Graphical User Interface using Swing
Part I - Basics

Samir Genaim

http://docs.oracle.com/javase/tutorial/uiswing/
http://www.macs.hw.ac.uk/guidebook

What's a GUI?

What's a GUI?

- A Graphical User Interface, is the interface between the user and the application
- Presents actions available to the user, usually performed by manipulating the graphical components (e.g. pushing a button)

What's a GUI?

- A Graphical User Interface, is the interface between the user and the application
- Presents actions available to the user, usually performed by manipulating the graphical components (e.g. pushing a button)
- Two ways for Programming GUIs:
  1. **The easy way**: use a GUI to build GUIs
  2. **The hard way**: write code to create components, connect them with listeners, etc.

What's a GUI?

- A Graphical User Interface, is the interface between the user and the application
- Presents actions available to the user, usually performed by manipulating the graphical components (e.g. pushing a button)
- Two ways for Programming GUIs:
  1. **The easy way**: use a GUI to build GUIs
  2. **The hard way**: write code to create components, connect them with listeners, etc.
- We will do it **the hard way**!! You'll learn much more!

Java GUI Libraries

- The Abstract Window Toolkit (**AWT**)
  1. Java Core in version 1.0, way back in the 1990s
  2. Uses native widgets to display GUI components
  3. Looks different on different platforms
  4. The least common denominator limitation: only widgets that exist on all platforms are supported
- **Swing** (Java Foundation Classes -- JFC)
  5. Started by Sun and Netscape in 1997
  6. Implements its own widgets - independent from OS
  7. Different skins, looks the same on all platforms (?)
- **SWT, SwingX, JGoodies, JavaFX, Apache Pivot, Qt Jambi, ...**

What we Will Learn

- The basics of Swing, there is much more that you can find by yourself!!
- First we will see Swing by Example, covering the different Swing components, layouts, etc.
- How to design GUI using MVC design pattern
- Later, when you're confident with Swing, we will talk more about how the Swing classes are organised, etc. This will give you an idea how to program your own components in Swing.
- When discussing Threads in Java we will come back to discuss Thread safety issues in Swing

Getting Started with Swing

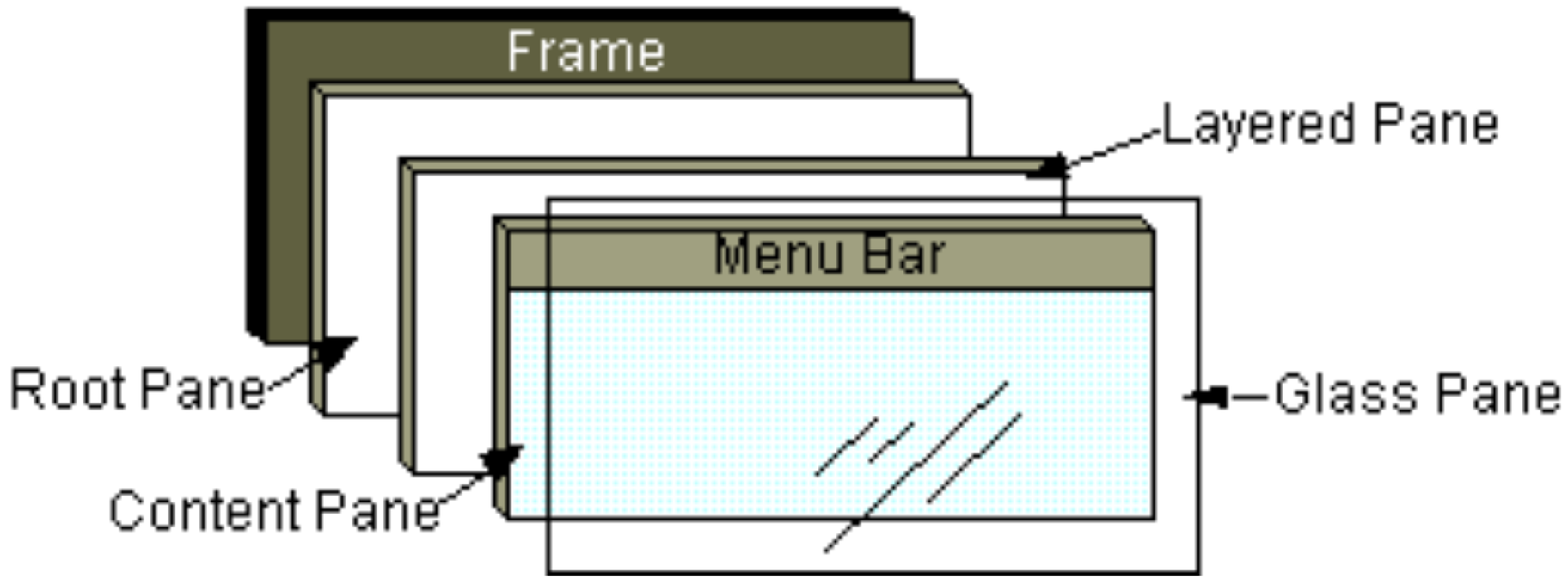
- Every GUI in the world starts with a main window, in which things are displayed.
- There are three main types of windows in Swing

  1. Frame
  2. Dialog
  3. Applet

- A Frame object in Swing is called a **JFrame**, which follows the naming conventions of Swing where default widgets have a **J** at the start (i.e **JButton**, **JLabel**, etc).

Frames in Swing: JFrame

The class for creating a frame in Swing is called **JFrame.** It represents a frame like this:



Content Pane is where we place widgets, the use of the other panes is not important for now.

Your First JFrame!

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

see: example.swing.misc.FrameExample

Your First JFrame!

import the swing classes

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

see: example.swing.misc.FrameExample

Your First JFrame!

import the swing classes

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

create the JFrame

see: example.swing.misc.FrameExample

Your First JFrame!

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

import the swing classes

create the JFrame

What to do when closing the window. If we don't use EXIT Swing will keep running in the background

see: example.swing.misc.FrameExample

Your First JFrame!

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

import the swing classes

create the JFrame

What to do when closing the window. If we don't use EXIT Swing will keep running in the background

set the size of the frame (pixels)

see: example.swing.misc.FrameExample

Your First JFrame!

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

import the swing classes

create the JFrame

What to do when closing the window. If we don't use EXIT Swing will keep running in the background

set the size of the frame (pixels)

make it visible

see: example.swing.misc.FrameExample

Your First JFrame!

```
import javax.swing.*;

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

import the swing classes

schedule createAndShowGUI to be invoked by the Thread of Swing

create the JFrame

What to do when closing the window. If we don't use EXIT Swing will keep running in the background

set the size of the frame (pixels)

make it visible

see: example.swing.misc.FrameExample

It's common to extend JFrame

```
import javax.swing.*;

public class FrameExample_1 extends JFrame {

    public FrameExample_1() {
        super("[=] Hello World [=]");
        initGUI();
    }

    private void initGUI() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 100);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FrameExample_1();
            }
        });
    }
}
```

see: example.swing.misc.FrameExample_1

It's common to extend JFrame

```
import javax.swing.*;

public class FrameExample_1 extends JFrame {

    public FrameExample_1() {
        super("[=] Hello World [=]");
        initGUI();
    }

    private void initGUI() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 100);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FrameExample_1();
            }
        });
    }
}
```

extend the JFrame class

see: example.swing.misc.FrameExample_1

It's common to extend JFrame

```
import javax.swing.*;

public class FrameExample_1 extends JFrame {

    public FrameExample_1() {
        super("[=] Hello World [=]");
        initGUI();
    }

    private void initGUI() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 100);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FrameExample_1();
            }
        });
    }
}
```

extend the JFrame class

in the constructor call the constructor of JFrame, and call your method to initialize the GUI

see: example.swing.misc.FrameExample_1

It's common to extend JFrame

```
import javax.swing.*;

public class FrameExample_1 extends JFrame {

    public FrameExample_1() {
        super("[=] Hello World [=]");
        initGUI();
    }

    private void initGUI() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 100);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FrameExample_1();
            }
        });
    }
}
```

extend the JFrame class

in the constructor call the constructor of JFrame, and call your method to initialize the GUI

It is a good idea to have a method called initGUI for initializing the GUI

see: example.swing.misc.FrameExample_1

It's common to extend JFrame

```
import javax.swing.*;

public class FrameExample_1 extends JFrame {

    public FrameExample_1() {
        super("[=] Hello World [=]");
        initGUI();
    }

    private void initGUI() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 100);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FrameExample_1();
            }
        });
    }
}
```

extend the JFrame class

in the constructor call the constructor of JFrame, and call your method to initialize the GUI

It is a good idea to have a method called initGUI for initializing the GUI

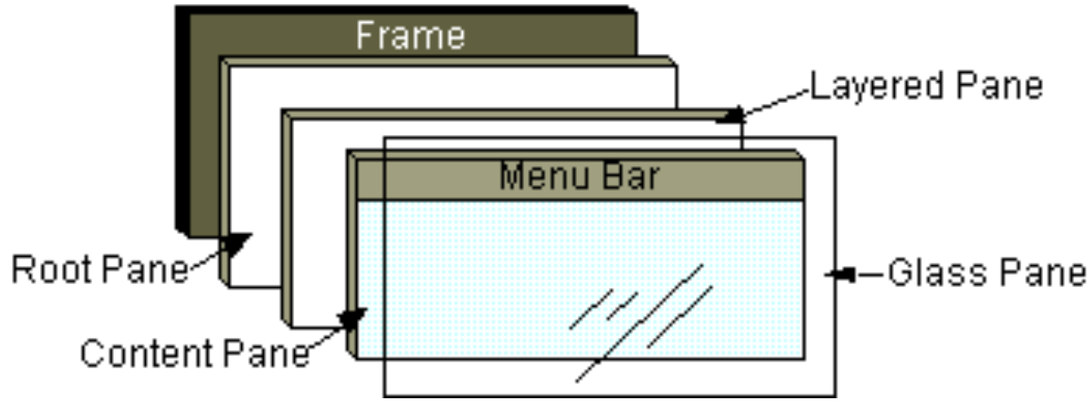
schedule a job to create an instance of your class

see: example.swing.misc.FrameExample_1

Use JPanel to Position Your Widgets

- Imagine your GUI as a Tree:

  if *A* is placed on B then
  A is a child of B



- We can position widgets relative to the top-left corner (0,0) of parent

- Grouping widgets is fundamental

  1. make operations on sets of components
  2. move them together from one place to another
  3. define common properties

- The **JPanel** component allows grouping widgets

Use JPanel to Position all Widgets

```
public class PanelExample extends JFrame {

    public PanelExample() {
        super("[=] There's a JPanel in here! [=]");
        initGUI();
    }

    private void initGUI() {
        JPanel mainPanel = new JPanel();
        mainPanel.setOpaque(true);

        this.setContentPane(mainPanel);

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
    }

    ...
}
```

see: example.swing.misc.PanelExample

Use JPanel to Position all Widgets

```
public class PanelExample extends JFrame {

    public PanelExample() {
        super("[=] There's a JPanel in here! [=]");
        initGUI();
    }

    private void initGUI() {
        JPanel mainPanel = new JPanel();
        mainPanel.setOpaque(true);

        this.setContentPane(mainPanel);

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
    }

    ...
}
```

create an instance of JPanel

see: example.swing.misc.PanelExample

Use JPanel to Position all Widgets

```
public class PanelExample extends JFrame {

    public PanelExample() {
        super("[=] There's a JPanel in here! [=]");
        initGUI();
    }

    private void initGUI() {
        JPanel mainPanel = new JPanel();
        mainPanel.setOpaque(true);

        this.setContentPane(mainPanel);

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
    }

    ...
}
```

create an instance of JPanel

make it opaque (nontransparent)
Not always needed

see: example.swing.misc.PanelExample

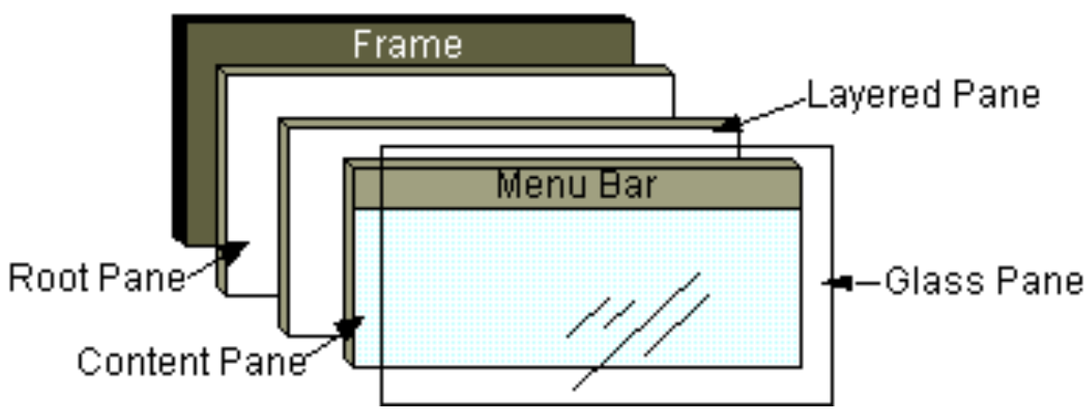
Use JPanel to Position all Widgets

```
public class PanelExample extends JFrame {

    public PanelExample() {
        super("[=] There's a JPanel in here! [=]");
        initGUI();
    }

    private void initGUI() {
        JPanel mainPanel = new JPanel();
        mainPanel.setOpaque(true);

        this.setContentPane(mainPanel);

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
    }

    ...
}
```

create an instance of JPanel

make it opaque (nontransparent)
Not always needed

place it in the Content Pane of the JFrame



see: example.swing.misc.PanelExample

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

create another JPanel

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

create another JPanel

set its color, position, and size

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

create another JPanel

set its color, position, and size

add it to the mainPanel

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

create another JPanel

set its color, position, and size

add it to the mainPanel

another blue panel

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

create another JPanel

set its color, position, and size

add it to the mainPanel

another blue panel

place mainPanel in the Content Pane of the JFrame

see: example.swing.misc.PanelExample_1

Nested JPanels

```
private void initGUI() {

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JPanel redPanel = new JPanel();
        redPanel.setBackground(Color.red);
        redPanel.setLocation(60, 10);
        redPanel.setSize(50, 50);
        mainPanel.add(redPanel);

        JPanel bluePanel = new JPanel();
        bluePanel.setBackground(Color.blue);
        bluePanel.setLocation(200, 10);
        bluePanel.setSize(50, 50);
        mainPanel.add(bluePanel);

        this.setContentPane(mainPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(290, 100);
        this.setVisible(true);
}
```

create an instance of JPanel

it uses absolute positioning to layout its widgets

create another JPanel

set its color, position, and size

add it to the mainPanel

another blue panel

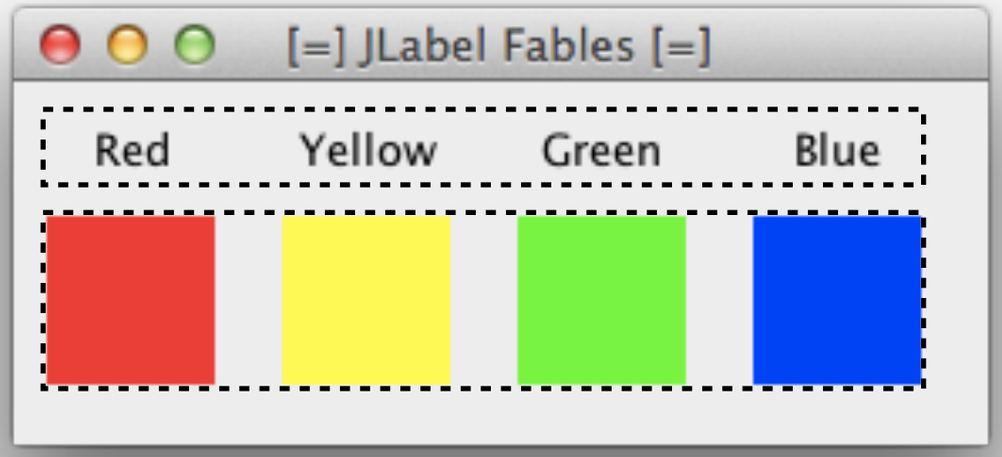
place mainPanel in the Content Pane of the JFrame

each component can be made visible or non-visible using setVisible

see: example.swing.misc.PanelExample_1

The JLabel Widget

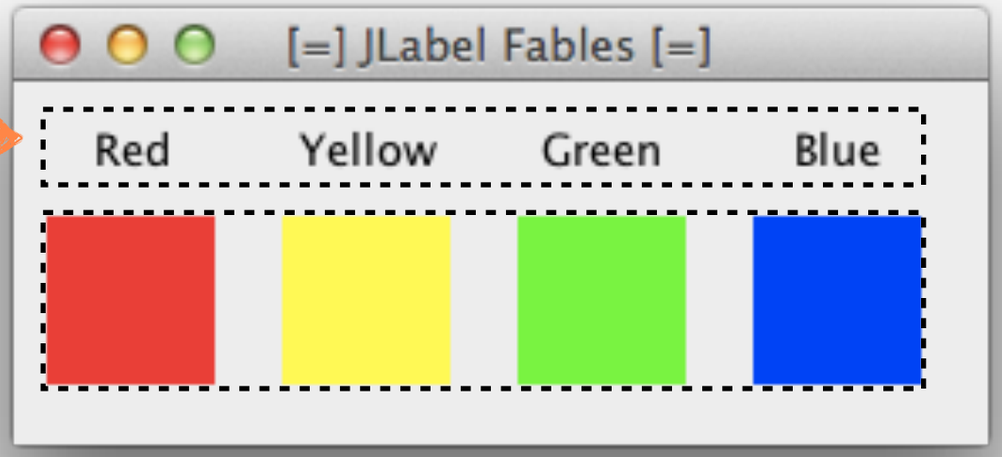
```
JPanel textPanel = new JPanel();
textPanel.setLayout(null);
textPanel.setLocation(10, 0);
textPanel.setSize(260, 30);
mainPanel.add(textPanel);

JLabel redLabel = new JLabel("Red");
redLabel.setLocation(0, 0);
redLabel.setSize(50, 40);
redLabel.setHorizontalAlignment(JLabel.CENTER);
textPanel.add(redLabel);
...
JPanel panelForPanels = new JPanel();
panelForPanels.setLayout(null);
panelForPanels.setLocation(10, 40);
panelForPanels.setSize(260, 50);
mainPanel.add(panelForPanels);

JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(0, 0);
redPanel.setSize(50, 50);
panelForPanels.add(redPanel);
...
```



see: example.swing.misc.LabelExample

The JLabel Widget

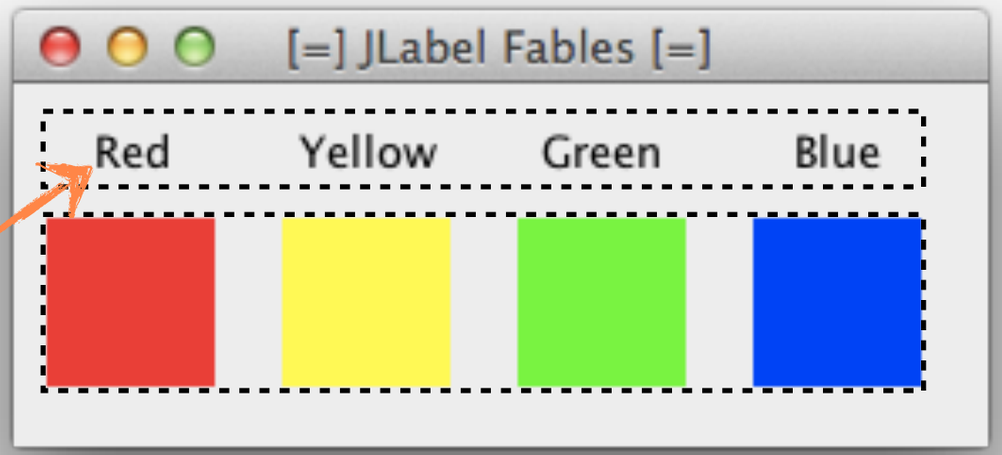
```
JPanel textPanel = new JPanel();
textPanel.setLayout(null);
textPanel.setLocation(10, 0);
textPanel.setSize(260, 30);
mainPanel.add(textPanel);

JLabel redLabel = new JLabel("Red");
redLabel.setLocation(0, 0);
redLabel.setSize(50, 40);
redLabel.setHorizontalAlignment(JLabel.CENTER);
textPanel.add(redLabel);
...
JPanel panelForPanels = new JPanel();
panelForPanels.setLayout(null);
panelForPanels.setLocation(10, 40);
panelForPanels.setSize(260, 50);
mainPanel.add(panelForPanels);

JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(0, 0);
redPanel.setSize(50, 50);
panelForPanels.add(redPanel);
...
```



see: example.swing.misc.LabelExample

The JLabel Widget

```
JPanel textPanel = new JPanel();
textPanel.setLayout(null);
textPanel.setLocation(10, 0);
textPanel.setSize(260, 30);
mainPanel.add(textPanel);

JLabel redLabel = new JLabel("Red");
redLabel.setLocation(0, 0);
redLabel.setSize(50, 40);
redLabel.setHorizontalAlignment(JLabel.CENTER);
textPanel.add(redLabel);
...
```

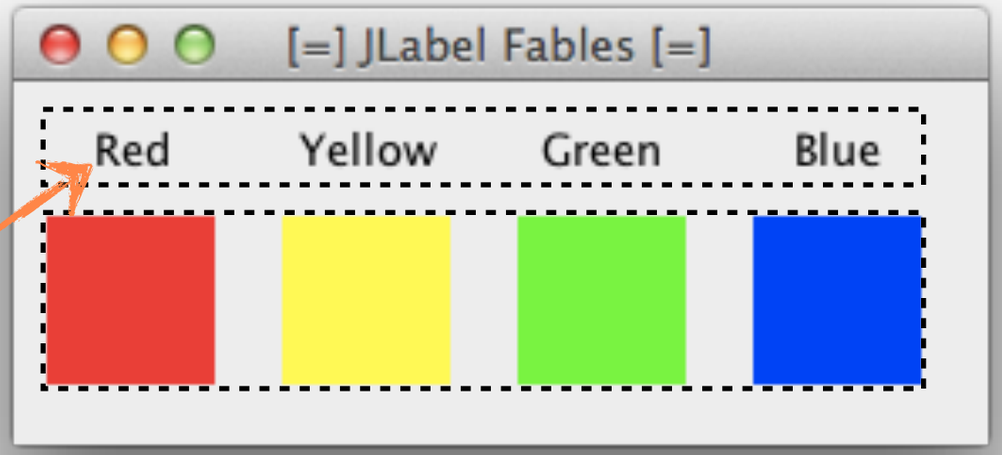



```
JPanel panelForPanels = new JPanel();
panelForPanels.setLayout(null);
panelForPanels.setLocation(10, 40);
panelForPanels.setSize(260, 50);
mainPanel.add(panelForPanels);

JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(0, 0);
redPanel.setSize(50, 50);
panelForPanels.add(redPanel);
...
```

see: example.swing.misc.LabelExample

The JLabel Widget

```
JPanel textPanel = new JPanel();
textPanel.setLayout(null);
textPanel.setLocation(10, 0);
textPanel.setSize(260, 30);
mainPanel.add(textPanel);

JLabel redLabel = new JLabel("Red");
redLabel.setLocation(0, 0);
redLabel.setSize(50, 40);
redLabel.setHorizontalAlignment(JLabel.CENTER);
textPanel.add(redLabel);
...
JPanel panelForPanels = new JPanel();
panelForPanels.setLayout(null);
panelForPanels.setLocation(10, 40);
panelForPanels.setSize(260, 50);
mainPanel.add(panelForPanels);

JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(0, 0);
redPanel.setSize(50, 50);
panelForPanels.add(redPanel);
...
```



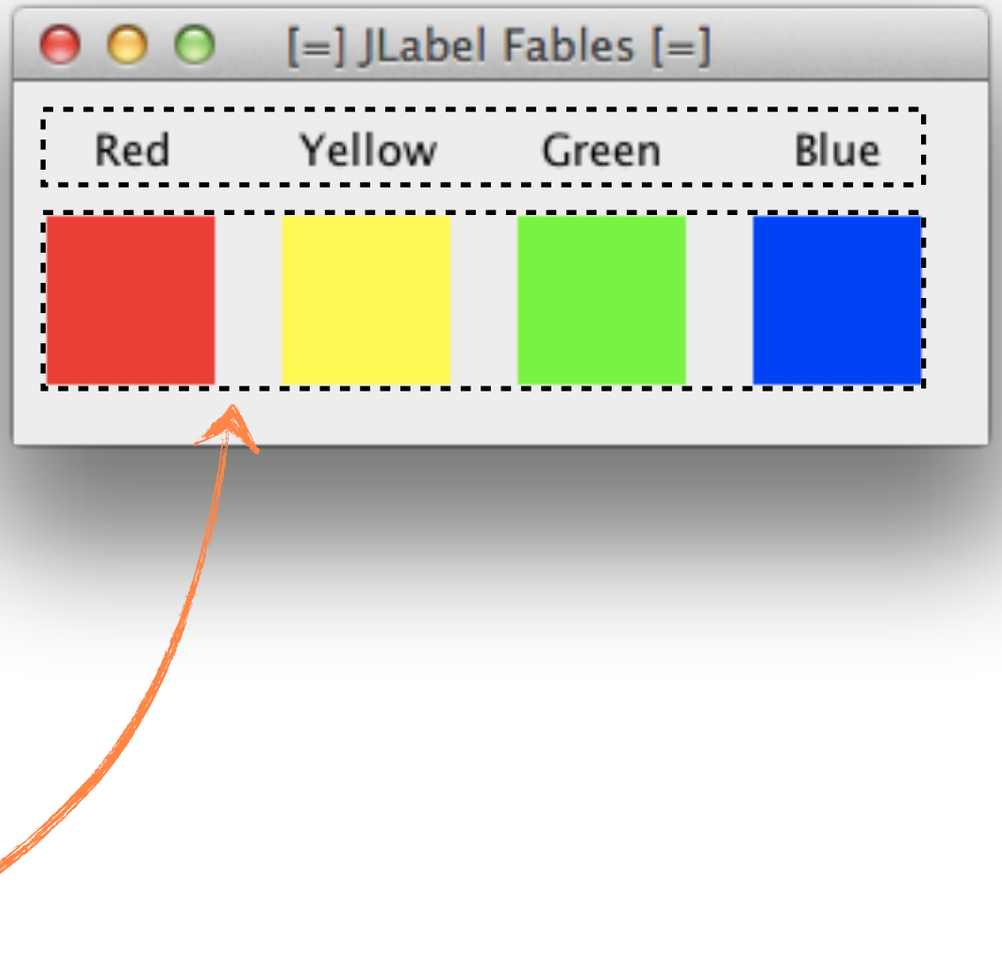
How to h-align the text inside the area 50x40 of the panel

JLabel.RIGHT
JLabel.LEFT
JLabel.CENTER
...

see: example.swing.misc.LabelExample

The JLabel Widget

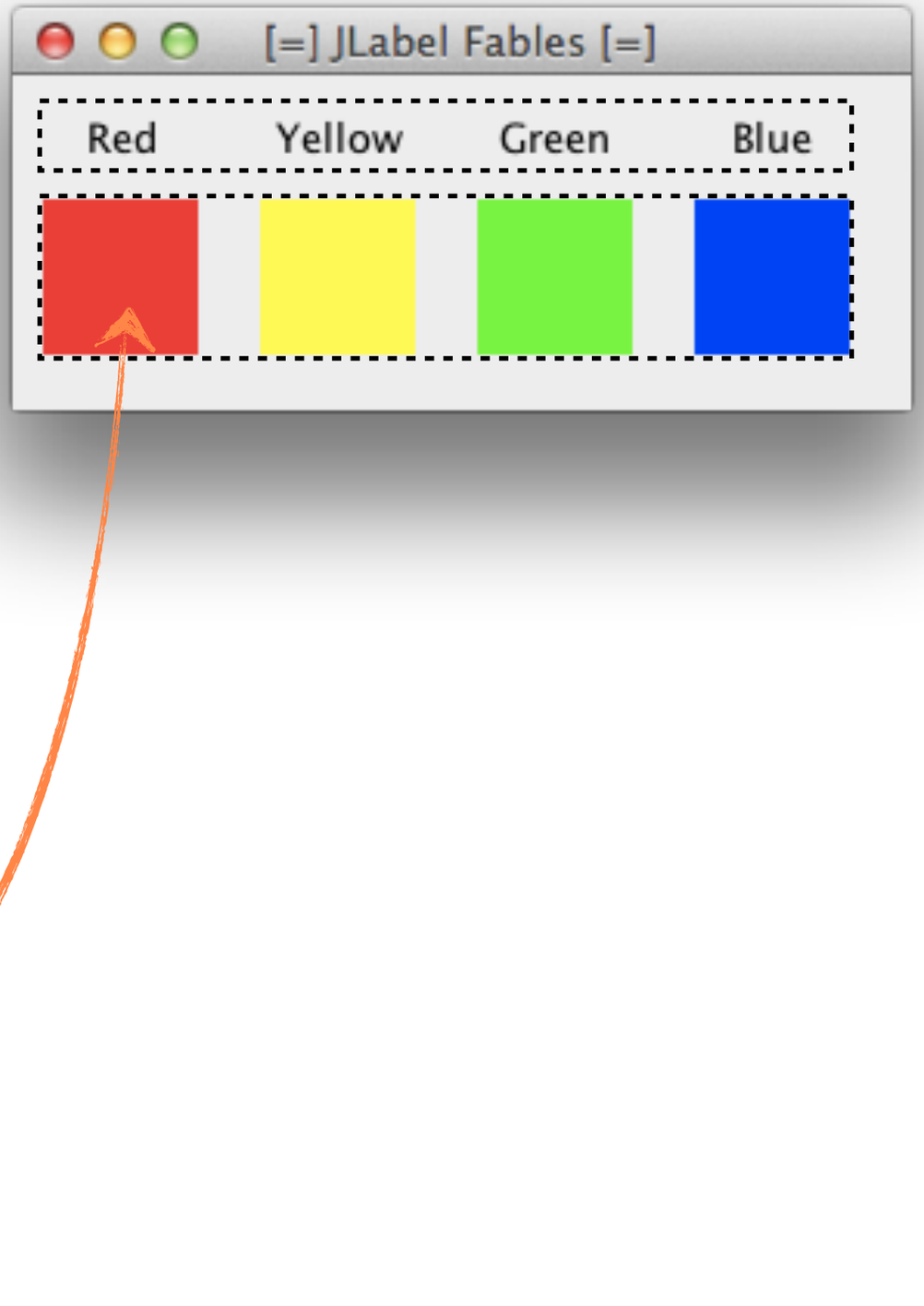
```
JPanel textPanel = new JPanel();
textPanel.setLayout(null);
textPanel.setLocation(10, 0);
textPanel.setSize(260, 30);
mainPanel.add(textPanel);

JLabel redLabel = new JLabel("Red");
redLabel.setLocation(0, 0);
redLabel.setSize(50, 40);
redLabel.setHorizontalAlignment(JLabel.CENTER);
textPanel.add(redLabel);
...
JPanel panelForPanels = new JPanel();
panelForPanels.setLayout(null);
panelForPanels.setLocation(10, 40);
panelForPanels.setSize(260, 50);
mainPanel.add(panelForPanels);

JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(0, 0);
redPanel.setSize(50, 50);
panelForPanels.add(redPanel);
...
```



see: example.swing.misc.LabelExample

The JLabel Widget

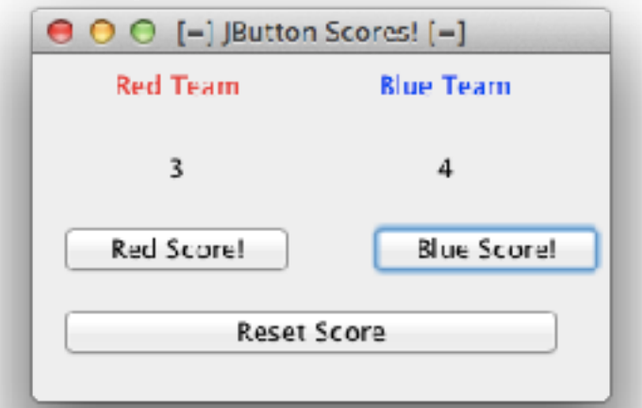
```
JPanel textPanel = new JPanel();
textPanel.setLayout(null);
textPanel.setLocation(10, 0);
textPanel.setSize(260, 30);
mainPanel.add(textPanel);

JLabel redLabel = new JLabel("Red");
redLabel.setLocation(0, 0);
redLabel.setSize(50, 40);
redLabel.setHorizontalAlignment(JLabel.CENTER);
textPanel.add(redLabel);
...
JPanel panelForPanels = new JPanel();
panelForPanels.setLayout(null);
panelForPanels.setLocation(10, 40);
panelForPanels.setSize(260, 50);
mainPanel.add(panelForPanels);

JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(0, 0);
redPanel.setSize(50, 50);
panelForPanels.add(redPanel);
...
```



see: example.swing.misc.LabelExample

The JButton Widget

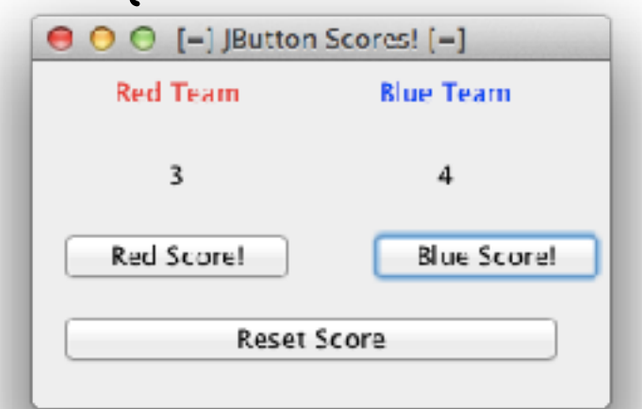
```
public class ButtonExample extends JFrame implements ActionListener {

    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...

    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener(this);
        buttonPanel.add(redButton);
      ...
    }

    public void actionPerformed(ActionEvent e) {
        if ( e.getSource() == redButton ) {
            redScoreAmount = redScoreAmount + 1;
            redScore.setText(""+redScoreAmount);
        } else ...
  }
}
```



see: example.swing.misc.ButtonExample

The JButton Widget

```
public class ButtonExample extends JFrame implements ActionListener {

    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...

    private void initGUI () {
       ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener(this);
        buttonPanel.add(redButton);
       ...
    }

    public void actionPerformed(ActionEvent e) {
        if ( e.getSource() == redButton ) {
            redScoreAmount = redScoreAmount + 1;
            redScore.setText(""+redScoreAmount);
        } else ...
   }
}
```

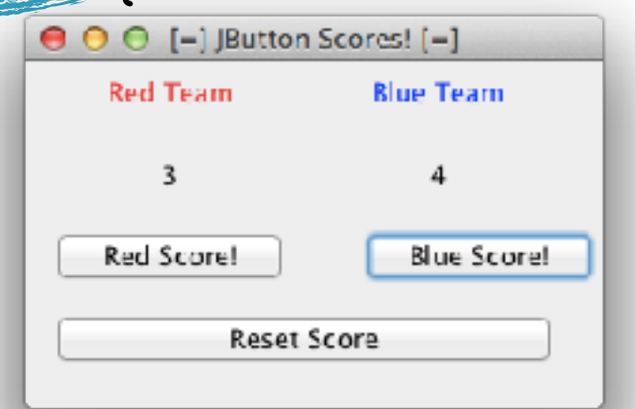


create a button, with the text "Red Score!"

see: example.swing.misc.ButtonExample

The JButton Widget

```
public class ButtonExample extends JFrame implements ActionListener {

    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...

    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener(this);
        buttonPanel.add(redButton);
      ...
    }

    public void actionPerformed(ActionEvent e) {
        if ( e.getSource() == redButton ) {
            redScoreAmount = redScoreAmount + 1;
            redScore.setText(""+redScoreAmount);
        } else ...
    }
}
```



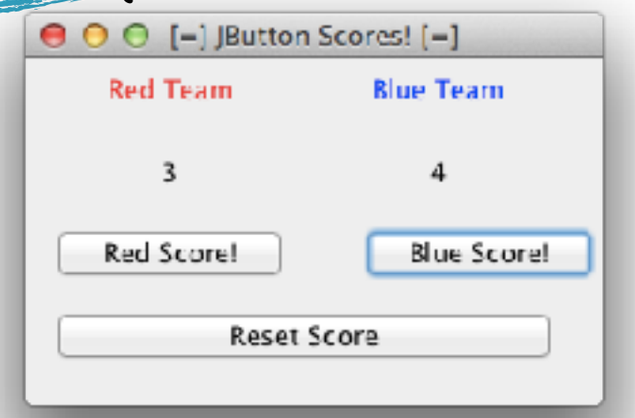
create a button, with the text "Red Score!"

add a Listener, to be called when clicking on the button

see: example.swing.misc.ButtonExample

The JButton Widget

```
public class ButtonExample extends JFrame implements ActionListener {

    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...

    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener(this);
        buttonPanel.add(redButton);
      ...
    }

    public void actionPerformed(ActionEvent e) {
        if ( e.getSource() == redButton ) {
            redScoreAmount = redScoreAmount + 1;
            redScore.setText(""+redScoreAmount);
        } else ...
    }
}
```



widget references are fields, so we can access them later when the listener is called

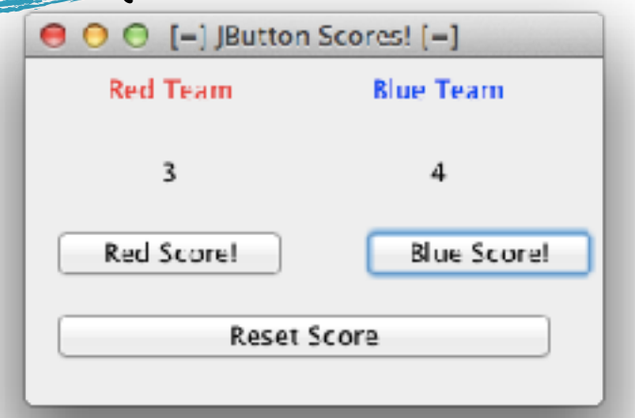
create a button, with the text "Red Score!"

add a Listener, to be called when clicking on the button

see: example.swing.misc.ButtonExample

The JButton Widget

```
public class ButtonExample extends JFrame implements ActionListener {

    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...

    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener(this);
        buttonPanel.add(redButton);
      ...
    }

    public void actionPerformed(ActionEvent e) {
        if ( e.getSource() == redButton ) {
            redScoreAmount = redScoreAmount + 1;
            redScore.setText(""+redScoreAmount);
        } else ...
    }
}
```



widget references are fields, so we can access them later when the listener is called

create a button, with the text "Red Score!"

add a Listener, to be called when clicking on the button

use **e.getSource** to know where the event comes from when the same object is listening to several JButton widgets

see: example.swing.misc.ButtonExample

Use Anonymous Classes as Listeners

```
public class ButtonExample  extends JFrame {
    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...
    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener( new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                redScoreAmount = redScoreAmount + 1;
                redScore.setText("" + redScoreAmount);
            }
        });
        buttonPanel.add(redButton);
      ...
    }
    ...
}
```

see: example.swing.misc.ButtonExample_1

Use Anonymous Classes as Listeners

```
public class ButtonExample  extends JFrame {
    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...
    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener( new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                redScoreAmount = redScoreAmount + 1;
                redScore.setText("" + redScoreAmount);
            }
        });
        buttonPanel.add(redButton);
      ...
    }
    ...
}
```

Every JButton has its own listener, no need to use e.source()

see: example.swing.misc.ButtonExample_1

Use Anonymous Classes as Listeners

```
public class ButtonExample  extends JFrame {
    ...
    JPanel titlePanel, scorePanel, buttonPanel;
    JLabel redLabel, blueLabel, redScore, blueScore;
    JButton redButton, blueButton, resetButton;
    ...
    private void initGUI () {
      ...
        redButton = new JButton("Red Score!");
        redButton.setLocation(0, 0);
        redButton.setSize(120, 30);
        redButton.addActionListener( new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                redScoreAmount = redScoreAmount + 1;
                redScore.setText("" + redScoreAmount);
            }
        });
        buttonPanel.add(redButton);
      ...
    }
    ...
}
```

Note that class ButtonExample does not implement the interface ActionListener, because it does not listen to any JButton.

Every JButton has its own listener, no need to use e.source()

see: example.swing.misc.ButtonExample_1

JButton with Icon


```
private void initGUI() {
    ...
    exitButton = new JButton("Exit");
    exitButton.setLocation(0, 0);
    exitButton.setIcon( new ImageIcon("src/tp/examples/swing/icons/exit.png") );
    mainPanel.add(exitButton);

    exitButton = new JButton("Undo");
    exitButton.setLocation(0, 0);
    exitButton.setIcon( new ImageIcon( IconsDir.class.getResource("undo.png") ));
    mainPanel.add(exitButton);
    ...
}
```

Absolute path or relative to the (project) root

Relative to the path of a class. Here we create an auxiliary class IconsDir in the same directory of the icons for this purpose.

see: example.swing.misc.ButtonExample_2

The JToggleButton Widget

```
public class ToggleButtonExample extends JFrame implements ItemListener {
  ...
  private void initGUI () {
        ...
        toggleButton = new JToggleButton("Off");
        toggleButton.setLocation(75,10);
        toggleButton.setSize(100,100);
        toggleButton.addItemListener(this);
        totalGUI.add(toggleButton);
        ...
    }

    public void itemStateChanged(ItemEvent e) {
        if(e.getStateChange() == ItemEvent.SELECTED) {
            toggleButton.setText("On!");
            totalGUI.setBackground(Color.green);
        }
        ...
    }
    ...
}
```

see: example.swing.misc.ToggleButton, example.swing.misc.ToggleButton_1

The JToggleButton Widget

```
public class ToggleButtonExample extends JFrame implements ItemListener {
  ...
  private void initGUI () {
        ...
        toggleButton = new JToggleButton("Off");
        toggleButton.setLocation(75,10);
        toggleButton.setSize(100,100);
        toggleButton.addItemListener(this);
        totalGUI.add(toggleButton);
        ...
    }

    public void itemStateChanged(ItemEvent e) {
        if(e.getStateChange() == ItemEvent.SELECTED) {
            toggleButton.setText("On!");
            totalGUI.setBackground(Color.green);
        }
        ...
    }
    ...
}
```

create a toggle button, with the text "Off"

see: example.swing.misc.ToggleButton, example.swing.misc.ToggleButton_1

The JToggleButton Widget

```
public class ToggleButtonExample extends JFrame implements ItemListener {
  ...
  private void initGUI () {
        ...
        toggleButton = new JToggleButton("Off");
        toggleButton.setLocation(75,10);
        toggleButton.setSize(100,100);
        toggleButton.addItemListener(this);
        totalGUI.add(toggleButton);
        ...
    }

    public void itemStateChanged(ItemEvent e) {
        if(e.getStateChange() == ItemEvent.SELECTED) {
            toggleButton.setText("On!");
            totalGUI.setBackground(Color.green);
        }
        ...
    }
    ...
}
```

create a toggle button, with the text "Off"

see: example.swing.misc.ToggleButton, example.swing.misc.ToggleButton_1

The JToggleButton Widget

```
public class ToggleButtonExample extends JFrame implements ItemListener {
  ...
  private void initGUI () {
        ...
        toggleButton = new JToggleButton("Off");
        toggleButton.setLocation(75,10);
        toggleButton.setSize(100,100);
        toggleButton.addItemListener(this);
        totalGUI.add(toggleButton);
        ...
    }

    public void itemStateChanged(ItemEvent e) {
        if(e.getStateChange() == ItemEvent.SELECTED) {
            toggleButton.setText("On!");
            totalGUI.setBackground(Color.green);
        }
        ...
    }
    ...
}
```

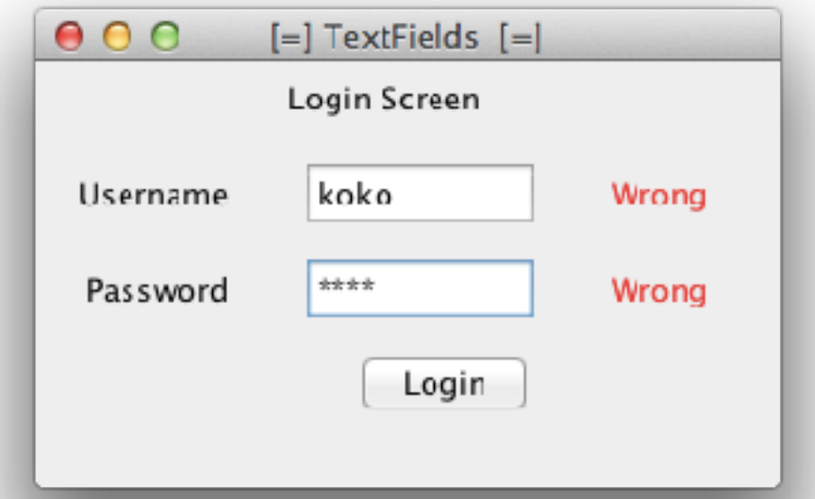
create a toggle button, with the text "Off"

ItemListener tracks the state change, independently from how it was changed. ActionListener tracks actions performed on the components

see: example.swing.misc.ToggleButton, example.swing.misc.ToggleButton_1

The JTextField/JPassword Widgets

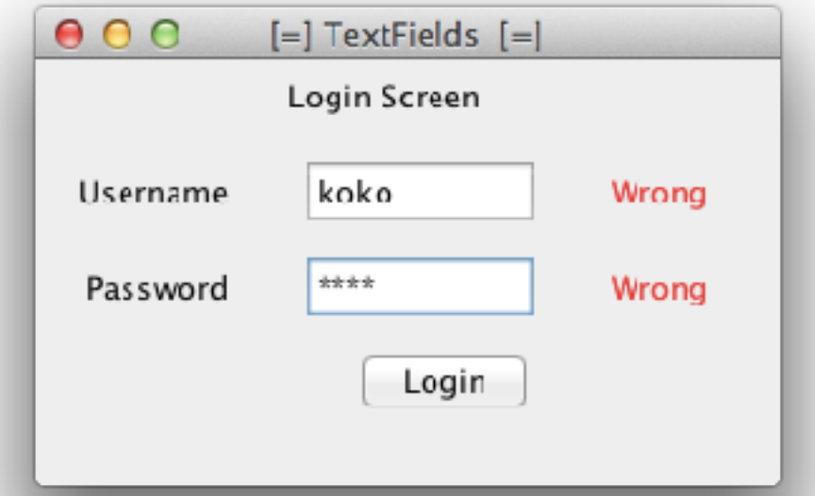
```
public class TextFieldExample extends JFrame implements  ActionListener {

    JTextField usernameField;
    JPasswordField passwdField;

    private void initGUI() {
    ...
        // Username Textfield
        usernameField = new JTextField(8);
        usernameField.setLocation(0, 0);
        usernameField.setSize(100, 30);
        panelForTextFields.add(usernameField);

        // Password Textfield
        passwdField = new JPasswordField(8);
        passwdField.setEchoChar('*');
        passwdField.setLocation(0, 40);
        passwdField.setSize(100, 30);
        panelForTextFields.add(passwdField);
    ...
    }
  ...
}
```



see: example.swing.misc.TextFieldExample

The JTextField/JPassword Widgets

```
public class TextFieldExample extends JFrame implements  ActionListener {

    JTextField usernameField;
    JPasswordField passwdField;

    private void initGUI() {
    ...
        // Username Textfield
        usernameField = new JTextField(8);
        usernameField.setLocation(0, 0);
        usernameField.setSize(100, 30);
        panelForTextFields.add(usernameField);

        // Password Textfield
        passwdField = new JPasswordField(8);
        passwdField.setEchoChar('*');
        passwdField.setLocation(0, 40);
        passwdField.setSize(100, 30);
        panelForTextFields.add(passwdField);
    ...
    }
  ...
}
```

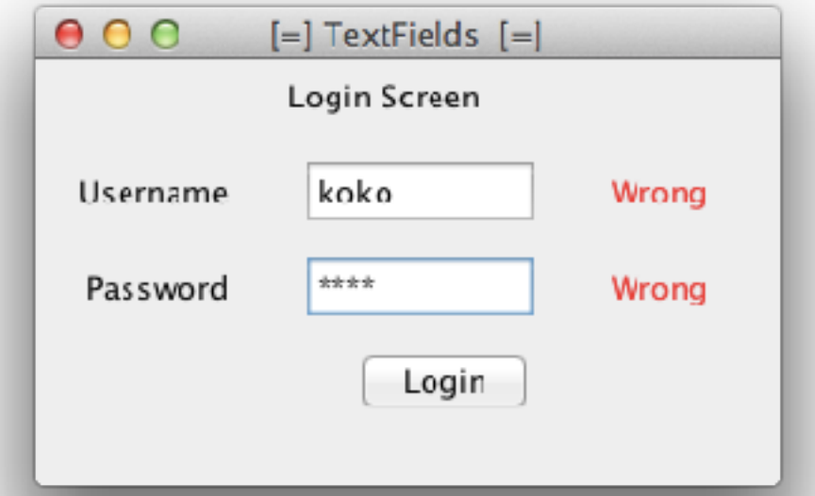


A normal text field

see: example.swing.misc.TextFieldExample

The JTextField/JPassword Widgets

```
public class TextFieldExample extends JFrame implements  ActionListener {

    JTextField usernameField;
    JPasswordField passwdField;

    private void initGUI() {
    ...
        // Username Textfield
        usernameField = new JTextField(8);
        usernameField.setLocation(0, 0);
        usernameField.setSize(100, 30);
        panelForTextFields.add(usernameField);

        // Password Textfield
        passwdField = new JPasswordField(8);
        passwdField.setEchoChar('*');
        passwdField.setLocation(0, 40);
        passwdField.setSize(100, 30);
        panelForTextFields.add(passwdField);
    ...
    }
  ...
}
```



A normal text field

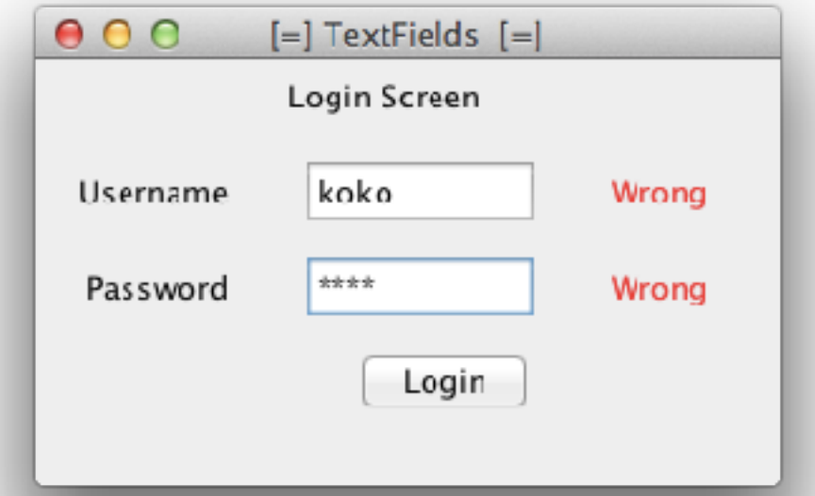
A text field with hidden text

****

see: example.swing.misc.TextFieldExample

The JTextField/JPassword Widgets

```
public class TextFieldExample extends JFrame implements  ActionListener {

    JTextField usernameField;
    JPasswordField passwdField;

    private void initGUI() {
    ...
        // Username Textfield
        usernameField = new JTextField(8);
        usernameField.setLocation(0, 0);
        usernameField.setSize(100, 30);
        panelForTextFields.add(usernameField);

        // Password Textfield
        passwdField = new JPasswordField(8);
        passwdField.setEchoChar('*');
        passwdField.setLocation(0, 40);
        passwdField.setSize(100, 30);
        panelForTextFields.add(passwdField);
    ...
    }
  ...
}
```



A normal text field

A text field with hidden text

****

To access the values use:

```
String s = usernameField.getText();
char[] input = loginField.getPassword();
```

see: example.swing.misc.TextFieldExample

Disable Widgets

```
public class TextFieldExample implements ActionListener {

    JTextField usernameField;
    JPasswordField passwdField;

    private void initGUI() {
        ...
    }

    public void actionPerformed(ActionEvent e) {
        ...

        // When 'Logging In' succeeds, disable
        // the login button.
        loginButton.setEnabled(false);
        ...
    }
    ...
}
```

setEnabled can be used to disable components, unlike setVisible(true) they don't disappear

see: example.swing.misc.TextFieldExample