

1. 概述

官方JavaDocsApi: [javax.swing.JFrame](#)

JFrame，窗口。JFrame 是一个可以独立显示的组件，一个窗口通常包含有标题、图标、操作按钮(关闭、最小化、最大化)，还可以为窗口添加菜单栏、工具栏等。一个进程中可以创建多个窗口，并可在适当时候进行显示、隐藏 或 销毁。

JFrame 常用构造方法:

```
1  /**
2   * 参数说明:
3   * title: 窗口标题
4   */
5  JFrame() JFrame(String title)
```

JFrame 常用方法:

```
1  // 设置窗口的 标题
2  void setTitle(String title)
3  // 设置窗口的 图标
4  void setIconImage(Image image)
5  // 设置窗口的 宽高
6  void setSize(int width,int height)
7  void setSize(Dimension d)
8  /**
9   * 设置窗口关闭按钮点击后的默认操作，参考值:
10   * WindowConstants.DO_NOTHING_ON_CLOSE: 不执行任何操作。
11   * WindowConstants.HIDE_ON_CLOSE: 隐藏窗口（不会结束进程），再次调用 setVisible(true) 将再次显示。
12   * WindowConstants.DISPOSE_ON_CLOSE: 销毁窗口，如果所有可显示的窗口都被 DISPOSE，则可能会自动结束进程。
13   * WindowConstants.EXIT_ON_CLOSE: 退出进程。
14   */
15  void setDefaultCloseOperation(int operation
16 )// 设置窗口是否可放大缩小
17  void setResizable(boolean resizable)
18  // 设置窗口的 位置（相对于屏幕左上角）
19  void setLocation(int x,int y)
20  void setLocation(Point p)
21  // 设置窗口的 位置 和 宽高
```

```

22 void setBounds(int x,int y,int width,int height)
23 void setBounds(Rectangle rect)
24 // 获取窗口的位置坐标（相对于屏幕坐标空间）
25 Point getLocationOnScreen()
26 // 获取窗口的位置坐标（相对于父级坐标空间，窗口的父级一般就是屏幕）
27 Point getLocation()
28 /**
29  * 设置窗口的相对位置。
30  * 如果 comp 整个显示区域在屏幕内，则将窗口放置到 comp 的中心；
31  * 如果 comp 显示区域有部分不在屏幕内，则将该窗口放置在最接近 comp 中心的一侧；
32  * comp 为 null，表示将窗口放置到屏幕中心。
33  */
34 void setLocationRelativeTo(Component comp)
35 // 设置将窗口 置顶 显示
36 void setAlwaysOnTop(boolean alwaysOnTop)
37 // 设置窗口的内容面板
38 void setContentPane(Container contentPane)
39 // 设置窗口是否可见，窗口对象刚创建和添加相应组件后通过 setVisible(true) 绘制
    窗口，其内部组件可能要此时才有宽高值
40 void setVisible(boolean b)
41 // 判断窗口是否处于显示状态
42 boolean isShowing()
43 // 销毁窗口，释放窗口及其所有子组件占用的资源，之后再次调用 setVisible(true)
    将会重构窗口
44 void dispose()
45 // 调整窗口的大小，以适合其子组件的首选大小和布局。
46 void pack()

```

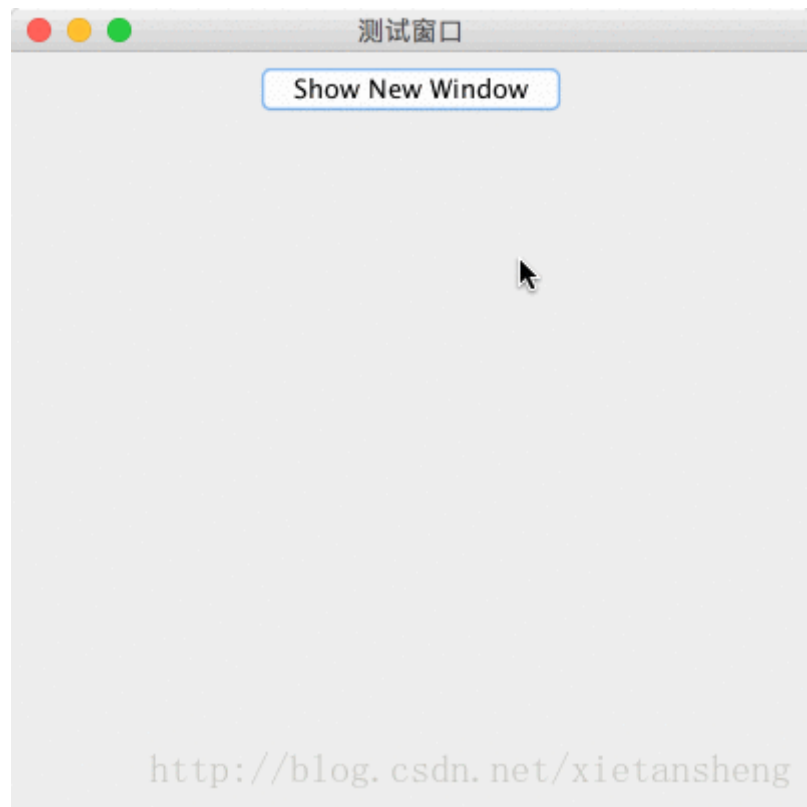
2. 代码实例

```

1 package com.xiets.swing;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 public class Main{
7     public static void main(String[] args){
8         final JFrame jf =new JFrame("测试窗口");
9         jf.setSize(400,400);
10        jf.setLocationRelativeTo(null);

```

```
11  jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
12  JPanel panel =new JPanel();
13  JButton btn =new JButton("Show New Window");
14  btn.addActionListener(new ActionListener(){
15      @Override
16      public void actionPerformed(ActionEvent e){
17          // 点击按钮，显示新的一个窗口
18          showNewWindow(jf);
19      }
20  });
21  panel.add(btn);
22  jf.setContentPane(panel);
23  jf.setVisible(true);
24  }
25  public static void showNewWindow(JFrame relativeWindow){
26      // 创建一个新窗口
27      JFrame newJFrame =newJFrame("新的窗口");
28      newJFrame.setSize(250,250);
29      // 把新窗口的位置设置到 relativeWindow 窗口的中心
30      newJFrame.setLocationRelativeTo(relativeWindow);
31      // 点击窗口关闭按钮，执行销毁窗口操作（如果设置为 EXIT_ON_CLOSE，则点击新窗口关闭按钮后，整个进程将结束）
32      newJFrame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
33      // 窗口设置为不可改变大小
34      newJFrame.setResizable(false);
35      JPanel panel =new JPanel(newGridLayout(1,1));
36      // 在新窗口中显示一个标签
37      JLabel label =new JLabel("这是一个窗口");
38      label.setFont(newFont(null, Font.PLAIN,25));
39      label.setHorizontalAlignment(SwingConstants.CENTER);
40      label.setVerticalAlignment(SwingConstants.CENTER);
41      panel.add(label);
42      newJFrame.setContentPane(panel);
43      newJFrame.setVisible(true);
44  }
45  }
```



result.gif