GUI编程：

CS系统没落的原因：就是他的维护成本高，原因就在，需要开发两套系统，一套是客户端，另外一套是服务端。

第一代图形化界面编辑器awt：不再使用的原因，就是他不能完全实现跨平台

第二代图形化界面编辑器swing：是可以实现跨平台

1. 为了我们基础的磨练，以及做出我们的第一个项目。
2. swing的学习有利于我们今后学习安卓。

组件

布局管理器

事件处理

图形

Component 和Container是swing的核心类。

容器主要分为两种：

第一种windows：可以自由停泊的顶级窗口

JFrame:创建一个windows窗口，可以进行布局，也可以为子类创建窗体。

窗口类，可以显示。

JPanel：是不可以停泊的，他的对象可以作为Component对象，是不能独立存在的，需要放在其他的容器里比如JFrame，当然这个容器也是可以进行布局的、

面板类，通过windows显示、

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** javax.swing.JFrame;  **public** **class** SwingDemo {  **public** **static** **void** main(String[] args) {  JFrame jf =**new** JFrame("第一个窗口");    jf.setBounds(100, 100, 200, 200);    jf.setVisible(**true**);  }  } |

JFrame的构造方法可以是默认的未命名的，也可以直接在构造的时候就传入窗口的名字。

setVisible(Boolean b);

setTitle(String s);

setSize(int w,int h);

setBounds(x,y,w,h);

setResizble()

setLayout(布局管理器的对象)：

JPanel：

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.Color;  **import** javax.swing.JFrame;  **import** javax.swing.JPanel;  **public** **class** SwingDemo {  **public** **static** **void** main(String[] args) {  JFrame jf =**new** JFrame("第一个窗口");  JPanel jp =**new** JPanel();  jf.setLayout(**null**);    jf.setBounds(100, 100, 200, 200);  jf.setBackground(Color.*BLUE*);    jp.setBounds(50, 50, 100, 100);  jp.setBackground(Color.*RED*);    jf.add(jp);    jf.setVisible(**true**);  }  } |

布局管理器：

流式布局（FlowLayout）：从左到右一行排满就放下一行，组件的大小不会改变，按照原始尺寸，但是我们可以设置组件的间距，行距，以及对其方式。

流式布局默认的对齐方式，居中。在安卓当中我们称这个布局叫线性布局。

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.Color;  **import** java.awt.FlowLayout;  **import** javax.swing.JButton;  **import** javax.swing.JFrame;  **public** **class** SwingDemo {  **public** **static** **void** main(String[] args) {  JFrame jf =**new** JFrame("第一个窗口");  JButton jbut1 = **new** JButton("按键1");  JButton jbut2 = **new** JButton("按键2");  JButton jbut3 = **new** JButton("按键3");  JButton jbut4 = **new** JButton("按键4");  JButton jbut5 = **new** JButton("按键5");  JButton jbut6 = **new** JButton("按键6");  JButton jbut7 = **new** JButton("按键7");  JButton jbut8 = **new** JButton("按键8");  JButton jbut9 = **new** JButton("按键9");    jf.setLayout(**new** FlowLayout(FlowLayout.*CENTER*,10,10));    jf.setBounds(100, 100, 200, 200);  jf.setBackground(Color.*BLUE*);        jf.add(jbut1);  jf.add(jbut2);  jf.add(jbut3);  jf.add(jbut4);  jf.add(jbut5);  jf.add(jbut6);  jf.add(jbut7);  jf.add(jbut8);  jf.add(jbut9);    jf.setVisible(**true**);  }  } |

JButton是按钮组件

FlowLayout是JPanel的默认布局方式。

边界布局(BorderLayout)

这种布局是JFrame的默认布局管理器

分成东西南北中五个区域，组件要添加到指定的区域里面去，如果不指定，那就默认放在Center（中）区域，每个区域只能放一个组件，多个组件加入的话，会被先前的覆盖。

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.BorderLayout;  **import** java.awt.Color;  **import** javax.swing.JFrame;  **import** javax.swing.JPanel;  **public** **class** SwingDemo {  **public** **static** **void** main(String[] args) {  JFrame jf =**new** JFrame("BorderLayoutDemo");  jf.setLayout(**new** BorderLayout());    jf.setBounds(100, 100, 200, 200);      JPanel jp1= **new** JPanel();  jp1.setBackground(Color.*BLACK*);  JPanel jp2= **new** JPanel();  jp2.setBackground(Color.*BLUE*);  JPanel jp3= **new** JPanel();  jp3.setBackground(Color.*CYAN*);  JPanel jp4= **new** JPanel();  jp4.setBackground(Color.*GREEN*);  JPanel jp5= **new** JPanel();  jp5.setBackground(Color.*MAGENTA*);  jf.add(jp1,BorderLayout.*EAST*);  jf.add(jp2,BorderLayout.*WEST*);  jf.add(jp3,BorderLayout.*NORTH*);  jf.add(jp4,BorderLayout.*SOUTH*);  jf.add(jp5,BorderLayout.*CENTER*);      jf.setResizable(**false**);    jf.setVisible(**true**);  }  } |

BorderLayout的缩放原则：

南北区域称为水平缩放public void **setHgap**(int hgap)

东西区域称为垂直缩放public void **setVgap**(int vgap)

表格布局：GridL:ayout

把空间分局矩阵网格，每个区域大小一致，从左到右填入，满一行后换行。

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.GridLayout;  **import** javax.swing.JButton;  **import** javax.swing.JFrame;  **public** **class** SwingDemo {  **public** **static** **void** main(String[] args) {  JFrame jf =**new** JFrame("BorderLayoutDemo");  jf.setLayout(**new** GridLayout(3,3,10,10));    jf.setBounds(100, 100, 800, 500);      JButton jbut1 = **new** JButton("按键1");  JButton jbut2 = **new** JButton("按键2");  JButton jbut3 = **new** JButton("按键3");  JButton jbut4 = **new** JButton("按键4");  JButton jbut5 = **new** JButton("按键5");  JButton jbut6 = **new** JButton("按键6");  JButton jbut7 = **new** JButton("按键7");  JButton jbut8 = **new** JButton("按键8");  JButton jbut9 = **new** JButton("按键9");  jf.add(jbut1);  jf.add(jbut2);  jf.add(jbut3);  jf.add(jbut4);  jf.add(jbut5);  jf.add(jbut6);  jf.add(jbut7);  jf.add(jbut8);  jf.add(jbut9);    jf.setResizable(**false**);    jf.setVisible(**true**);  }  } |

构造函数里的参数（rows,clows,行距，间距）

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
|  | |
|  |  |  |  |
|  |  |

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.BorderLayout;  **import** java.awt.GridLayout;  **import** javax.swing.JButton;  **import** javax.swing.JFrame;  **import** javax.swing.JPanel;  **public** **class** LayoutDemo {  **public** **static** **void** main(String[] args) {  JFrame jf = **new** JFrame();  jf.setLayout(**new** GridLayout(2,1));  jf.setBounds(100, 100, 800, 400);  JPanel jp1 =**new** JPanel(**new** BorderLayout());  JPanel jp2 =**new** JPanel(**new** GridLayout(2,1));  JButton jbut1=**new** JButton("1");  JButton jbut2=**new** JButton("2");  JButton jbut3=**new** JButton("3");  JButton jbut4=**new** JButton("4");  jp1.add(jbut1,BorderLayout.*WEST*);  jp1.add(jbut4,BorderLayout.*EAST*);  jp1.add(jp2,BorderLayout.*CENTER*);  jp2.add(jbut2);  jp2.add(jbut3);    JPanel jp3=**new** JPanel(**new** GridLayout(1,4));  JButton jbut5=**new** JButton("5");  JPanel jp4=**new** JPanel(**new** GridLayout(2,1));  JButton jbut6=**new** JButton("6");  JButton jbut7=**new** JButton("7");  jp4.add(jbut6);  jp4.add(jbut7);  JPanel jp5=**new** JPanel(**new** GridLayout(2,1));  JButton jbut8=**new** JButton("8");  JButton jbut9=**new** JButton("9");  jp5.add(jbut8);  jp5.add(jbut9);  JButton jbut10=**new** JButton("10");    jp3.add(jbut5);  jp3.add(jp4);  jp3.add(jp5);  jp3.add(jbut10);            jf.add(jp1);  jf.add(jp3);  jf.setVisible(**true**);      }  } |

JTextField 文本框

JLable 标签

事件的监听：

事件源对象🡪发生事件🡪实现某种监听器接口的对象🡪事件处理

按钮事件：

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.GridLayout;  **import** java.awt.event.ActionEvent;  **import** java.awt.event.ActionListener;  **import** javax.swing.JButton;  **import** javax.swing.JFrame;  **public** **class** ButtonEventDemo **implements** ActionListener{  **public** **static** **void** main(String[] args) {  ButtonEventDemo bed=**new** ButtonEventDemo();  bed.getMain();  }  JButton jbut=**new** JButton("按键1");  JButton jbut1=**new** JButton("按键2");  **public** **void** getMain(){  JFrame jf=**new** JFrame("按钮监听");  jf.setLayout(**new** GridLayout(2,1));  jf.setBounds(100, 100, 100, 100);    jf.add(jbut);  jf.add(jbut1);  jbut.addActionListener(**this**);  jbut1.addActionListener(**this**);    jf.setVisible(**true**);  }  @Override  **public** **void** actionPerformed(ActionEvent e) {  **if**(jbut.equals(e.getSource())){  System.*out*.println(e.getActionCommand()+"：点我干啥？");  }**else**{  System.*out*.println(e.getActionCommand()+":快点我呀！");  }    }  } |

e.getSource()返回的是发起事件对象。

e.getActionCommand()返回该对象的标记（名字或者值）

适配器类：

适配器类是实现了某种监听接口的类：

|  |
| --- |
| **package** com.wanczy.SwingDemo;  **import** java.awt.event.MouseAdapter;  **import** java.awt.event.MouseEvent;  **public** **class** ButtonListener **extends** MouseAdapter{  **public** **void** mouseClicked(MouseEvent e){  System.*out*.println("我别点了"+"\t");  }    **public** **void** mouseExited(MouseEvent e){  System.*out*.println("爸爸去哪儿？"+"\t");  }  }  **package** com.wanczy.SwingDemo;  **import** java.awt.event.MouseListener;  **import** javax.swing.JButton;  **import** javax.swing.JFrame;  **public** **class** AdapterDemo {  **public** **static** **void** main(String[] args) {  JFrame jf = **new** JFrame("Adapter");  jf.setBounds(100, 100, 100, 100);  JButton jbut=**new** JButton("按钮");  jf.add(jbut);  //注册监听  jbut.addMouseListener(**new** ButtonListener());  jf.setVisible(**true**);  }  } |