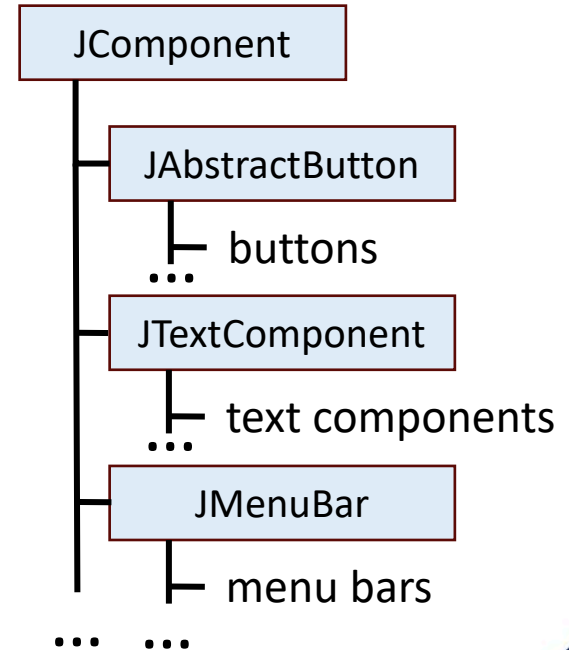


JC2002 Java Programming

Lecture 22: More JComponents

Selection of JComponents

- There is a large variety of different GUI components inherited from JComponent class for different purposes
 - In this session, we will only cover the most important classes
 - Different components have different methods for customising their appearance, adjusting their absolute size and position, setting and getting their internal state etc.



Using JLabel to display images

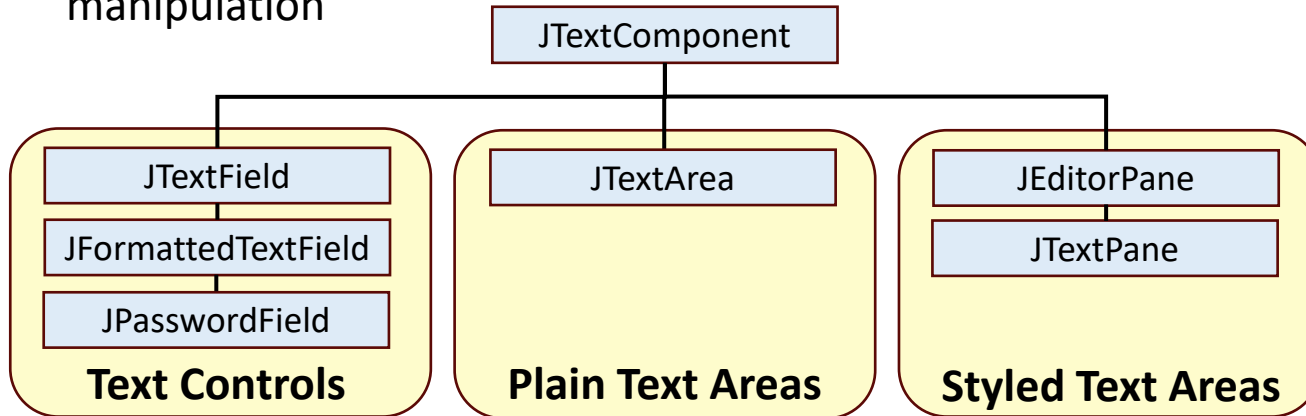
- JLabel is inherited directly from JComponent and it does not react to any user input
 - We have already used JLabel to show text, but it can be used also to display images by using its setIcon() method

```
1  import javax.swing.*;
2  class LabelExample{
3      public static void main(String args[]){
4          JFrame f = new JFrame("Label Example");
5          ImageIcon icon = new ImageIcon("image.jpg");
6          JLabel label = new JLabel();
7          label.setIcon(icon);
8          f.add(label);
9          f.setSize(200,150);
10         f.setVisible(true);
11     }
12 }
```



Text components

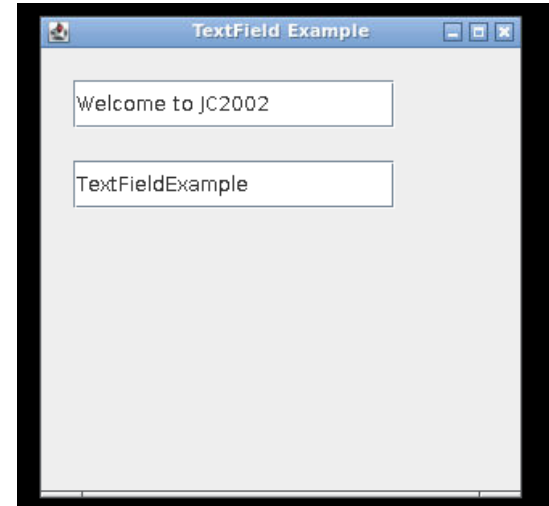
- Swing provides six text components, along with supporting classes and interfaces, for displaying and editing text
 - All the Swing text components inherit from `JTextComponent`, which provides a highly configurable and powerful foundation for text manipulation



Example of JTextField

- Use JTextField class to create a component that allows editing a single line of text

```
1  import javax.swing.*;  
2  class TextFieldExample{  
3      public static void main(String args[]){  
4          JFrame f= new JFrame("TextField Example");  
5          JTextField t1,t2;  
6          t1 = new JTextField("welcome to JC2002");  
7          t1.setBounds(20,20, 200,30);  
8          t2=new JTextField("TextFieldExample");  
9          t2.setBounds(20,70, 200,30);  
10         f.add(t1); f.add(t2);  
11         f.setSize(300,300);  
12         f.setLayout(null);  
13         f.setVisible(true);  
14     }  
15 }
```



Example of JTextArea

- Use JTextArea class to create a component that allows editing a multiple lines of text

```
1  import javax.swing.*;
2  class TextAreaExample{
3      public static void main(String args[]){
4          JFrame f= new JFrame("TextFieLd Example");
5          JTextArea t1;
6          t1 = new JTextField("welcome to \nJC2002!");
7          t1.setBounds(20,20, 200,100);
8          f.add(t1);
9          f.setSize(300,300);
10         f.setLayout(null);
11         f.setVisible(true);
12     }
13 }
```



Example of JTextPane (1)

- Use JTextPane class for editing or displaying *styled* text (even HTML)

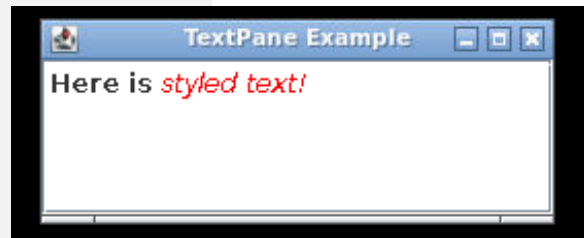
```
...
11 public class TextPaneExample {
12     public static void main(String args[]) throws BadLocationException {
13         JFrame frame = new JFrame("TextPane Example");
14         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15         Container cp = frame.getContentPane();
16         JTextPane pane = new JTextPane();
17         SimpleAttributeSet attributeSet = new SimpleAttributeSet();
18         StyleConstants.setBold(attributeSet, true);
19         pane.setCharacterAttributes(attributeSet, true);
20         pane.setText("Here is ");
21         attributeSet = new SimpleAttributeSet();
22         StyleConstants.setItalic(attributeSet, true);
23         StyleConstants.setForeground(attributeSet, Color.red);
24         Document doc = pane.getStyledDocument();
25         doc.insertString(doc.getLength(), "styled text!", attributeSet);
26         JScrollPane scrollPane = new JScrollPane(pane);
27         cp.add(scrollPane, BorderLayout.CENTER);
28         frame.setSize(400, 300);
29         frame.setVisible(true);
30     }
31 }
```

Imports not shown here

Different styles (colours, italic, bold, etc.) can be defined for the text

Example of JTextPane (2)

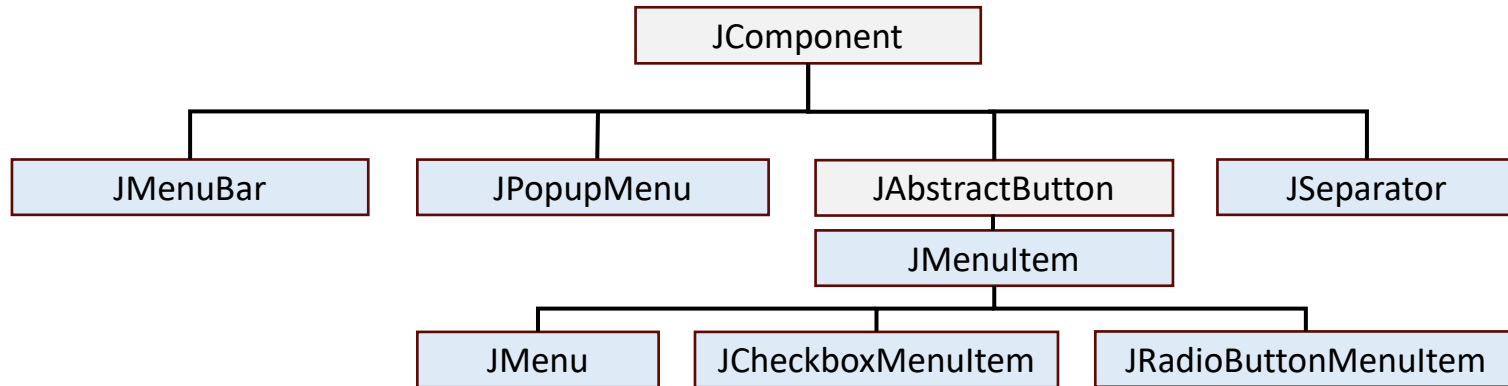
```
...
11 public class TextPaneExample {
12     public static void main(String args[]) throws BadLocationException {
13         JFrame frame = new JFrame("TextPane Example");
14         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15         Container cp = frame.getContentPane();
16         JTextPane pane = new JTextPane();
17         SimpleAttributes attributeSet = new SimpleAttributes();
18         StyleConstants.setBold(attributeSet, true);
19         pane.setCharacterAttributes(attributeSet, true);
20         pane.setText("Here is ");
21         attributeSet = new SimpleAttributes();
22         StyleConstants.setItalic(attributeSet, true);
23         StyleConstants.setForeground(attributeSet, Color.red);
24         Document doc = pane.getStyledDocument();
25         doc.insertString(doc.getLength(), "styled text!", attributeSet);
26         JScrollPane scrollPane = new JScrollPane(pane);
27         cp.add(scrollPane, BorderLayout.CENTER);
28         frame.setSize(200, 100);
29         frame.setVisible(true);
30     }
31 }
```



- For more information, see:
<https://docs.oracle.com/javase/tutorial/uiswing/components/editorpane.html>

Menu components

- There are several menu components providing many options for implementing menus in a small space
- Menu items (JMenuItem) are different kinds of special buttons showing labels



Example of using menu components (1)

- Example of creating a menu bar with a submenu

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```

Example of using menu components (2)

- Instantiate menu bar, menu, and submenu

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```

Example of using menu components (3)

- Create five menu item objects of class JMenuItem

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and JMenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```

Example of using menu components (4)

- Add the first three menu items to the main menu

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```

Example of using menu components (5)

- Add the last two menu items to the submenu

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```

Example of using menu components (6)

- Add submenu to the main menu, and main menu to the menu bar

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```

Example of using menu components (7)

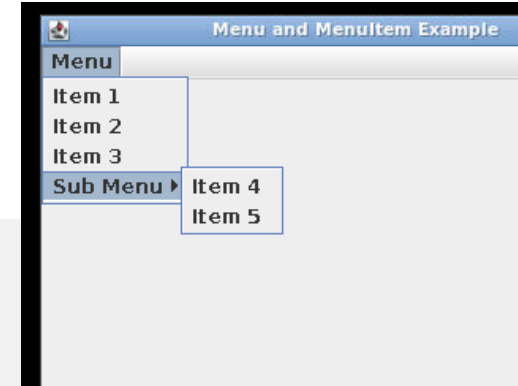
- Add menu bar to the frame using method setJMenuBar()

```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```


Example of using menu components (8)

- Run the program

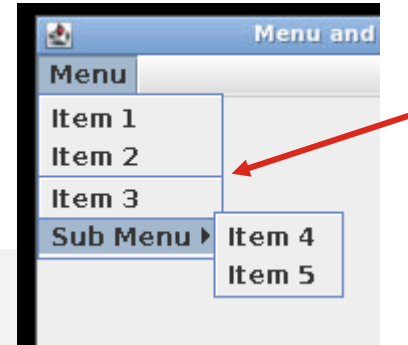
```
1  import javax.swing.*;
2  class MenuExample {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      MenuExample(){
6          JFrame f = new JFrame("Menu and MenuItem Example");
7          JMenuBar mb=new JMenuBar();
8          menu=new JMenu("Menu");
9          submenu=new JMenu("Sub Menu");
10         i1=new JMenuItem("Item 1");
11         i2=new JMenuItem("Item 2");
12         i3=new JMenuItem("Item 3");
13         i4=new JMenuItem("Item 4");
14         i5=new JMenuItem("Item 5");
15         menu.add(i1);
16         menu.add(i2);
17         menu.add(i3);
18         submenu.add(i4);
19         submenu.add(i5);
20         menu.add(submenu);
21         mb.add(menu);
22         f.setJMenuBar(mb);
23         f.setSize(400,400);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public static void main(String args[]){
28         new MenuExample();
29     }
30 }
```



Example of using JSeparator

- Use JSeparator instance to separate groups of menu items

```
1  import javax.swing.*;
2  class MenuExample2 {
3      JMenu menu, submenu;
4      JMenuItem i1, i2, i3, i4, i5;
5      JSeparator s;
6      MenuExample(){
7          JFrame f = new JFrame("Menu and MenuItem Example");
8          JMenuBar mb=new JMenuBar();
9          menu=new JMenu("Menu");
10         submenu=new JMenu("Sub Menu");
11         i1=new JMenuItem("Item 1");
12         i2=new JMenuItem("Item 2");
13         i3=new JMenuItem("Item 3");
14         i4=new JMenuItem("Item 4");
15         i5=new JMenuItem("Item 5");
16         s=new JSeparator();
17         menu.add(i1);
18         menu.add(i2);
19         menu.add(s);
20         menu.add(i3);
21         submenu.add(i4);
22         submenu.add(i5);
23         menu.add(submenu);
24         mb.add(menu);
25         f.setJMenuBar(mb);
26         f.setSize(400,400);
27         f.setLayout(null);
28         f.setVisible(true);
29     }
30     public static void main(String args[]){
31         new MenuExample();
32     }
33 }
```



Summary

- A graphical user interface provides a user-friendly mechanism to interact with an app that is easy to learn.
 - GUIs are usually based on similar logic, but different look-and-feel in different platforms.
- In Java, Swing is a commonly used library for implementing GUI
 - In Swing, different GUI components are implemented using classes inherited from `JComponent` class.
 - Swing allows different layouts for organising the GUI components.
 - Some of the most typical GUI components were introduced, including buttons, text fields, and menu items.

Questions, comments?