

GUI

2018年7月23日 23:16

GUI: Graphical User Interface

CLI: command line User Interface

Java.awt

Javax.swing

组件: 基本组件+容器组件

窗体

2018年7月29日 18:13

```
/*
 * 窗体
 */
public class FrameDemo {
    public static void main(String[] args) {
        // 创建窗体对象
        Frame f = new Frame();
        // Frame f = new Frame("helloworld");

        // 设置窗体标题
        f.setTitle("helloworld");

        // 调用一个方法，设置让方法可见
        // f.show();
        f.setVisible(true);

        // 设置窗体大小
        f.setSize(400, 300); // 默认为像素

        // 设置窗体位置
        f.setLocation(400, 200);
    }
}
```

窗体改进

2018年7月29日 22:12

```
public class FrameDemo2 {  
    public static void main(String[] args) {  
        Frame f = new Frame("设置方法调用的前后关系");  
  
        // f.setSize(400,300);  
        Dimension d = new Dimension(400, 300);  
        f.setSize(d);  
  
        // f.setLocation(400, 200);  
        Point p = new Point(400, 200);  
        f.setLocation(p);  
  
        // 一个方法确定位置  
        f.setBounds(400, 200, 400, 300);  
  
        try {  
            Thread.sleep(3000);  
        } catch (InterruptedException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
  
        f.setVisible(true);  
    }  
}
```

事件监听机制

2018年7月29日 22:13

- 1.事件源：事件发生的地方
- 2.事件：要发生的事情
- 3.事件处理：针对发生的事情做出的处理方案
- 4.事件监听：把事件源与事件关联起来

```
import java.awt.Frame;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class FrameDemo {
    public static void main(String[] args) {
        Frame f = new Frame("窗体关闭案例");

        f.setBounds(400, 200, 400, 300);

        // 事件源
        // 事件：关闭
        // 事件处理：关闭窗口(System.out)
        // 事件监听
        f.addWindowListener(new WindowListener() {

            @Override
            public void windowOpened(WindowEvent e) {
            }

            @Override
            public void windowIconified(WindowEvent e) {
            }

            @Override
            public void windowDeiconified(WindowEvent e) {
            }

            @Override
            public void windowDeactivated(WindowEvent e) {
            }
        });
    }
}
```

```
        @Override
        public void windowClosing(WindowEvent e) {
            System.exit(0);
            ;
        }

        @Override
        public void windowClosed(WindowEvent e) {
        }

        @Override
        public void windowActivated(WindowEvent e) {
        }
    });

    f.setVisible(true);
}
}
```

适配器设计模式

2018年7月29日 22:43

接口（方法比较多）-----实现类（需要实现所有方法，哪怕是空实现）

解决方案：

接口（方法比较多）-----适配器类（实现接口，仅仅空实现）-----实现类（用哪个重写哪个）

```
public class FrameDemo {  
    public static void main(String[] args) {  
        Frame f = new Frame("窗体关闭案例");  
  
        f.setBounds(400, 200, 400, 300);  
  
        f.addWindowListener(new WindowAdapter() {  
            @Override  
            public void windowClosing(WindowEvent e) {  
                System.exit(0);  
            }  
        });  
  
        f.setVisible(true);  
    }  
}
```

窗体添加按钮

2018年7月29日 23:01

```
/*
 * 把按钮添加到窗体，并对按钮添加点击事件
 */
public class FrameDemo {
    public static void main(String[] args) {
        Frame f = new Frame("添加按钮");
        f.setBounds(400, 200, 400, 300);
        //设置布局为流式布局
        f.setLayout(new FlowLayout());

        Button bu = new Button("我是按钮");
        bu.setSize(20,10);

        f.add(bu);

        f.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        bu.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                System.out.println("点一点");
            }
        });

        f.setVisible(true);
    }
}
```

窗体布局：窗体中组件的排列方式

流式布局FlowLayout

边界布局BorderLayout

网格布局GridLayout

网格包布局

卡片布局

文本框

2018年7月29日 23:44

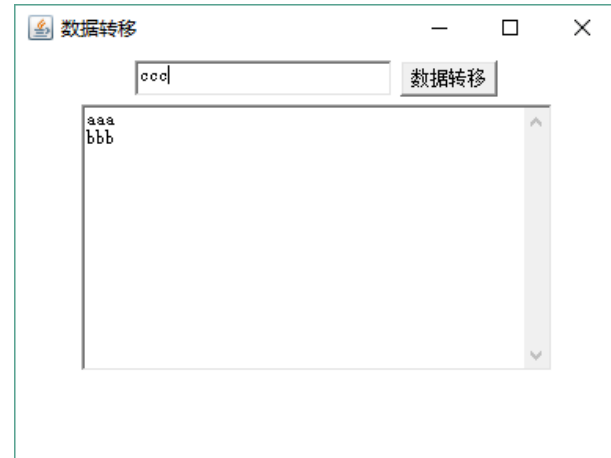
```
public class FrameDemo {
    public static void main(String[] args) {
        Frame f = new Frame("数据转移");
        f.setBounds(400, 200, 400, 300);
        f.setLayout(new FlowLayout());

        // 创建文本框
        TextField tf = new TextField(20);
        Button bu = new Button("数据转移");
        // 创建文本域
        TextArea ta = new TextArea(10, 40);

        // 添加组件
        f.add(tf);
        f.add(bu);
        f.add(ta);

        f.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        // 对按钮添加事件
        bu.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // 获取文本框的值
                String tf_str = tf.getText().trim();
                // 追加给文本域
                ta.append(tf_str + "\r\n");
                // 清空文本框
                tf.setText("");
                // 获取光标
                tf.requestFocus();
            }
        });
        f.setVisible(true);
    }
}
```



鼠标事件

2018年7月30日 9:36

```
public class FrameDemo {
    public static void main(String[] args) {
        Frame f = new Frame("更改背景色");
        f.setBounds(400, 200, 400, 300);
        f.setLayout(new FlowLayout());

        Button redButton = new Button("红色");
        f.add(redButton);

        f.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        // 对按钮添加鼠标进入事件
        redButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                f.setBackground(Color.RED);
            }
        });

        redButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseExited(MouseEvent e) {
                f.setBackground(Color.WHITE);
            }
        });

        f.setVisible(true);
    }
}
```



键盘录入事件

2018年7月30日 9:37

```
public class FrameDemo {
    public static void main(String[] args) {
        Frame f = new Frame("不能输入非数字字符");
        f.setBounds(400, 200, 400, 300);
        f.setLayout(new FlowLayout());

        // 创建Label对象
        Label label = new Label("请输入你的QQ号码，不能是非数字");
        TextField tf = new TextField(40);

        f.add(label);
        f.add(tf);

        f.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        tf.addKeyListener(new KeyAdapter() {
            @Override
            public void keyPressed(KeyEvent e) {
                // 不是数字字符取消事件
                char ch = e.getKeyChar();
                if (!(ch >= '0' && ch <= '9')) {
                    e.consume();
                }
            }
        });

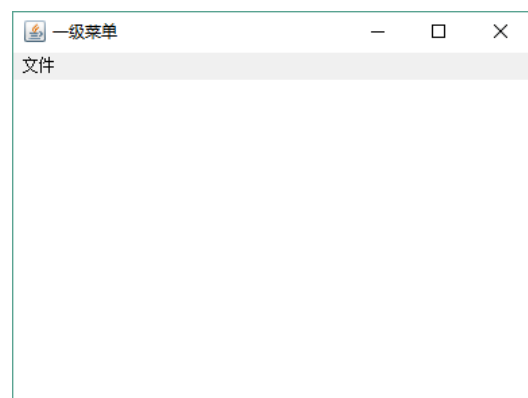
        f.setVisible(true);
    }
}
```

一级菜单

2018年7月30日 10:05

菜单栏
菜单项
菜单

```
public class FrameDemo {  
    public static void main(String[] args) {  
        Frame f = new Frame("一级菜单");  
        f.setBounds(400, 200, 400, 300);  
        f.setLayout(new FlowLayout());  
  
        // 创建菜单栏  
        MenuBar mb = new MenuBar();  
        // 创建菜单  
        Menu m = new Menu("文件");  
        // 创建菜单项  
        MenuItem mi = new MenuItem("退出系统");  
  
        m.add(mi);  
        mb.add(m);  
        f.setMenuBar(mb);  
  
        f.addWindowListener(new WindowAdapter() {  
            @Override  
            public void windowClosing(WindowEvent e) {  
                System.exit(0);  
            }  
        });  
  
        mi.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                System.exit(0);  
            }  
        });  
  
        f.setVisible(true);  
    }  
}
```



多级菜单

2018年7月30日 10:15

```
public class FrameDemo2 {
    public static void main(String[] args) {
        Frame f = new Frame("多级菜单");
        f.setBounds(400, 200, 400, 300);
        f.setLayout(new FlowLayout());

        // 创建菜单栏
        MenuBar mb = new MenuBar();
        // 创建菜单
        Menu m1 = new Menu("文件");
        Menu m2 = new Menu("更改名称");
        // 创建菜单项
        MenuItem mi1 = new MenuItem("好好学习");
        MenuItem mi2 = new MenuItem("天天向上");
        MenuItem mi3 = new MenuItem("恢复标题");
        MenuItem mi4 = new MenuItem("打开记事本");
        MenuItem mi5 = new MenuItem("退出系统");

        m2.add(mi1);
        m2.add(mi2);
        m2.add(mi3);
        m1.add(m2);
        m1.add(mi4);
        m1.add(mi5);
        mb.add(m1);
        f.setMenuBar(mb);

        mi1.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                f.setTitle(mi1.getLabel());
            }
        });

        mi4.addActionListener(new ActionListener() {
```

```

        @Override
        public void actionPerformed(ActionEvent e) {
            Runtime r = Runtime.getRuntime();
            try {
                r.exec("notepad");
            } catch (IOException e1) {
                e1.printStackTrace();
            }
        }
    });

    f.addWindowListener(new WindowAdapter() {
        @Override
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });

    mi5.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.exit(0);
        }
    });

    f.setVisible(true);
}
}

```