

# java GUI

---

- 这是什么
- 他怎么玩
- 该如何去在我们平时运用?
- 组件
  - 窗口
  - 弹窗
  - 面板
  - 文本框
  - 列表框
  - 按钮
  - 图片
  - 监听事件
  - 鼠标
  - 键盘事件
  - 破解工具

做完就可以用来做外挂

## 简介

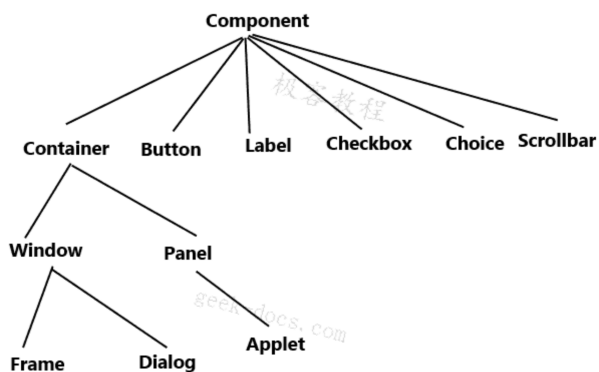
Gui的核心技术: Swing AWT 是GUI的核心编程工具, 现在不流行, 因为不美观, 还需要我们的jre环境! 太大了!

为什么我们要学习, 因为他是MVC的基础, 可以写出自己心中想要的小工具, 工作的时候也可能维护到swing界面, 概率极小, 主要是为了了解MVC架构, 了解监听!

## AWT

abstract windows toolkit

包含了很多类和接口



## 创建窗口

java.awt.Frame 类

属于 `Container`

`Frame` 是顶级窗口（是一个独立的窗口，它不是其他窗口的子窗口。顶级窗口可以独立存在，不依赖于其他窗口。它有自己的标题和边框，可以被用户直接看到和操作）

### Constructor

1. 空参构造
2. `Frame(String title)`

`title` 为窗口名

### Method

1. `setBounds()`
2. `setBackground()`
3. `setResizable()`
4. `setLayout()`
5. `addWindowListener()`
6. `add(Component component)`
7. `setVisible()`

### 注意

1. 创建一个窗口的时候，要调用 `setResizable()`、`setLayout()`、`setBounds()`、`setBackground()`、`addWindowEvent()`、`setVisible()`
2. `setLayout()` 影响的是容器里面的东西，而不是容器本身
3. `setVisible()` 方法的使用顺序确实很重要：
  - **在添加所有组件后调用**：`setVisible(true)` 会导致窗口及其所有子组件被绘制。如果在调用 `setVisible(true)` 后添加组件，可能需要额外的步骤（如调用 `validate()` 和 `repaint()`）来确保新添加的组件被正确绘制。
  - **在设置布局 and 大小后调用**：布局管理器和窗口大小会影响组件的位置和大小。
  - **在 `setUndecorated()` 之前调用**：当使用 `setUndecorated()` 方法来移除窗口的装饰（如标题栏），在调用 `setVisible(true)` 之前调用 `setUndecorated(true)`。一旦窗口变为可见，就不能更改其装饰状态。
  - **在 `dispose()` 之后调用**：调用了 `dispose()` 来释放窗口的资源，通过调用 `setVisible(true)` 来重新创建窗口和其资源。
4. `setBounds()` 等效于 `setLocation()` + `setSize()`;
5. 要使用 `setBounds()`，请将 `frame.setLayout(null)`，如果您使用各种布局，则 `setBounds()` 将不起作用

### 布局管理器

#### FlowLayout

```
Frame frame = new Frame();
frame.setLayout(new FlowLayout(FlowLayout.***));
```

## BorderLayout

```
Frame frame = new Frame();
frame.setLayout(new BorderLayout());

frame.add(component, BorderLayout.**);
```

## GridLayout

```
Frame frame = new Frame();
frame.setLayout(new GridLayout(int row, int column));

frame.add(component)
frame.add(component)
frame.add(component)
```

### 注意事项

1. 给一个容器设置了布局之后，就不能使用 `setBounds()`，`setLocation()`，`setSize()`
2. 使用 `BorderLayout` 要在容器调用 `add()` 方法的时候使用
3. `BorderLayout`，如果只给了一个 `NORTH`，那就就是 `NORTH` 会被沾满，但是别的地方是没被使用的，就是是一个大块的状态，如果要使用也只能通过 `BorderLayout`
4. 使用 `GridLayout` 要在 `new GridLayout` 的时候设置好行列数，添加的时候就不用指定在哪行哪列，默认从左到右，从上到下添加
5. 所以一个容器如果被设置了 `BorderLayout`，最多只能往里面放五个组件吗？

不完全是这样的。虽然 `BorderLayout` 将容器划分为五个区域（北、南、东、西和中心），并且每个区域默认只能添加一个组件，但是您可以通过在一个区域中添加一个容器（如 `Panel` 或 `JPanel`），然后在这个容器中添加多个组件，从而在一个区域中放置多个组件。

## 创建面板

`java.awt.Pannel` 类

面板是内嵌在窗口的一个东西，所以是 `Frame` 对象.`add(Pannel对象)`

面板的坐标是相对于 `Frame` 对象的坐标

### 注意事项

1. `panel` 是中间层窗口，不能独立存在，必须添加到其他窗口使用
2. 记得给 `panel` 设置一个颜色，不然就是 `frame` 的颜色，这样你看不出来你的 `panel` 在哪
3. `setBounds()` 的坐标相对的包括了程序的边框大小。非常非常重要的一点！！！！你用 `setBounds` 手动调坐标的时候，图形界面的顶部边框的宽度也是包括在坐标内的，也就是给 `frame` 加容器的时候，记得 `y` 坐标给加上边框的宽度，因为相对坐标也是从最右上角的点算的。

另外一点就是，当你觉得你的代码，布局啥的都 ok 的时候，还是得不到想要的结果的时候，可以慢慢的调一调组件的位置和大小（建议把组件调到刚好可以显示标签的大小，就可以去动位置，看看是不是位置的问题了）

## 布局管理器

有三种布局管理器：FlowLayout、BorderLayout、GridLayout

使用：

### 1.FlowLayout

```
Frame frame = new Frame();  
  
frame.se
```

一个frame设置了布局管理器，和一个panel设置了布局管理器，会产生冲突是吗？那正确的做法是什么呢？

## 按钮

java.awt.Button类

要添加多个按钮，先设置好Frame对象的布局（有三种布局）

```
frame.setLayout();
```

里面需要传入一个layout：流式布局：FlowLayout；东西南北中布局：BorderLayout；表格布局：GridLayout

用上面这几个类，然后你在一个软件的界面里面看到的是多种布局的嵌套使用。

然后再add

## 事件监听

```
button.addActionListener
```

## 文本框监听

文本框：java.awt.TextField

文本框监听：textField.addActionListener()

addActionListener()里面传入的对象实现的方法是：e.getSource()，然后强转型，操作textField对象

练习：简易计算器，并且分别使用组合类和内部类进行代码优化

有个Label类，也可以add到frame里面去

## 画笔

`java.awt.Frame.paint()`方法，这个方法需要传入一个`Graphics`对象，是用这个对象来实现各种切换颜色什么的。

## 鼠标监听

适配器模式：因为你想用一个接口就得重写他里面的所有方法，这时候会有一个这个接口的适配器类（就是个类，只不过我们叫他适配器），一般是接口名称后面加上一个`Adapter`就是了，这个类实现了接口里面的所有方法，但是实现的是空方法，但是你继承这个类的时候就不用重写全部方法了，你可以只重写你需要的方法就行了。

鼠标监听事件：因为鼠标（比如单击，英文叫`press`）的对象是窗口（如果是针对画图工具的话），那么就是 `e.getSource()`返回的是`Frame`对象，可以得到你点击的`x, y`坐标

用`Point`类把`x, y`坐标转化为点

那么怎么存点呢？存东西有集合类和`Map`类，`map`感觉更像键值对，就是你要用的话你得知道键，但你都存起来了，怎么知道呢？那就是集合，但我们的点有`x`，和`y`两个属性啊，一个一个存在集合不会乱吗？不会，还记得泛型吗？/偷笑，对，集合里面存`Point`对象不就好了！！

然后去用`frame.paint`方法，把这个点画出来。

然后不能只画一次，要每次都刷新，再调用`frame.repaint()`方法

问题：`paint`方法怎么没调用就能画东西，好好去看看`paint`方法的文档。

## 窗口监听事件

去看`WindowAdapter`的源码，有几个方法

## 键盘监听事件

用`Frame`里面的`addKeyListener`方法，可以自己写类继承`Frame`去用这个方法，也可以直接对`frame`对象使用。

传入`KeyAdapter`对象，有一个方法，`keyPressed`，需要传入一个`KeyEvent` 对象，但是似乎不要我们传入，去看看源码是怎么回事？

## Swing

---

### 窗口、面板

`JFrame` 类

窗口关闭事件被封装了：`setDefaultCloseOperation(WindowConstants.里面有些参数)`

`Container`

**与 `awt` 不同的地方：**`frame` 设置颜色设置的是 `Container` 的颜色，所以通过 `JFrame.getContentPane()` 得到容器，用 `Container` 对象接收

然后给 container 设置颜色就好了。

然后，要在 JFrame 里面放东西，都是放在 Container 里面，如 button, label 等等

## 绝对定位

绝对定位：给定坐标之后是定死的，你给窗口缩放，如 button 的大小是不会发生变化的（Frame 的时候是会跟着一起缩放的）

## Dialog 对话框（弹窗）

明确一点的是：对话框也是一个 Frame 或者说 JFrame

但是弹窗用的是 JDialog 类了，不是 JFrame，可以去源码看看 JDialog 类是个什么情况，其实他里面用的东西都是 JFrame 的东西。

### 注意

1. JDialog 里面自动封装了 setDefaultCloseOperation(WindowConstants.里面有些参数) 方法，你就不用给它加了。

## JLabel

JLabel 对应 awt 里面的 Label 类

给 label 居中，用 Label 的 setHorizontalAlignment() 方法

里面传入的是一个 int 类型，可以用 SwingConstants.各种方式

```
JLabel label = new JLabel("");
```

## Icon 图标

Icon 可以放在 Label 上，也可以放在 JButton 上

还有 ImageIcon，把图片作为图标

然后要把 icon 放在标签上，可以用 Label 的构造，也可以用 Label.setIcon 方法。

**注意** 通过类可以获得这个类同级目录下的东西：`Class.class.getResource("文件名")`

## JPanel 面板

用 JPanel 类

有个可以传 Layout 的构造，这里我们可以传入一个 GridLayout。

panel 的 add 方法

然后一个 panel 对象可以 add 多少个东西是有 GridLayout 设定的布局数决定的。

### 注意

1. JFrame 返回的 Container 可以通过 GridLayout 传入一个水平间距和垂直间距的参数

## JScrollPane 滚动条

它的一个构造就是把一个组件放进去，就可以对其实现滚动条功能，如把 JTextArea 对象放进去，当输入的内容超出了文本框的大小，就会出现滚动条。

## 按钮

### 普通按钮

JButton, 他有一些方法可以去看看。

JButton.setToolTipText() 就是你鼠标悬浮在按钮上会出现的文字提示。

### 单选按钮

JRadioButton(), 在构造内可以给按钮的文本

可以 new 多个对象，然后对多个对象进行分组之后，然后同一组里面的按钮，只能选中一个按钮。

那么就有一个组对象， ButtonGroup, 使用 add 方法把组员加进来

### 注意

1. 最后使用 container add组件的时候不用把 ButtonGroup 对象add, add JRadioButton 对象就行了

### 复选按钮

JCheckBox

直接new 出多个JCheckBox对象，把这些按钮add到container里面去，就是一组复选按钮了，不用分组。

## 列表

### 下拉框

JComboBox

用里面的 addItem 方法来加选项

### 列表框

JList

可以在构造的时候传入一个 String[]

直接就得到了这个 String[] 表示的列表

## 文本框

### 文本框

TextField

密码框

PasswordField

文本域

TextArea

## 小结

---

以上这些内容对你理解后面的东西是有帮助的，以后这块的内容会搬到前端