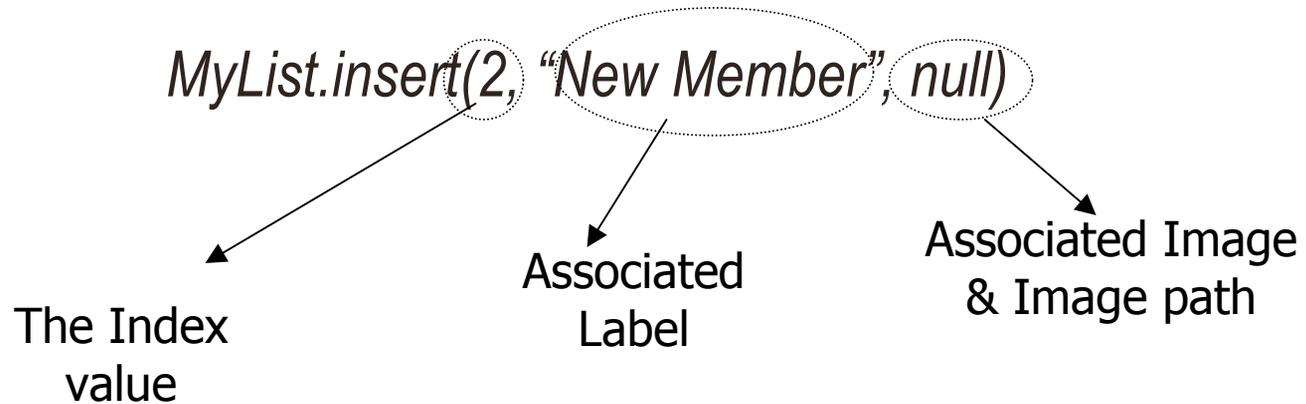
D:\midlets\Example8-1>javac -bootclasspath c:\j2me\midp2.0fcs\classes ImplList.java
D:\midlets\Example8-1>preverify -classpath c:\j2me\midp2.0fcs\classes;. -d . ImplList
D:\midlets\Example8-1>midp -classpath . ImplList
D:\midlets\Example8-1>javac -bootclasspath c:\j2me\midp2.0fcs\classes ImplList.java
D:\midlets\Example8-1>preverify -classpath c:\j2me\midp2.0fcs\classes;. -d . ImplList
D:\midlets\Example8-1>midp -classpath . ImplList
Next
Previous
New
New Task
```





Notice

- If you wish to insert a new member to an existing list you can choose the position:





Multiple Choice List

(Assignment)

- Create a Multiple Choice Application

```
IsMyList = new List("Choose Options:", List.MULTIPLE, alternatives, images);
```

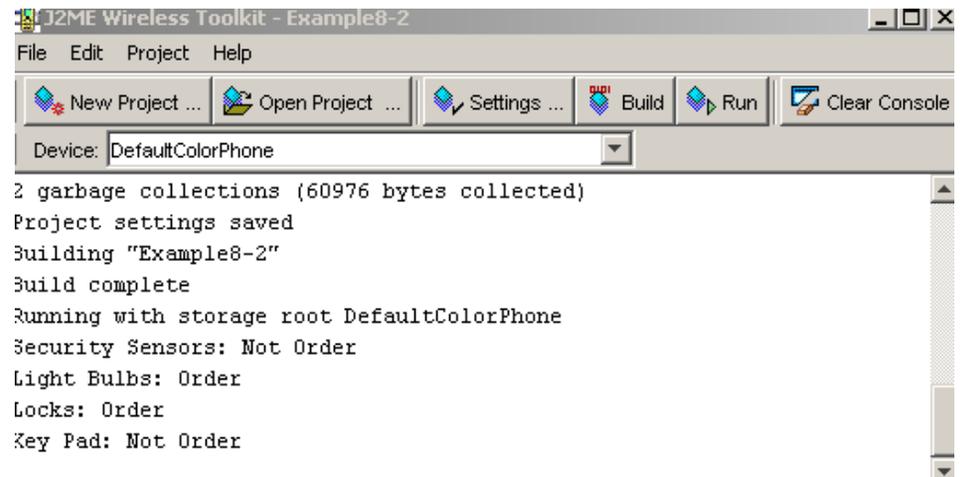
String Elements Image Elements

- Add Exit and View Command
- Once the view command is selected send a message to the consol which alternatives have been chosen

E:10-2



Result





TextBox

- **TextBox** is similar to **TextField** and functions as a multi-line text entry interface
- You can restrict the characters you would like to be held on a **TextBox**
- *getMaxSize()* method can show what is the maximum number of characters allowed in a **TextBox**



TextBox vs. TextField

- **TextField** is part of Form therefore other components can also be displayed at the same time. **This is not the case with TextBox. TextBox is the only component visible**
- **TextBox** events are handled through Command objects. There is no *ItemStateListener* like **TextField** (why?)



Example

Create a simple clipboard to mark, copy and paste characters in a Textbox.

- Example 10-3





Alert and Alert type

- **Alert** is a dialogue box containing image and text (optional)
- Command objects are not allowed on an **Alert**
- Implementation deals with user interaction
- You cannot add *TextBox* or *TextField* to prompt for information



Types of Alert

1. **Modal:** An **Alert** is on display until user dismisses it
2. **Time:** The **Alert** is on display for a given time (in milliseconds)



Declaration and Methods

```
Alert Ding;  
Ding = new Alert("Error", "Error message", null, null);  
Ding.setTimeout(Alert.FOREVER);  
...  
Display.setCurrent(Ding)
```

AlertType

AlertImage



Example Assignment

- Create an Alert with an Image
- Example 10-4.