GUI

awt常用的类有：非菜单组件类，菜单组件类，Graphics类，Font类，Toolkit类，事件类，监听接口。

Component是没有菜单的awt组件的抽象父类。

Awt组件继承图：

Component(Abstract)

Container

其他非容器组件

Lable

Window

Panel

ScrollPane

Dialog

Frame

Awt与swing的区别

1)AWT 是基于本地方法的C/C++程序，其运行速度比较快；Swing是基于AWT的Java程序，其运行速度比较慢。  
       2)AWT的控件在不同的平台可能表现不同，而Swing在所有平台表现一致。

Swing中常用的组件有：

JButton ,JcheckBox,Jlabel,TJextField,TJextArea,JChoice,JRadioButton,JPasswordField

,JComBox,JFileChooser,JTable,JColorChooser.

组件练习1：

**package** cn.com.component;

**import** java.awt.FlowLayout;

**import** javax.swing.ImageIcon;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.SwingConstants;

**public** **class** JFrameTest1 {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**new** MyJFrame();

}

}

**class** MyJFrame **extends** JFrame{

**private** JLabel label;

**public** MyJFrame(){

label = **new** JLabel("文件",**new** ImageIcon("Images/222.png"),SwingConstants.*RIGHT*);//Icoin是装饰组建的小图标

init();

}

**public** **void** init(){

setVisible(**true**);

setLocationRelativeTo(**null**);//窗体在屏幕中央。

setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);//关闭窗口时，JVM已停止运行。

setSize(100,200);

setLayout(**new** FlowLayout(FlowLayout.*LEFT*,10,5));

//pack();

add(label);

setResizable(**false**);

}

}

练习2：

**package** cn.com.component;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.GridLayout;

**import** java.awt.TextField;

**import** java.awt.Toolkit;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

**import** javax.swing.SwingUtilities;

**public** **class** JFrameDemo1 {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable(){

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**new** Frame1();

}

});

}

}

**class** Frame1 **extends** JFrame{

**private** JLabel label1;

**private** JLabel label2;

**private** JTextField text1;

**private** JTextField text2;

**private** JButton button;

**public** Frame1(){

label1= **new** JLabel("firstName");

label2 = **new** JLabel("lastName");

text1 = **new** JTextField();

text2 = **new** JTextField();

button = **new** JButton("regsiter");

LuanchFrame();

}

**public** **void** LuanchFrame(){

setVisible(**true**);

setLayout(**new** GridLayout(3,2));

**int** frameWidth = 200;

**int** frameHeight = 100;

Dimension dimension = Toolkit.*getDefaultToolkit*().getScreenSize();

setBounds((**int**)(dimension.getWidth())-frameWidth,0,frameWidth,frameHeight);

label1.setBackground(Color.*CYAN*);

label2.setBackground(Color.*BLUE*);

text1.setBackground(Color.*yellow*);

text2.setBackground(Color.*yellow*);

button.setBackground(Color.*white*);

add(label1);

add(text1);

add(label2);

add(text2);

add(button);

}

}

练习三：

JTextArea的用法：

Windows窗体的大小为;width:1024,length 768;

**package** cn.com.component;

**import** java.awt.Container;

**import** java.awt.Dimension;

**import** java.awt.GridLayout;

**import** java.awt.ScrollPane;

**import** java.awt.TextArea;

**import** java.awt.Toolkit;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.SwingUtilities;

**public** **class** JTextAreaTest {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable() {

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**new** MyFrame2();

}

});

}

}

**class** MyFrame2 **extends** JFrame{

**private** TextArea textarea1 =**null**;

**private** TextArea textarea2 = **null**;

JScrollPane scroll = **null**;

**public** MyFrame2(){

textarea1 = **new** TextArea(4,10);

textarea2 = **new** TextArea(4,10);

scroll = **new** JScrollPane(textarea1,JScrollPane.*VERTICAL\_SCROLLBAR\_ALWAYS*,JScrollPane.*HORIZONTAL\_SCROLLBAR\_ALWAYS*);

Init();

}

**private** **void** Init() {

// **TODO** Auto-generated method stub

setLayout(**new** GridLayout(1,2));

//setBounds(100, 100, 400, 500);

setBounds(Toolkit.*getDefaultToolkit*().getScreenSize().width/4,Toolkit.*getDefaultToolkit*().getScreenSize().width/4,200,200);

setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);

add(scroll);

//add(textarea1);

add(textarea2);

//textarea1.setPreferredSize(new Dimension(100,100));

setVisible(**true**);

}

}

,练习四：单选按钮

**package** cn.com.component;

**import** java.awt.FlowLayout;

**import** javax.swing.ButtonGroup;

**import** javax.swing.ImageIcon;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JRadioButton;

**import** javax.swing.SwingUtilities;

**public** **class** JRadioButtonTest {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable(){

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

*luanch*();

}

});

}

**public** **static** **void** luanch(){

JFrame frame = **new** JFrame();

JLabel label = **new** JLabel("car:");

JRadioButton radio1 = **new** JRadioButton("宝马");//接文本，才有选择框

JRadioButton radio2 = **new** JRadioButton(**new** ImageIcon("Images/fa.gif"));

JRadioButton radio3 = **new** JRadioButton(**new** ImageIcon("Images/fu.gif"));

radio1.setSelected(**true**);

ButtonGroup group = **new** ButtonGroup();

group.add(radio1);

group.add(radio2);

group.add(radio3);

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

frame.setLocationRelativeTo(**null**);

frame.add(label);

frame.add(radio1);

frame.add(radio2);

frame.add(radio3);

frame.setVisible(**true**);

frame.setLayout(**new** FlowLayout(FlowLayout.*LEFT*));

frame.pack();

}

}

练习五：列表JList.

**package** cn.com.component;

**import** java.awt.FlowLayout;

**import** java.util.Vector;

**import** javax.swing.JFrame;

**import** javax.swing.JList;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingUtilities;

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**public** **class** JListTest {

/\*\*

\* **@param** args

\*/

JFrame frame = **new** JFrame();

JScrollPane scroll ;

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable(){

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**new** JListTest().Luanch();

}

});

}

**public** **void** Luanch() {

// **TODO** Auto-generated method stub

String [] selections = **new** String []{"tennis","tabletennis","football","baseball","basketball"};

// Vector<String> selection = new Vector<String>();

// selection.add("tennis");

// selection.add("tabletennis");

**final** JList list = **new** JList(selections);//设计Jlist对象为常量。

// int [] a = {1,2};

// list.setSelectedIndices(a);

list.setSelectedIndex(0);

list.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION*);//set选项方式是单选还是多选。

list.setFixedCellWidth(200);//列表显示宽度

list.setFixedCellHeight(20);//列表显示的高度

list.setVisibleRowCount(2);//列表可显示的行数

/\*

\* list的监听为：实现addListSelectionListener接口，重写valueChanged方法。

\*/

list.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

// **TODO** Auto-generated method stub

**if**(list.getSelectedValue()=="tennis"){

JOptionPane.*showMessageDialog*(**null**, "选择的球类是网球");

} **else** **if**(list.getSelectedValue()=="tabletennis"){

JOptionPane.*showMessageDialog*(**null**, "选择的球类是乒乓球", "ballinformation", JOptionPane.*INFORMATION\_MESSAGE*);

} **else** **if**(list.getSelectedValue()=="football"){

JOptionPane.*showMessageDialog*(**null**, "选择的球类是足球");

} **else** **if**(list.getSelectedValue()=="baseball"){

JOptionPane.*showMessageDialog*(**null**, "选择的球类是棒球");

} **else** **if**(list.getSelectedValue()=="basketball"){

JOptionPane.*showMessageDialog*(**null**, "选择的球类是篮球");

}

}

});

scroll = **new** JScrollPane(list);

System.*out*.println(list.getSelectedValue());

frame.setLayout(**new** FlowLayout());

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

frame.add(scroll);

frame.pack();

frame.setVisible(**true**);

}

}

练习六：

下拉列表JComboBox：

**package** cn.com.component;

**import** java.awt.FlowLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ItemEvent;

**import** java.awt.event.ItemListener;

**import** javax.swing.JComboBox;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.SwingUtilities;

**public** **class** JComboBoxTest {

**private** **static** JComboBox *box* = **null**;

**private** **static** JFrame *f* = **null**;

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable(){

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

*Luanch*();

}

});

}

**private** **static** **void** Luanch() {

// **TODO** Auto-generated method stub

*f* = **new** JFrame();

//f.setLayout(new FlowLayout());

String [] seletions = {"red","blue","black"};

*box* = **new** JComboBox(seletions);

*box*.insertItemAt("yellow", 1);//将指定选项插到指定的目录

*box*.addItem("gray");//添加指定的选项。

System.*out*.println(*box*.getItemAt(2));//返回指定下标的选项。

System.*out*.println(*box*.getSelectedIndex());//返回已选选项的下标

System.*out*.println(*box*.getSelectedItem());

System.*out*.println(*box*.getItemCount());//返回所有选项的数目。

*box*.removeItem("blue");//删除指定项目。

/\*

\* 监听原理：实现ITeemLister接口，重写itemstateChanged方法。

\*/

*box*.addItemListener(**new** ItemListener() {

@Override

**public** **void** itemStateChanged(ItemEvent e) {

// **TODO** Auto-generated method stub

**if**(e.getItem().equals("yellow")&&e.getStateChange()==ItemEvent.*SELECTED*){

JOptionPane.*showMessageDialog*(**null**, "已选的颜色为黄色");

} **else** **if**(e.getItem().equals("black")&&e.getStateChange()==ItemEvent.*SELECTED*){

JOptionPane.*showMessageDialog*(**null**, "已选的颜色为黑色");

} **else** **if**(e.getItem().equals("red")&&e.getStateChange()==ItemEvent.*SELECTED*){

JOptionPane.*showMessageDialog*(**null**, "已选的颜色为红色");

} **else** **if**(e.getItem().equals("gray")&&e.getStateChange()==ItemEvent.*SELECTED*){

JOptionPane.*showMessageDialog*(**null**, "已选的颜色为灰色");

}

}

});

/\*动作监听。

\*/

// box.addActionListener(new ActionListener() {

//

// @Override

// public void actionPerformed(ActionEvent e) {

// // **TODO** Auto-generated method stub

// Object temp = e.getSource();

// if(temp.equals(JComboBoxTest.this.box)){

// if(JComboBoxTest.this.box.getSelectedItem()=="yellow"){

// JOptionPane.showMessageDialog(null, "已选的颜色为黄色");

// } else if(JComboBoxTest.this.box.getSelectedItem()=="red"){

// JOptionPane.showMessageDialog(null, "已选的颜色红色");

// } else if(JComboBoxTest.this.box.getSelectedItem()=="black"){

// JOptionPane.showMessageDialog(null, "已选的颜色黑色");

// } else if(JComboBoxTest.this.box.getSelectedItem()=="gary"){

// JOptionPane.showMessageDialog(null, "已选的颜色灰色");

// }

// }

// }

// });

*f*.add(*box*);

*box*.setSelectedIndex(1);

System.*out*.println(*box*.getSelectedItem());

*f*.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

*f*.setSize(400,200);

*f*.setLayout(**new** FlowLayout());

*f*.setVisible(**true**);

}

}

练习7：下拉式菜单Choice：

**package** cn.com.component;

**import** java.awt.Choice;

**import** java.awt.Dialog;

**import** java.awt.TextArea;

**import** java.awt.event.KeyAdapter;

**import** javax.swing.JFrame;

**public** **class** ChoiceTest {

/\*\*

\* **@param** args

\* 下拉式菜单。

\*/

**static** JFrame *choiceFrame* = **new** JFrame("choiceframe");

Dialog dialog = **new** Dialog(**new** InfFrame());

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Choice choice =**new** Choice();

choice.add("open");

choice.add("sava");

choice.add("exit");

choice.select(1);

choice.addMouseListener(**new** MyMouseAdapter());

*choiceFrame*.add(choice);

*choiceFrame*.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

*choiceFrame*.setSize(100,200);

*choiceFrame*.setVisible(**true**);

}

}

**class** InfFrame **extends** JFrame{

**private** TextArea ta;

**public** InfFrame(){

ta = **new** TextArea();

init();

}

**private** **void** init() {

// **TODO** Auto-generated method stub

**this**.setSize(100,100);

**this**.add(ta);

ta.setText("你选择了sava操作");

**this**.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

**this**.setVisible(**true**);

}

}

**package** cn.com.component;

**import** java.awt.Choice;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**public** **class** MyMouseAdapter **extends** MouseAdapter {

@Override

**public** **void** mousePressed(MouseEvent e) {

// **TODO** Auto-generated method stub

Choice choice = (Choice) e.getSource();

**if**(choice.getSelectedItem()=="sava"){

**new** ChoiceTest().dialog.setVisible(**true**);

**new** ChoiceTest().*choiceFrame*.setVisible(**false**);

}

}

}

8.JOptionpane容器的用法

**package** cn.com.component;

**import** java.awt.FlowLayout;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.SwingUtilities;

/\*

\* 显示用户信息用showMessageDialog（）来显示信息。

\*/

**public** **class** JOptionPaneTest {

**public** **static** **void** main(String[] args) {

SwingUtilities.*invokeLater*(**new** Runnable(){

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

*Luanch*();

}

});

}

**public** **static** **void** Luanch(){

JFrame frame = **new** JFrame();

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

//JOptionPane.showMessageDialog(null, "thanks");

//JOptionPane.showMessageDialog(null,"thank you Welcome to us store","thanks",JOptionPane.INFORMATION\_MESSAGE);

// JOptionPane.showMessageDialog(null, "you are not save the file","sure",JOptionPane.WARNING\_MESSAGE);

//showConfirm();

*showInput*();

frame.setLayout(**new** FlowLayout());

frame.pack();

frame.setVisible(**true**);

}

**public** **static** **void** showMessage(){

//JDialog.setDefaultLookAndFeelDecorated(true);

JOptionPane.*showMessageDialog*(**null**, "you first name is not values","sure",JOptionPane.*ERROR\_MESSAGE*);

}

**public** **static** **void** showConfirm(){//请求用户确认用 showConfirmDialog()方法，JOptionPane中方法一般是静态方法。

JDialog.*setDefaultLookAndFeelDecorated*(**true**);

**int** selections = JOptionPane.*showConfirmDialog*(**null**, "Do you want to continue","sure",JOptionPane.*YES\_NO\_CANCEL\_OPTION*);

**if**(selections==JOptionPane.*OK\_OPTION*){

System.*out*.println("come on ");

} **else** **if**(selections==JOptionPane.*NO\_OPTION*){

System.*out*.println("no");

} **else** **if**(selections==JOptionPane.*CLOSED\_OPTION*){

System.*out*.println("close");

} **else** **if**(selections==JOptionPane.*CANCEL\_OPTION*){

System.*out*.println("cancel");

}

}

**public** **static** **void** showInput(){//获得用户输入用showinputDialog();

// Object [] selections ={"dog","cat","horse"};

// String initselection = "dog";

// String selection = (String) JOptionPane.showInputDialog(null, "yyyy");

// String selection = (String) JOptionPane.showInputDialog("yyyy");

// String selection = (String) JOptionPane.showInputDialog(null, "please input a aminal","anmianl",JOptionPane.INFORMATION\_MESSAGE);

// String selection = (String) JOptionPane.showInputDialog(null, "please input a aminal","a");

String selection = (String) JOptionPane.*showInputDialog*(**null**, "please input a aminal","anmianl",JOptionPane.*QUESTION\_MESSAGE*,**null**,**new** String [] {"dog","animal"},"dog");

System.*out*.println(selection);

}

}

练习9：菜单的用法：

**package** cn.com.component;

**import** java.awt.event.KeyEvent;

**import** javax.swing.JFrame;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JPanel;

**import** javax.swing.KeyStroke;

**public** **class** JMenuFrame **extends** JFrame{

**public** JMenuFrame(){

**this**.add(**new** MyMenuPanel());

**this**.setSize(300,200);

**this**.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

**this**.setVisible(**true**);

}

**public** **static** **void** main(String[] args) {

**new** JMenuFrame();

}

}

**class** MyMenuPanel **extends** JPanel{

**private** JMenuBar menuBar;

**private** JMenu fileMenu;

**private** JMenu editMenu;

**private** JMenu newMenu;

**private** JMenuItem newItem;

**private** JMenuItem newFileItem;

**private** JMenuItem newFolderItem;

**private** JMenuItem openItem;

**private** JMenuItem saveItem;

**private** JMenuItem printItem;

**private** JMenuItem exitItem;

**private** JMenuItem cancleItem;

**private** JMenuItem copyItem;

**private** JMenuItem pasteItem;

**private** JMenuItem allItem;

**public** MyMenuPanel(){

menuBar = **new** JMenuBar();

fileMenu = **new** JMenu("文件");

editMenu = **new** JMenu("编辑");

newMenu = **new** JMenu("New");

newItem = **new** JMenuItem("新建");

openItem = **new** JMenuItem("打开");

saveItem = **new** JMenuItem("保存");

printItem = **new** JMenuItem("打印");

exitItem = **new** JMenuItem("退出");

cancleItem = **new** JMenuItem("撤销");

copyItem = **new** JMenuItem("复制");

pasteItem = **new** JMenuItem("粘贴");

allItem = **new** JMenuItem("全选");

newFileItem = **new** JMenuItem("新建文件");

newFolderItem = **new** JMenuItem("新建文件夹");

init();

}

**private** **void** init() {

// **TODO** Auto-generated method stub

menuBar.add(fileMenu);

menuBar.add(editMenu);

fileMenu.add(newMenu);

// newMenu.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_N,KeyEvent.CTRL\_MASK));

newMenu.setMnemonic('N');

fileMenu.add(newMenu);

fileMenu.setMnemonic('F');

newMenu.add(newFileItem);

newFileItem.setMnemonic('f');

newFileItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_F*,KeyEvent.*SHIFT\_MASK*));

newMenu.add(newFolderItem);

newFolderItem.setMnemonic('f');

//设计快捷键

newFolderItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_F*,KeyEvent.*ALT\_MASK*));

fileMenu.add(openItem);

//设计记忆符号

openItem.setMnemonic('o');

openItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_O*,KeyEvent.*CTRL\_MASK*));

fileMenu.add(saveItem);

saveItem.setMnemonic('s');

saveItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_S*,KeyEvent.*CTRL\_MASK*));

fileMenu.addSeparator();

fileMenu.add(printItem);

printItem.setMnemonic('p');

printItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_P*,KeyEvent.*CTRL\_MASK*));

fileMenu.addSeparator();

fileMenu.add(exitItem);

exitItem.setMnemonic('e');

exitItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_E*,KeyEvent.*CTRL\_MASK*));

editMenu.setMnemonic('E');

editMenu.add(cancleItem);

cancleItem.setMnemonic('c');

openItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_C*,KeyEvent.*CTRL\_MASK*));

editMenu.add(copyItem);

copyItem.setMnemonic('c');

copyItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_C*,KeyEvent.*CTRL\_MASK*));

editMenu.add(pasteItem);

pasteItem.setMnemonic('p');

pasteItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_P*,KeyEvent.*CTRL\_MASK*));

editMenu.add(allItem);

allItem.setMnemonic('a');

allItem.setAccelerator(KeyStroke.*getKeyStroke*(KeyEvent.*VK\_A*,KeyEvent.*CTRL\_MASK*));

**this**.add(menuBar);

}

}

\*事件监听原理描述如下：

1. 将监听器注册到事件源对象上。
2. 由于外部动作使得事件源产生事件。
3. 将事件传送到监听器。
4. 监听器通过方法对事件进行处理。

监听图如下：

监听器

注册

监听器处理事件

将对象传送到监听器

事件源对象

产生对象