

## 布局管理器

```
1 凌子健 16:44:09
2 布局管理器
3 import java.awt.*;
4 import javax.swing.*;
5 public class amDemo extends JFrame{
6     //布局管理器
7     /* BorderLayout
8     amDemo(){
9         setLayout(new BorderLayout());
10        setSize(250,200);
11        setVisible(true);
12        JButton b1=new JButton("north");
13        JButton b2=new JButton("east");
14        JButton b3=new JButton("west");
15        JButton b4=new JButton("south");
16        JButton b5=new JButton("center");
17        add(b1,BorderLayout.NORTH);
18        add(b2,BorderLayout.EAST);
19        add(b3,BorderLayout.WEST);
20        add(b4,BorderLayout.SOUTH);
21        add(b5,BorderLayout.CENTER);
22    }
23    */
24    /*FlowLayout
25    amDemo(){
26        setLayout(new FlowLayout());
27        setSize(200,250);
28        setVisible(true);
29        add(new JButton("left"),FlowLayout.LEFT);
30        add(new JButton("center"),FlowLayout.CENTER);
31        add(new JButton("right"),FlowLayout.RIGHT);
32    }
33    */
34    /*CardLayout
35    amDemo(){
36        setLayout(new CardLayout());
37        setSize(200,250);
38        setVisible(true);
39        add("1",new JButton("1"));
40        add("2",new JButton("2"));
41        add("3",new JButton("3"));
42
43    }
44    */
45    //还有GridLayout和GirdBagLayout
46    public static void main(String[] args) {
47        EventQueue.invokeLater(new Runnable(){
48            public void run(){
49                amDemo a=new amDemo();
50            }
51        });
52    }
53 }
54
55 凌子健 16:44:23
56 事件处理机制
```

```

57 import java.awt.*;
58 import java.awt.event.ActionEvent;
59 import java.awt.event.ActionListener;
60 import javax.swing.*;
61
62 import static javax.swing.JOptionPane.ERROR_MESSAGE;
63 /*方法1: 让窗体本身来完成这个监听器
64 public class GuiTest extends JFrame implements ActionListener {
65     JButton b1,b2;
66     public GuiTest(){
67         setLayout(new BorderLayout());
68         setVisible(true);
69         setBounds(200,200,300,200);
70         b1=new JButton("进入");
71         b2=new JButton("退出");
72         add(b1,BorderLayout.CENTER);
73         add(b2,BorderLayout.EAST);
74         b1.addActionListener(this);
75         b2.addActionListener(this);
76     }
77     public void actionPerformed(ActionEvent e){
78         if(e.getSource()==b1) {
79             JOptionPane.showMessageDialog(null, "alert", "alert", ERROR_MESSAGE);
80         }
81         else if(e.getSource()==b2){
82             System.exit(ERROR);
83         }
84     }
85     */
86 /*方法2: 匿名类（重点掌握）
87     */
88 public class GuiTest extends JFrame {
89     JButton b1=new JButton("禁用");
90     GuiTest(){
91         setLayout(new BorderLayout());
92         setBounds(200,200,300,200);
93         setVisible(true);
94         add(b1,BorderLayout.CENTER);
95         b1.addActionListener(new ActionListener(){
96             public void actionPerformed(ActionEvent e){
97                 if(e.getSource()==b1)b1.setEnabled(false);
98             }
99         });
100     }
101 // */
102     public static void main(String[] args){
103         EventQueue.invokeLater(new Runnable(){
104             public void run() {
105                 GuiTest a = new GuiTest();
106             }
107         });
108     }
109
110 }
111
112

```

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4  public class PaintDemo extends JFrame{
5      /*画圆
6      public PaintDemo(){
7          setBounds(200,200,300,200);
8          setVisible(true);
9      }
10     public void paint(Graphics g){
11         super.paint(g);
12         g.setColor(Color.red);
13         g.drawOval(50,50,50,50);
14         g.fillOval(100,100,100,100);
15     }
16     */
17     /*点击鼠标让圆移动
18     */
19     int i=0;
20     public PaintDemo(){
21         setBounds(200,200,500,300);
22         setVisible(true);
23         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         addMouseListener(new MouseAdapter(){
25             public void mouseClicked(MouseEvent e){
26                 i++;
27                 repaint();
28             }
29         });
30     }
31     public void paint(Graphics g){
32         super.paint(g);
33         if((i!=0)&&(300>150+i*10)&&(300>150+i*10))
34             g.fillOval(50+i*10,50+i*10,100,100);
35         else if(i!=0) i=0;
36     }
37     // */
38     public static void main(String[] args){
39         EventQueue.invokeLater(new Runnable(){
40             public void run(){
41                 PaintDemo aa=new PaintDemo();
42             }
43         });
44     }
45 }

```

## 多线程

```

1  /*    //方法: Thread.yield()
2  //public class athread extends Thread{
3  public class athread implements Runnable{
4      private int i0;
5      private int m;
6      public athread(int first,int order){
7          i0=first;
8          m=order;
9      }
10     public void run(){
11         for(int i=i0;i<=20;i+=1){

```

```

12         Thread.yield();
13         System.out.println(m);
14         Thread.yield();
15     }
16 }
17 }
18 class Test{
19     public static void main(String[] args){
20         //athread a1=new athread(1,1);
21         //athread a2=new athread(0,2);
22         Thread a1=new Thread(new athread(1,1));
23         Thread a2=new Thread(new athread(0,2));
24         a1.start();
25         a2.start();
26         System.out.println("main over");
27     }
28 }
29 */
30 // 第二种方法: join();
31 class EvenOdd extends Thread{
32     private int order;
33     public EvenOdd(int order){
34         this.order=order;
35     }
36     public void run(){
37         for(int i=0;i<=20;i+=2){
38             System.out.println(order);
39         }
40     }
41 }
42 public class athread{
43     public static void main(String[] args){
44         EvenOdd ot=new EvenOdd(1);
45         ot.start();
46
47         try{
48             System.out.println("Before ot.join");
49             ot.join();    //join必须带有try catch语句
50         }catch(Exception e){
51             e.printStackTrace();
52         }
53
54         System.out.println("over");
55     }
56 }
57 }
58 //其他方法: setDaemon (true) 设置后台进程
59 //thread.setPriority(n)设置优先级

```

## 独立线程画图

```

1 import java.awt.event.*;
2 import java.awt.*;
3 import javax.swing.*;
4 //最好用Runnable方法
5 public class myFrame extends JFrame implements Runnable{
6     int i=0;
7     Thread t=new Thread();
8     public myFrame(){

```

```

9      setVisible(true);
10     setBounds(200,200,500,300);
11     setLayout(null);
12     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13     //重点: 要记下来
14     addMouseListener(new MouseAdapter(){
15         public void mouseClicked(MouseEvent e){
16             startT();
17         }
18     });
19     //
20 }
21 private void startT(){
22     //这里, 如果使用implements Runnable方法, 这里一定要加this;
23     t=new Thread(this);
24     t.start();
25 }
26 public void run(){           //线程start调用
27     for(i=1;i<=10;i++){
28         try {
29             repaint();
30             Thread.sleep(100);
31         }catch(Exception e){
32
33         }
34     }
35 }
36 public void paint(Graphics g){ //自动调用
37     super.paint(g);
38     if(i!=0)
39         g.fillOval(50+i*10,50+i*10,100,100);
40 }
41 public static void main(String[] args) {
42
43     EventQueue.invokeLater(new Runnable(){
44         public void run(){
45             myFrame e=new myFrame();
46         }
47     });
48 }
49 }
50

```

## 利用鼠标事件启动3个线程分别在三个窗口中同时绘制动态图形

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4  public class TestFrame extends JFrame implements Runnable{
5      int i=0;
6      Thread t1=new Thread(this);
7      TestFrame(){
8          setVisible(true);
9          setBounds(200,200,500,300);
10         setLayout(null);
11         addMouseListener(new MouseAdapter(){
12             public void mouseClicked(MouseEvent e){
13                 startT();
14             }
15         }
16     }
17 }
18

```

```
15     });
16 }
17 public void startT(){
18     t1=new Thread(this);
19     t1.start();
20 }
21 public void run(){
22     for(i=1;i<=55;i++)
23     try{
24         repaint();
25         Thread.sleep(50);
26     }catch(Exception e){
27
28     }
29
30 }
31 public void paint(Graphics g){
32     super.paint(g);
33     if(i!=0)
34         g.fillOval(i*1+50,i*1+50,100,100);
35 }
36 public static void main(String[] args){
37     EventQueue.invokeLater(new Runnable(){
38         public void run(){
39             TestFrame wwe=new TestFrame();
40             wwe.setBounds(100,100,300,300);
41             TestFrame wwe1=new TestFrame();
42             wwe1.setBounds(400,100,300,300);
43             TestFrame wwe2=new TestFrame();
44             wwe2.setBounds(700,100,300,300);
45
46         }
47     });
48 }
49 }
50
```