

1. 概述

官方JavaDocsApi:

- [javax.swing.SpringLayout](#)
- [javax.swing.SpringLayout.Constraints](#)
- [javax.swing.Spring](#)

SpringLayout，弹性布局管理器。使用该布局的容器内的 **每一个组件或容器** 都对应着一个约束，通过该约束定义组件或容器四条边的坐标位置 来实现对组件的布局。该布局主要涉及三个类: **SpringLayout**、**SpringLayout.Constraints**、**Spring**，分别表示 *布局管理器*、*约束*、*坐标距离*。

(1) SpringLayout

表示一个布局管理器，通过该布局管理器可以获取组件或容器的约束对象，如下:

```
1 // 创建 弹性布局管理器 和 容器，容器 使用 弹性布局
2 SpringLayout layout =newSpringLayout();
3 JPanel panel =newJPanel(layout);
4 // 创建 按钮组件，并添加到 容器J
5 Button btn =newJButton("Button");
6 panel.add(btn);
7 // 获取 按钮组件 的 约束对象（如果没有，会自动创建）
8 SpringLayout.Constraints btnCons = layout.getConstraints(btn);
9 // 获取 容器组件 的 约束对象（如果没有，会自动创建）
10 SpringLayout.Constraints panelCons = layout.getConstraints(panel);
```

(2) SpringLayout.Constraints

表示对 **组件位置和尺寸的约束**，每个组件实例都对应着一个约束对象，通过该约束可以 *获取和设置* 组件四条边的 *坐标和宽高*，常用方法如下:

```
1 /** 组件左上角坐标的设置（这里 X 相当于 WEST，Y 相当于 NORTH） */
2 void setX(Spring x)
3 void setY(Spring y)
4 Spring getX()
5 Spring getY()
6 /** 组件宽高的设置 */
7 void setWidth(Spring w)
8 void setHeight(Spring h)
```

```

9  Spring getHeight()
10 Spring getWidth()
11 /** 组件指定边的的坐标或长度设置
12  * edgeName 的值为如下常量之一:
13  * SpringLayout.NORTH, SpringLayout.SOUTH, SpringLayout.EAST, SpringLayout.WEST
14  * SpringLayout.HORIZONTAL_CENTER, SpringLayout.VERTICAL_CENTER
15  * SpringLayout.BASELINE
16  * SpringLayout.WIDTH, SpringLayout.HEIGHT
17  */
18 Spring getConstraint(String edgeName)
19 void setConstraint(String edgeName, Spring s)

```

(3) Spring

Spring 可以看做是在 SpringLayout 中对 **距离的度量**。一个 Spring 实例，表示一段距离或长度，并且支持简单的算术运算（通过 Spring 提供的静态方法进行计算），常用方法如下：

```

1  // 创建一个指定长度的 spring
2  static Spring constant(int pref)
3  // 两个 spring 相加，得到新的一个 spring
4  static Spring sum(Spring s1, Spring s2)
5  // 计算两个 spring 之间的较大者
6  static Spring max(Spring s1, Spring s2)
7  // 对 spring 的缩放
8  static Spring scale(Spring s,float factor)
9  // 计算指定组件的宽度所表示的 spring
10 static Spring width(Component c)
11 // 计算指定组件的高度所表示的 spring
12 static Spring height(Component c)
13 // 对 spring 所表示的数值大小的获取和设置
14 int getValue()
15 void setValue(int value)

```

2. 代码实例

```

1  package com.xiets.swing;
2  import javax.swing.*.*;
3  public class Main{
4  public static voidmain(String[] args){

```

```
5 // 创建窗口
6 JFrame jf =newJFrame("测试窗口");
7 jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
8 jf.setSize(300,200);
9 jf.setLocationRelativeTo(null);
10 // 创建内容面板, 使用 弹性布局
11 SpringLayout layout =newSpringLayout();
12 JPanel panel =newJPanel(layout);
13 jf.setContentPane(panel);
14 // 创建组件
15 JLabel label =newJLabel("Test JLabel: ");
16 JButton btn =newJButton("Btn");
17 JTextField textField =newJTextField("Text Field");
18 // 添加组件到内容面板
19 panel.add(label);
20 panel.add(btn);
21 panel.add(textField);
22 /** 组件的约束设置 (弹性布局设置的关键) */
23 // 标签组件约束: 设置标签的左上角坐标为 (5, 5)
24 SpringLayout.Constraints labelCons = layout.getConstraints(label);
25 // 从布局中获取指定组件的约束对象 (如果没有, 会自动创建)
26 labelCons.setX(Spring.constant(5));
27 labelCons.setY(Spring.constant(5));
28 // 按钮组件约束: 设置左上角 水平坐标为5, 垂直坐标为 标签的南边坐标; 设置东边坐
    标为 标签的东边坐标
29 SpringLayout.Constraints btnCons = layout.getConstraints(btn);
30 btnCons.setX(Spring.constant(5));
31 btnCons.setY(labelCons.getConstraint(SpringLayout.SOUTH));
32 btnCons.setConstraint(SpringLayout.EAST, labelCons.getConstraint(Spring
    Layout.EAST));
33 // 文本框约束: 设置左上角 水平坐标为 标签的东边坐标 + 5, 垂直坐标为 5
34 SpringLayout.Constraints textFieldCons = layout.getConstraints(textFiel
    d);
35 textFieldCons.setX(Spring.sum(labelCons.getConstraint(SpringLayout.EAST),Sp
    ring.constant(5)));
36 textFieldCons.setY(Spring.constant(5));
37 /** 内容面板 (容器) 的约束设置, 即确定 组件 和 容器的右边和底边 之间的间隙大小
    */
38 SpringLayout.Constraints panelCons = layout.getConstraints(panel);
39 // 获取容器的约束对象
40 // 设置容器的 东边坐标 为 文本框的东边坐标 + 5
```

```
41 panelCons.setConstraint(SpringLayout.EAST, Spring.sum(textFieldCons.getConstraint(SpringLayout.EAST), Spring.constant(5)));
42 // 计算出 按钮 和 文本框 的 南边坐标 的 值较大者
43 Spring maxHeightSpring =
Spring.max(btnCons.getConstraint(SpringLayout.SOUTH), textFieldCons.getConstraint(SpringLayout.SOUTH));
44 // 设置容器的 南边坐标 为 maxHeightSpring + 5
45 panelCons.setConstraint(SpringLayout.SOUTH, Spring.sum(maxHeightSpring, Spring.constant(5)));
46 // 显示窗口
47 jf.setVisible(true);
48 }
49 }
50
```

结果展示:



result.png

SpringLayout 更详细具体的介绍和使用方式, 详见 Java 官方文档: [How to Use SpringLayout](#)