# 1.awt和swing的区别

所有图形组件类都在java.awt和javax.swing包里

在awt包中存在的图形类,在swing一样存在,区别在swing的图形类都是以J开头的

awt和swing包的区别:

1. awt包中的所有图形类的图形都是依赖系统的图形库的,而swing中的图形类的都是sun自己实现的

所有的图形类都称为组件:

1)容器组件

2)非容器组件

# 2.JFrame类

## 2.1定义

Class JFrame

java.lang.Object

java.awt.Component

java.awt.Container

java.awt.Window

java.awt.Frame

javax.swing.JFrame

实现的接口: ImageObserver, MenuContainer, Serializable, Accessible, RootPaneContainer, WindowConstants

## 2.2注意事项

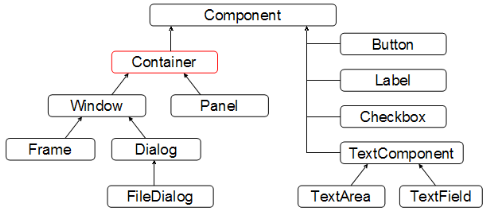
1. 默认是不可见的,setVisible(true)可见
2. 默认产生的窗口点击叉叉按钮后,程序是不会关闭的,要设置窗体关闭事件

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

## 2.3常用方法

|  |  |
| --- | --- |
| public JFrame()  throws HeadlessException | 无参构造方法 |
| public JFrame(String title)  throws HeadlessException | 构造方法,参数是指定一个标题 |
| public void setBounds(int x,  int y,int width,int height) | 从windows继承而来的,设置窗口出现的位置和大小 |

# 3.GUI组件继承关系图



# 4.Toolkit系统工具包

要用到java.awt包里的系统工具包Toolkit

Toolkit的方法:

返回系统默认工具包对象: public static Toolkit getDefaultToolkit()

获取尺寸对象: public abstract Dimension getScreenSize() throws HeadlessException

**Dimension**有两个方法:

获取宽度: public double getWidth()

获取长度: public double getHeight()

范例:获取屏幕的长宽:

|  |
| --- |
| //获取默认系统工具包  Toolkit toolkit = Toolkit.getDefaultToolkit();  //获取屏幕的分辨率 ,返回尺寸对象  Dimension d = toolkit.getScreenSize();  System.out.println("x轴:" + d.getWidth());  System.out.println("y轴:" + d.getHeight()); |

# 5.对话框

这两个类都是对话框

## 5.1JDialog类

Class JDialog

java.lang.Object

java.awt.Component

java.awt.Container

java.awt.Window

java.awt.Dialog

javax.swing.JDialog

方法:

|  |  |
| --- | --- |
| public JDialog(Dialog owner,  String title,  boolean modal) | 构造函数  owner:对话框的所有者  title:对话框的标题  modal:模式  true:对话框没有关闭时,不允许操作所有者  false:对话框没有关闭时,还是可以操作所有者 |

范例:

|  |
| --- |
| //创建一个窗口  JFrame frame = new JFrame("窗体");  JDialog dialog = new JDialog(frame,"对话框",true);  dialog.setBounds(300,200,100,200);  frame.setVisible(true);  frame.setSize(100,200);  dialog.setVisible(true);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

这个对话框太丑了,不建议使用

## 5.2JOptionPane类

可以弹出不同类型的对话框

java.lang.Object

java.awt.Component

java.awt.Container

javax.swing.JComponent

javax.swing.JOptionPane

方法:

|  |  |
| --- | --- |
| public static void showMessageDialog(Component parentComponent,  Object message,  String title,  int messageType)  throws HeadlessException | parentComponent:对话框的所有者  message:显示的信息  title:对话框的标题  messageType:指定对话框的类型(消息,警告,错误) |
| public static int showConfirmDialog(Component parentComponent,  Object message)  throws HeadlessException | 确认对话框  是:返回0  否:返回1  取消:返回2 |
| public static String showInputDialog(Component parentComponent,  Object message)  throws HeadlessException | 输入对话框,重载此函数有5个 |

范例:各种对话框

|  |
| --- |
| //创建一个窗口  JFrame frame = new JFrame("窗体");  frame.setVisible(true);  frame.setSize(100,200);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  //消息对话框  // JOptionPane.showMessageDialog(frame,"请注意,倒车","通知:",JOptionPane.INFORMATION\_MESSAGE);  //警告对话框  // JOptionPane.showMessageDialog(frame,"明天放假了,爽","通知:",JOptionPane.WARNING\_MESSAGE);  //错误对话框  // JOptionPane.showMessageDialog(frame,"系统奔溃了-.-","通知:",JOptionPane.ERROR\_MESSAGE);  //确认对话框  // int num = JOptionPane.showConfirmDialog(frame,"软件继续安装吗?");  // System.out.println("num:" + num);  //输入对话框  String str = JOptionPane.showInputDialog(frame,"请输入你要取得的金额");  System.out.print(str); |

## 5.3FileDialog类

awt里文件对话框,没有JfileDialog

java.lang.Object

java.awt.Component

java.awt.Container

java.awt.Window

java.awt.Dialog

java.awt.FileDialo

方法:

|  |  |
| --- | --- |
| public FileDialog(Dialog parent,  String title,  int mode) | parent:所有者  title:标题  mode:模式  FileDialog.LOAD 加载  FileDialog.SAVE 保存 |
| public String getDirectory() | 返回文件的所在的路径 |
| public String getFile() | 返回文件名 |

范例:打开文件

|  |
| --- |
| //创建一个窗口  JFrame frame = new JFrame("窗体");  frame.setVisible(true);  frame.setSize(300,200);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  FileDialog dialog = new FileDialog(frame,"文件对话框",FileDialog.LOAD);  dialog.setVisible(true);  System.out.println("文件路径:"+ dialog.getDirectory());  System.out.println("文件名:"+ dialog.getFile()); |

# 6.非容器组件

## 6.1常见的非容器组件

|  |
| --- |
| JFrame frame = new JFrame("注册");  //创建一个面板  JPanel panel = new JPanel();  //创建一个标签对象  JLabel nameLabel = new JLabel("用户名:");  //输入框  JTextField nameField = new JTextField("用户名:",12);  //把组件添加到面板  panel.add(nameLabel);  panel.add(nameField);  //密码  JLabel passLabel = new JLabel("密码:");  JPasswordField passField = new JPasswordField(12);  panel.add(passLabel);  panel.add(passField);  //性别:单选框  //单选框一定要进行分组,在同一组的单选框只能选择其中的一个  JLabel sexLabel = new JLabel("性别:");  JRadioButton man = new JRadioButton("男",true); //单选框的默认值  JRadioButton woman = new JRadioButton("女");  //分组  ButtonGroup group = new ButtonGroup();  group.add(man);  group.add(woman);  panel.add(sexLabel);  panel.add(man);  panel.add(woman);  //城市--下拉框  JLabel cityLabel = new JLabel("来自的城市");  JComboBox<String> cityBox = new JComboBox<>(new String[]{"北京","上海","广州"});  panel.add(cityLabel);  panel.add(cityBox);  //爱好--复选框  JLabel hobitLabel = new JLabel("爱好");  JCheckBox java = new JCheckBox("java");  JCheckBox javascript = new JCheckBox("javascript");  JCheckBox write = new JCheckBox("敲java");  panel.add(hobitLabel);  panel.add(java);  panel.add(javascript);  panel.add(write);  //自我简介--文本域  JLabel introLabel = new JLabel("自我简介");  JTextArea area = new JTextArea(15,15);  panel.add(introLabel);  panel.add(area);  frame.add(panel);  frame.setVisible(true);  frame.setSize(300,300);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

## 6.2菜单组件

菜单条(JMenuBar) 菜单(JMenu) 菜单项(JMenuItem)

关系是:

**菜单条添加菜单,菜单添加菜单项**，注意如果菜单添加菜单,则被添加的菜单为二级菜单

|  |
| --- |
| //窗体  JFrame frame = new JFrame("记事本");  //菜单条  JMenuBar bar = new JMenuBar();  //菜单  JMenu fileMenu = new JMenu("文件");  JMenu editMenu = new JMenu("编辑");  JMenu helpMenu = new JMenu("帮助");//二级菜单  //菜单项  JMenuItem open = new JMenuItem("打开");  JMenuItem save = new JMenuItem("保存");  JMenuItem copy = new JMenuItem("拷贝");  JMenuItem about = new JMenuItem("关于");  JMenuItem version = new JMenuItem("升级");  //二级菜单,菜单添加到菜单上  editMenu.add(helpMenu);  helpMenu.add(about);  helpMenu.add(version);  //文本域  JTextArea area = new JTextArea(20,20);  //把菜单添加到菜单条上  bar.add(fileMenu);  bar.add(editMenu);  //把菜单项添加到菜单上  fileMenu.add(open);  fileMenu.add(save);  fileMenu.add(copy);  //把菜单条添加到窗口上去  frame.add(bar,BorderLayout.NORTH);  frame.add(area);  frame.setVisible(true);  frame.setSize(300,300);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

# 7.布局管理器

作用:用户摆放组件，每种布局管理器都有不同的风格

布局管理器在awt包中

## 7.1边框布局管理器

public class **BorderLayout**

extends Object

implements LayoutManager2, Serializable

Diagram of an applet demonstrating BorderLayout.
      Each section of the BorderLayout contains a Button corresponding to its position in the layout, one of:
      North, West, Center, East, or South.

注意:

1. 如果添加组件时没有指定具体的方位,默认在中间
2. JFrame默认使用的布局管理器就是**BorderLayout**
3. 东南西北哪个组件是缺少的,中间的组件就会占据其他空缺的位置

范例:

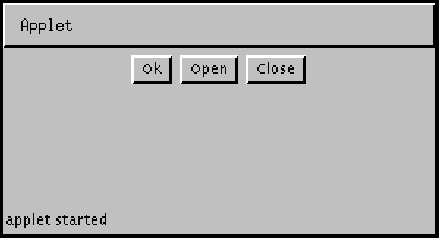
|  |
| --- |
| JFrame frame = new JFrame("边框布局管理器");  //创建边框布局管理器  BorderLayout borderLayout = new BorderLayout();  //让窗体使用边框布局管理器  frame.setLayout(borderLayout);  frame.add(new JButton("北"),BorderLayout.NORTH);  frame.add(new JButton("南"),BorderLayout.SOUTH);  frame.add(new JButton("西"),BorderLayout.WEST);  frame.add(new JButton("东"),BorderLayout.EAST);  frame.add(new JButton("中"),BorderLayout.CENTER);  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

## 7.2流式布局管理器

public class **FlowLayout**

extends Object

implements LayoutManager, Serializable



注意:

1)使用FlowLayout的时候默认是居中的对齐的,在构造函数中能指定对齐方式

public FlowLayout(int align)

2)默认流式布局的中两个水平方向的组件的距离和垂直方向的距离为5个像素,可以通过构造函数来设置

public FlowLayout(int align,int hgap, int vgap)

3)JPanel默认使用的布局管理器就是流式布局管理器

范例:

|  |
| --- |
| JFrame frame = new JFrame("边框布局管理器");  //面板  JPanel panel = new JPanel();  //创建一个流式布局管理器  FlowLayout flowLayout = new FlowLayout(FlowLayout.LEFT,5,10);  //让面板使用流式布局管理器  panel.setLayout(flowLayout);  frame.add(panel);  panel.add(new JButton("按钮1"));  panel.add(new JButton("按钮2"));  panel.add(new JButton("按钮3"));  panel.add(new JButton("按钮4"));  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

## 7.3表格布局管理器

public class **GridLayout**

extends Object

implements LayoutManager, Serializable

Shows 6 buttons in rows of 2. Row 1 shows buttons 1 then 2.
 Row 2 shows buttons 3 then 4. Row 3 shows buttons 5 then 6. Shows 6 buttons in rows of 2. Row 1 shows buttons 2 then 1.
 Row 2 shows buttons 4 then 3. Row 3 shows buttons 6 then 5.

水平,从左到右 水平,从右到左

注意:如果添加的组件个数超过表格的个数,那么会添加多的一列

范例:

|  |
| --- |
| JFrame frame = new JFrame("计算器");  //创建一个表格布局管理器3行2列  GridLayout gridLayout = new GridLayout(3,2);  frame.setLayout(gridLayout);  for(int i = 0;i<6;i++){  frame.add(new JButton(String.valueOf(i)));  }  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

# 8.事件

事件:当某个组件发生了指定的动作时,会有相应的处理方案

事件:

事件源:

事件:

监听器:

处理方式:

## 8.1动作监听器

动作监听器对鼠标和空格键都是起作用的

|  |
| --- |
| ActionListener接口定义:  java.awt.event  Interface ActionListener  接口里只有一个方法:  public void actionPerformed(ActionEvent e) |

|  |
| --- |
| JFrame frame = new JFrame("计算器");  JButton button = new JButton("点我啊");  //给按钮添加一个监听器  //匿名内部类  button.addActionListener(new ActionListener() {  //如果发生了鼠标点击或者按下空格键就会调用actionPerformed方法  @Override  public void actionPerformed(ActionEvent e) { //ActionEvent 事件  JButton button = (JButton)e.getSource();//获取事件源对象  String content = button.getText();  if("点我啊".equals(content)){  button.setText("点他啊");  }else{  button.setText("点我啊");  }  }  });  frame.add(button);  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

## 8.2鼠标监听器

|  |
| --- |
| JFrame frame = new JFrame("计算器");  JButton button = new JButton("点我啊");  //给按钮添加一个鼠标监听器  button.addMouseListener(new MouseListener() {  //鼠标单击  @Override  public void mouseClicked(MouseEvent e) {  System.out.println("鼠标单击");  }  //鼠标按下  @Override  public void mousePressed(MouseEvent e) {  System.out.println("鼠标按下");  }  //鼠标松开  @Override  public void mouseReleased(MouseEvent e) {  System.out.println("鼠标松开");  }  //鼠标进入  @Override  public void mouseEntered(MouseEvent e) {  System.out.println("鼠标进入");  }  //鼠标离开  @Override  public void mouseExited(MouseEvent e) {  System.out.println("鼠标离开");  }  });  frame.add(button);  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

注意:以上代码有个弊端,如果我只是想实现这5个接口中的一个方法,那么就用MouseAdapter(鼠标适配器)

MouseAdapter是一个虚类

public abstract class MouseAdapter

extends Object

implements MouseListener, MouseWheelListener, MouseMotionListener

这个类实现了MouseListener接口,对于接口的每个方法实现的方式的函数体都是空的

范例:

|  |
| --- |
| JFrame frame = new JFrame("计算器");  JButton button = new JButton("点我啊");  //给按钮添加一个鼠标监听器  button.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  System.out.print("啊哈哈哈\n");  }  });  frame.add(button);  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

## 8.3键盘监听器

|  |
| --- |
| JFrame frame = new JFrame("计算器");  JButton button = new JButton("点我啊");  //添加一个键盘监听器  button.addKeyListener(new KeyAdapter() {  @Override  public void keyPressed(KeyEvent e) {  //每个键可能没有对象的字符,但是一定对应有对应的值  System.out.println("按下键的字符:"+ e.getKeyChar() + "键的code:" + e.getKeyCode());  }  });  frame.add(button);  frame.setVisible(true);  frame.setSize(300,400);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |

# 9.画图

画图有两个类可以用,JPanel和Canvas(画布)

## 9.1JPanel

需要使用JPanel类里的一个方法(是继承JComponent类的方法) :

public void paint(Graphics g)

### 9.1.1画矩形

|  |
| --- |
| public class A extends JPanel {  @Override  public void paint(Graphics g) { //Graphics是一个画笔,该画笔可以画图形,也可以写文字  //设置画笔的颜色  g.setColor(Color.RED);  //使用画笔画矩形,最后一个参数控制凹凸感  g.fill3DRect(0,0,50,50,true);  g.fill3DRect(50,0,50,50,true);  }  public static void main(String args[]){  JFrame frame = new JFrame("画笔");  A a = new A();  frame.add(a);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setSize(200,400);  frame.setVisible(true);  }  } |

### 9.1.2写字

|  |
| --- |
| public class A extends JPanel {  @Override  public void paint(Graphics g) { //Graphics是一个画笔,该画笔可以画图形,也可以写文字  //设置画笔的颜色  g.setColor(Color.RED);  //设置画笔的字体  g.setFont(new Font("宋体",Font.BOLD,20));  g.drawString("2017-10-23网盛生意宝",100,100);  }  public static void main(String args[]){  JFrame frame = new JFrame("画笔");  A a = new A();  frame.add(a);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setSize(200,400);  frame.setVisible(true);  }  } |