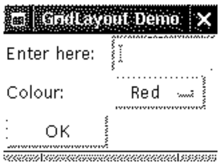
Layout Managers

The AWT comes with these layout managers:

- `FlowLayout`: arranges components in a row or several rows
- `GridLayout`: arranges components in a matrix/grid
- `CardLayout`: each component is a card, one shown at a time
- `BorderLayout`: arranges five components: north, south, east, west, center
- `GridBagLayout`: flexible, sophisticated grid

GridLayout Example

```
import java.awt.*;

public class GridDemo
{
  public static void main(String[] args) {
    Frame frame = new Frame("GridLayout Demo");
    frame.setBackground(Color.white);
    frame.setLayout(new GridLayout(3, 2));
    frame.add(new Label("Enter here:"));
    frame.add(new TextField(5));
    frame.add(new Label("Colour:"));
    Choice choice = new Choice();
    choice.add("Red");
    choice.add("Green");
    choice.add("Blue");
    frame.add(choice);
    frame.add(new Button("OK"));
    frame.pack();  // auto-size
    frame.show();
  }
}
```



CardLayout Example

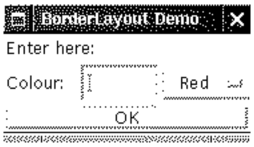
```
import java.awt.*;

public class CardDemo
{
  public static void main(String[] args) {
    Frame frame =
      new Frame("CardLayout Demo");
    frame.setBackground(Color.white);
    CardLayout c;
    frame.setLayout(c = new CardLayout());
    Panel panel;

    // first card
    frame.add(panel = new Panel(), "text");
    panel.setLayout(new GridLayout(2, 2));
    panel.add(new Label("Enter here:"));
    panel.add(new TextField(5));
    Button b = new Button("Prev");
    b.setEnabled(false);
    panel.add(b);
    panel.add(new Button("Next"));
```


CardLayout Example (cont.)

```
    // second card
    frame.add(panel = new Panel(), "colour");
    panel.setLayout(new GridLayout(2, 2));
    panel.add(new Label("Colour:"));
    Choice choice = new Choice();
    choice.add("Red");
    choice.add("Green");
    choice.add("Blue");
    panel.add(choice);
    panel.add(new Button("Prev"));
    panel.add(new Button("OK"));

    frame.pack();  // auto-size
    if (args.length > 0) c.next(frame);
    frame.show();
  }
}
```

BorderLayout Example

```
import java.awt.*;

public class BorderDemo
{
  public static void main(String[] args) {
    Frame frame = new Frame("BorderLayout Demo");
    frame.setBackground(Color.white);
    frame.add(new Label("Enter here:"), "North");
    frame.add(new TextField(5), "Center");
    frame.add(new Label("Colour:"), "West");
    Choice choice = new Choice();
    choice.add("Red");
    choice.add("Green");
    choice.add("Blue");
    frame.add(choice, "East");
    frame.add(new Button("OK"), "South");

    frame.pack();  // auto-size
    frame.show();
  }
}
```



GridBagLayout Example

```
import java.awt.*;

public class GridBagDemo
{
  public static void main(String[] args) {
    Frame frame = new Frame("GridBagLayout Demo");
    frame.setBackground(Color.white);
    frame.setLayout(new GridBagLayout());

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.weightx = gbc.weighty = 1.0;
    gbc.fill = GridBagConstraints.BOTH;

    // row 1
    frame.add(new Label("Enter here:"), gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    frame.add(new TextField(5), gbc);
    // row 2
    gbc.gridwidth = 1;
    frame.add(new Label("Colour:"), gbc);
    Choice choice = new Choice();
    choice.add("Red");
    choice.add("Green");
    choice.add("Blue");
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    frame.add(choice, gbc);
    // row 3
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    frame.add(new Button("OK"), gbc);
```


GridBagLayout Example (cont.)

```
    frame.pack();  // auto-size
    frame.show();
  }
}
```

Events

(The `java.awt.event.*` package)

What happens when user gives input to a component?

Events are generated by the component.

Event examples:

- `MouseEvent`: mouse clicks on a component
- `KeyEvent`: key clicks on a component
- `ActionEvent`: pushing a button, hitting enter in a textfield, selecting a menu item, etc.
- `WindowEvent`: window is opened, closed, etc.

Listeners

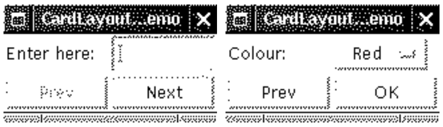
Each component maintains a list of *listeners* for each event type. Example:

1. `Button` maintains a list of `ActionListener` objects.
2. `addActionListener` and `removeActionListener` changes this list.
3. When an `ActionEvent` is generated, it is dispatched to all listeners on this list by calling their respective `actionPerformed` methods. (Like callback functions.)

So, you implement an `ActionListener` and add it to the list.

Similar for other event types.

Event Example




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

public class CardDemo2
{
  static CardLayout c;
  static Frame frame;

  static class PrevNext implements ActionListener {
    public void actionPerformed(ActionEvent e) {
      if ("Next".equals(e.getActionCommand()))
        c.next(frame);
      else
        c.previous(frame);
    }
  }

  public static void main(String[] args) {
    frame = new Frame("CardLayout Demo");
    frame.setBackground(Color.white);
    Button b;
    frame.setLayout(c = new CardLayout());
    Panel panel;
    PrevNext pn = new PrevNext();
```


Event Example (cont.)

```
    // first card
    frame.add(panel = new Panel(), "text");
    panel.setLayout(new GridLayout(2, 2));
    panel.add(new Label("Enter here:"));
    panel.add(new TextField(5));
    panel.add(b = new Button("Prev"));
    b.setEnabled(false);
    panel.add(b = new Button("Next"));
    b.addActionListener(pn);

    // second card
    frame.add(panel = new Panel(), "colour");
    panel.setLayout(new GridLayout(2, 2));
    panel.add(new Label("Colour:"));
    Choice choice = new Choice();
    choice.add("Red");
    choice.add("Green");
    choice.add("Blue");
    panel.add(choice);
    panel.add(b = new Button("Prev"));
    b.addActionListener(pn);
    panel.add(b = new Button("OK"));
    b.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        frame.dispose();
        System.exit(0);
      }
    });
```


Event Example (cont.)

```
    frame.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        frame.dispose();
        System.exit(0);
      }
    });
    frame.pack();  // auto-size
    if (args.length > 0) c.next(frame);
    frame.show();
  }
}
```