

# Chapter 9A.

## Swing Components



2020-2021

COMP2396 Object-Oriented Programming and Java

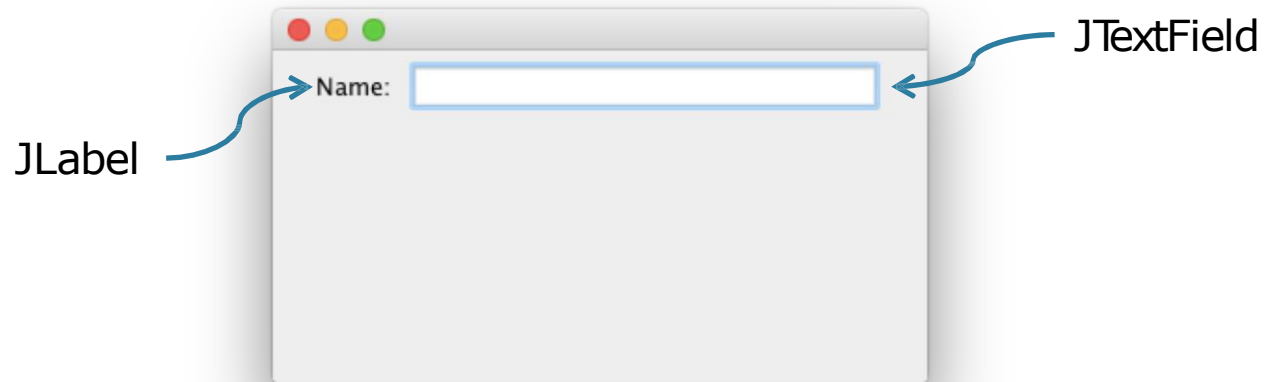
Dr. T.W. Chim (E-mail: [twchim@cs.hku.hk](mailto:twchim@cs.hku.hk))

Department of Computer Science, The University of Hong Kong

# Swing Components

- Some common Swing components
  - JTextField
  - JTextArea
  - JCheckBox
  - JList
  - JMenuBar, JMenu, JMenuItem

# JTextField



## — Constructors

```
JTextField field = new JTextField(20);
```

20 columns

```
JTextField field = new JTextField("Your name");
```

# TextField

- How to use it

- Get text out of it

```
System.out.println(field.getText());
```

- Put text in it

```
field.setText("whatever"); field.setText("");  
// this clears the field
```

- Get an ActionEvent when the user presses return or enter

```
field.addActionListener(myActionListener);
```

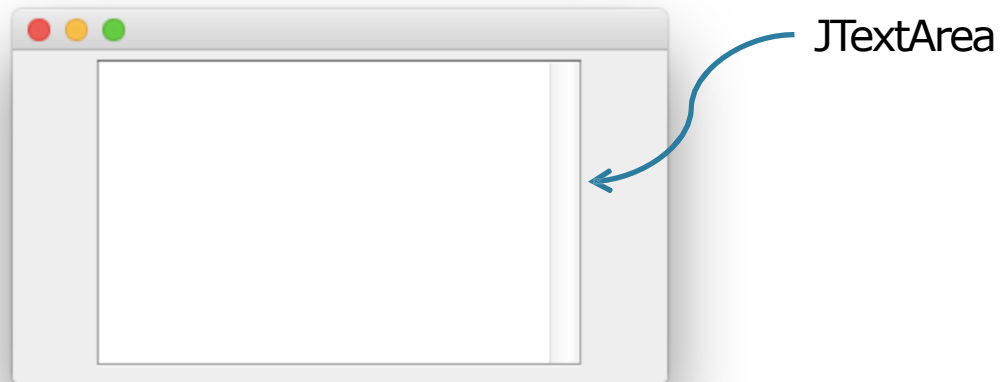
- Select/Highlight the text in the field

```
field.selectAll();
```

- Put the cursor back in the field

```
field.requestFocus();
```

# JTextArea



## — Constructor

```
JTextArea text = new JTextArea(10, 20);
```

10 lines and 20 columns

# JTextArea

- How to use it
  - Make it have a vertical scrollbar only

```
JScrollPane scroller = new JScrollPane(text);  
text.setLineWrap(true); // make text to multiple rows  
scroller.setVerticalScrollBarPolicy(  
    ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);  
scroller.setHorizontalScrollBarPolicy(  
    ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);  
panel.add(scroller); // add scroller, but not text
```

- Replace the text that is in it

```
text.setText("Not all who are lost are wandering");
```

# JTextArea

- How to use it

- Append to the text that is in it

```
text.append("button clicked");
```

- Select/Highlight the text in the field

```
text.selectAll();
```

- Put the cursor back in the field

```
text.requestFocus();
```

# JTextArea

## —Example

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class TextAreaEx implements ActionListener {
    JTextArea text;

    public static void main(String[] args) {
        TextAreaEx gui = new TextAreaEx();
        gui.go();
    }

    public void go() {
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



# JTextArea

## —Example

```
JPanel panel = new JPanel();
JButton button = new JButton("Just Click It");
button.addActionListener(this);
text = new JTextArea(10, 20);
text.setLineWrap(true); // go to a new line when your text reaches the right boundary
```

```
JScrollPane scroller = new JScrollPane(text);
scroller.setVerticalScrollBarPolicy(
    ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
scroller.setHorizontalScrollBarPolicy(
    ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
```

If you don't set the scroll bar policy, the default value will be `*_SCROLLBAR_AS_NEEDED`.

```
panel.add(scroller);
frame.add(panel, BorderLayout.CENTER);
frame.add(button, BorderLayout.SOUTH);
```

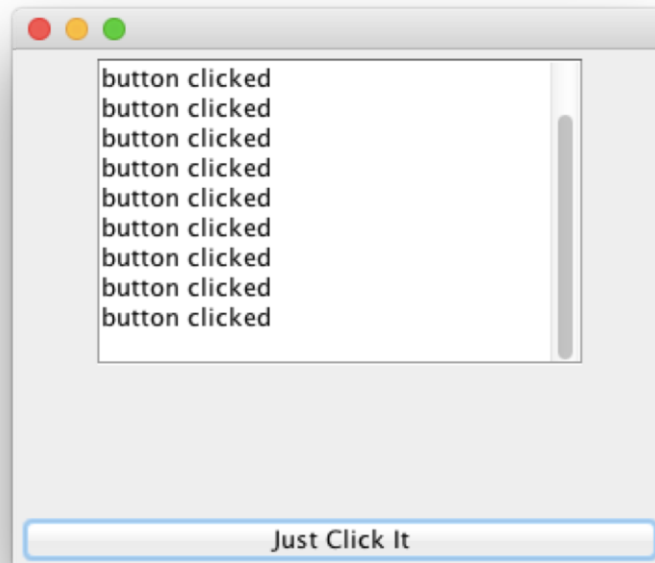
# JTextArea

## —Example

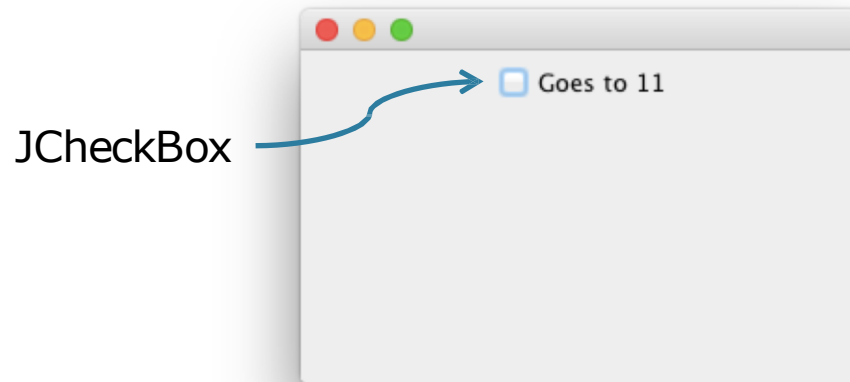
```
    frame.setSize(350, 300);  
    frame.setVisible(true);  
}  
  
public void actionPerformed(ActionEvent event) {  
    text.append("button clicked\n");  
}  
}
```

# JTextArea

## —Sample output



# JCheckBox



## — Constructor

```
JCheckBox check = new JCheckBox("Goes to 11");
```

# JCheckBox

- How to use it
  - Register a listener for the ItemEvent

```
check.addItemListener(this);
```

- Handle the ItemEvent

```
public void itemStateChanged(ItemEvent event) {  
    String onOrOff = "off";  
    if (check.isSelected()) onOrOff = "on";  
    System.out.println("Check box is " + onOrOff);  
}
```

- Select or deselect it in code

```
check.setSelected(true);  
check.setSelected(false);
```

# JList



## —Constructor

```
String[] entries = {"alpha", "beta", "gamma", "delta",  
                    "epsilon", "zeta", "eta", "theta"};  
JList<String> list = new JList<String>(entries);
```

JList constructor takes an array of any object type. They do not have to be Strings, but a string representation will appear in the list

# JList

- How to use it
  - Make it have a vertical scrollbar

```
JScrollPane scroller = new JScrollPane(list);  
scroller.setVerticalScrollBarPolicy(  
    ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);  
scroller.setHorizontalScrollBarPolicy(  
    ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);  
panel.add(scroller); // add scroller, but not text
```

- Set the number of lines to show before scrolling

```
list.setVisibleRowCount(4);
```

- Restrict the user to selecting only 1 thing at a time

```
list.setSelectionMode(  
    ListSelectionModel.SINGLE_SELECTION);
```

# JList

- How to use it
  - Register a listener for the ListSelectionEvent

```
list.addListSelectionListener(this);
```

- Handle the ListSelectionEvent

```
import javax.swing.event.*;
```

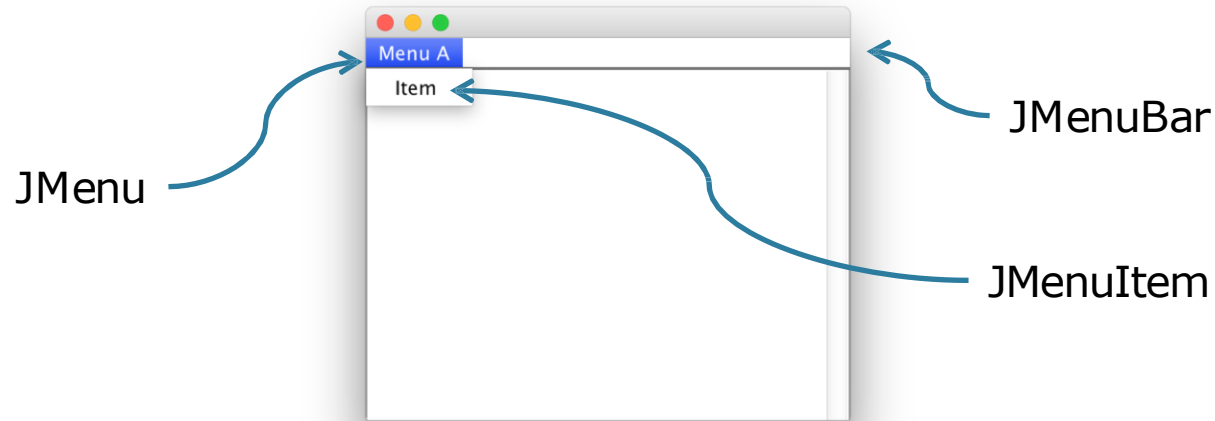
```
public void valueChanged(ListSelectionEvent event)
{
    if (!event.getValueIsAdjusting()) {
        String selection = (String) list.getSelectedValue();
        System.out.println(selection);
    }
}
```

getSelectedValue() actually  
returns an object

You will get the event twice if you  
do not put in this if test!



# JMenuBar, JMenu, JMenuItem



## —Constructors

```
JMenuBar menuBar = new JMenuBar();  
JMenu menu = new JMenu("Menu A");  
JMenuItem menuItem = new JMenuItem("Item");
```

# JMenuBar, JMenu, JMenuItem

- How to use them

- Add a menu item to a menu

```
menu.add(menuitem);
```

- Add a menu to a menu bar

```
menuBar.add(menu);
```

- Set a menu bar to a frame

```
frame.setJMenuBar(menuBar);
```

# JMenuBar, JMenu, JMenuItem

—How to use them

—Register a listener for the ActionEvent

```
menuItem.addActionListener(this);
```

—Handle the ActionEvent

```
public void actionPerformed(ActionEvent event) {  
    // codes for handling the selection of the menu item  
}
```

# JMenuBar, JMenu, JMenuItem

## —Example

```
import javax.swing.*;
import java.awt.event.*;

public class SimpleMenu implements ActionListener {
    JTextArea text;

    public static void main(String[] args) {
        SimpleMenu simpleMenu = new SimpleMenu();
        simpleMenu.go();
    }

    public void go() {
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

# JMenuBar, JMenu, JMenuItem

## —Example

```
JMenuBar menuBar = new JMenuBar();
JMenu menu = new JMenu("Menu A");
JMenuItem menuItem = new JMenuItem("Item");
menuItem.addActionListener(this);
menu.add(menuItem); menuBar.add(menu);
frame.setJMenuBar(menuBar);

text = new JTextArea();
text.setLineWrap(true);
JScrollPane scroller = new JScrollPane(text);
scroller.setVerticalScrollBarPolicy(
    ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
scroller.setHorizontalScrollBarPolicy(
    ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
```

# JMenuBar, JMenu, JMenuItem

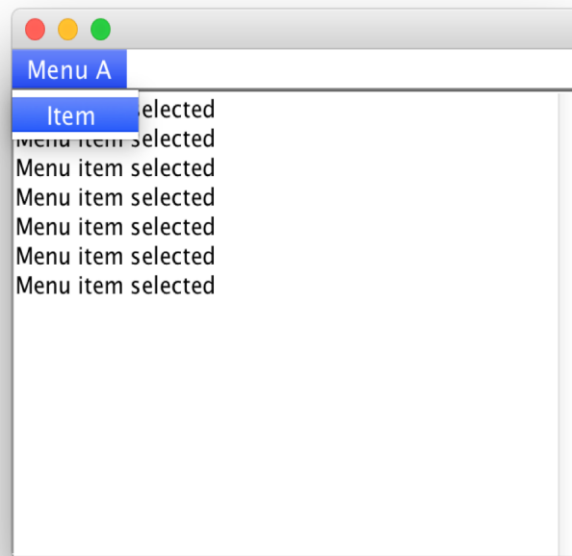
## —Example

```
frame.add(scroller);
frame.setSize(350, 300);
frame.setVisible(true);
}

public void actionPerformed(ActionEvent event) {
    text.append("Menu item selected\n");
}
}
```

# JMenuBar, JMenu, JMenuItem

## —Sample output



# Chapter 9A.

# End



2020-2021

COMP2396 Object-Oriented Programming and Java

Dr. T.W. Chim (E-mail: [twchim@cs.hku.hk](mailto:twchim@cs.hku.hk))

Department of Computer Science, The University of Hong Kong