布局管理器

```
凌子健 16:44:09
布局管理器
import java.awt.*;
import javax.swing.*;
public class amDemo extends JFrame{
    //布局管理器
    /* BorderLayout
    amDemo(){
        setLayout(new BorderLayout());
        setSize(250,200);
        setVisible(true);
        JButton b1=new JButton("north");
        JButton b2=new JButton("east");
        JButton b3=new JButton("west");
        JButton b4=new JButton("south");
        JButton b5=new JButton("center");
        add(b1,BorderLayout.NORTH);
        add(b2,BorderLayout.EAST);
        add(b3,BorderLayout.WEST);
        add(b4,BorderLayout.SOUTH);
        add(b5,BorderLayout.CENTER);
    }
    */
    /*FlowLayout
    amDemo(){
        setLayout(new FlowLayout());
        setSize(200,250);
        setVisible(true);
        add(new JButton("left"),FlowLayout.LEFT);
        add(new JButton("center"),FlowLayout.CENTER);
        add(new JButton("right"),FlowLayout.RIGHT);
    }
    */
    /*CardLayout
    amDemo(){
        setLayout(new CardLayout());
        setSize(200,250);
        setVisible(true);
        add("1",new JButton("1"));
        add("2",new JButton("2"));
        add("3",new JButton("3"));

    }
     */
    //还有GridLayout和GirdBagLayout
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable(){
            public void run(){
                amDemo a=new amDemo();
            }
        });
    }
}

凌子健 16:44:23
事件处理机制
```

```java
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

import static javax.swing.JOptionPane.ERROR_MESSAGE;
/*方法1：让窗体本身来完成这个监听器
public class GuiTest extends JFrame implements ActionListener {
    JButton b1,b2;
    public GuiTest(){
        setLayout(new BorderLayout());
        setVisible(true);
        setBounds(200,200,300,200);
        b1=new JButton("进入");
        b2=new JButton("退出");
        add(b1,BorderLayout.CENTER);
        add(b2,BorderLayout.EAST);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==b1) {
            JOptionPane.showMessageDialog(null, "alert", "alert", ERROR_MESSAGE);
        }
        else if(e.getSource()==b2){
            System.exit(ERROR);
        }
    }
     */
/*方法2：匿名类（重点掌握）
 */
public class GuiTest extends JFrame {
    JButton b1=new JButton("禁用");
    GuiTest(){
        setLayout(new BorderLayout());
        setBounds(200,200,300,200);
        setVisible(true);
        add(b1,BorderLayout.CENTER);
        b1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                if(e.getSource()==b1)b1.setEnabled(false);
            }
        });
    }
// */
    public static void main(String[] args){
        EventQueue.invokeLater(new Runnable(){
            public void run() {
                GuiTest a = new GuiTest();
            }
        });
    }

}


```

画图

```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class PaintDemo extends JFrame{
    /*画圆
    public PaintDemo(){
        setBounds(200,200,300,200);
        setVisible(true);
    }
    public void paint(Graphics g){
        super.paint(g);
        g.setColor(Color.red);
        g.drawOval(50,50,50,50);
        g.fillOval(100,100,100,100);
    }
     */
    /*点击鼠标让圆移动
     */
    int i=0;
    public PaintDemo(){
        setBounds(200,200,500,300);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        addMouseListener(new MouseAdapter(){
            public void mouseClicked(MouseEvent e){
                i++;
                repaint();
            }
        });
    }
    public void paint(Graphics g){
        super.paint(g);
        if((i!=0)&&(300>150+i*10)&&(300>150+i*10))
            g.fillOval(50+i*10,50+i*10,100,100);
        else if(i!=0) i=0;
    }
    // */
    public static void main(String[] args){
        EventQueue.invokeLater(new Runnable(){
            public void run(){
                PaintDemo aa=new PaintDemo();
            }
        });
    }
}
```

多线程

```java
/*   //方法：Thread.yield()
//public class athread extends Thread{
public class athread implements Runnable{
    private int i0;
    private int m;
    public athread(int first,int order){
        i0=first;
        m=order;
    }
    public void run(){
        for(int i=i0;i<=20;i+=1){
```

```
            Thread.yield();
            System.out.println(m);
            Thread.yield();
        }
    }
}
class Test{
    public static void main(String[] args){
        //athread a1=new athread(1,1);
        //athread a2=new athread(0,2);
        Thread a1=new Thread(new athread(1,1));
        Thread a2=new Thread(new athread(0,2));
        a1.start();
        a2.start();
        System.out.println("main over");
    }
}
*/
  // 第二种方法：join();
class EvenOdd extends Thread{
    private int order;
    public EvenOdd(int order){
        this.order=order;
    }
    public void run(){
        for(int i=0;i<=20;i+=2){
            System.out.println(order);
        }
    }
}
public class athread{
    public static void main(String[] args){
        EvenOdd ot=new EvenOdd(1);
        ot.start();

        try{
            System.out.println("Before ot.join");
            ot.join();   //join必须带有try catch语句
        }catch(Exception e){
            e.printStackTrace();
        }

        System.out.println("over");

    }
}
//其他方法：setDaemon（true）设置后台进程
//thread.setPriority(n)设置优先级
```

独立线程画图

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
//最好用Runnable方法
public class myFrame extends JFrame implements Runnable{
    int i=0;
    Thread t=new Thread();
    public myFrame(){
```

```
        setVisible(true);
        setBounds(200,200,500,300);
        setLayout(null);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //重点：要记下来
        addMouseListener(new MouseAdapter(){
            public void mouseClicked(MouseEvent e){
                startT();
            }
        });
        //
    }
    private void startT(){
        //这里，如果使用implements Runnable方法，这里一定要加this；
        t=new Thread(this);
        t.start();
    }
    public void run(){          //线程start调用
        for(i=1;i<=10;i++){
            try {
                repaint();
                Thread.sleep(100);
            }catch(Exception e){

            }
        }
    }
    public void paint(Graphics g){  //自动调用
        super.paint(g);
        if(i!=0)
            g.fillOval(50+i*10,50+i*10,100,100);
    }
    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable(){
            public void run(){
                myFrame e=new myFrame();
            }
        });
    }
}

```

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

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class TestFrame extends JFrame implements Runnable{
    int i=0;
    Thread t1=new Thread(this);
    TestFrame(){
        setVisible(true);
        setBounds(200,200,500,300);
        setLayout(null);
        addMouseListener(new MouseAdapter(){
            public void mouseClicked(MouseEvent e){
                startT();
            }
```

```
        });
    }
    public void startT(){
        t1=new Thread(this);
        t1.start();
    }
    public void run(){
        for(i=1;i<=55;i++)
        try{
            repaint();
            Thread.sleep(50);
        }catch(Exception e){

        }

    }
    public void paint(Graphics g){
        super.paint(g);
        if(i!=0)
            g.fillOval(i*1+50,i*1+50,100,100);
    }
    public static void main(String[] args){
        EventQueue.invokeLater(new Runnable(){
            public void run(){
                TestFrame wwe=new TestFrame();
                wwe.setBounds(100,100,300,300);
                TestFrame wwe1=new TestFrame();
                wwe1.setBounds(400,100,300,300);
                TestFrame wwe2=new TestFrame();
                wwe2.setBounds(700,100,300,300);

            }
        });
    }
}

```