GLOBAL 2007

Обучение Java

Тема **Layout Managers**

Часть **Виды менеджеров**

Автор **ASKIL (omendba@gmail.com)**

3.06.2007

Менеджер расположения LM устанавливает компоненты контейнера в определенном положении. Он помещает и устанавливает размер компонентов в пределах области показа контейнера согласно специфической схеме расположения. AWT и Swing поставляются с стандартными менеджерами расположения, которые можно использовать в типичных ситуациях.

Каждый контейнер имеет менеджера расположения по умолчанию; поэтому, когда Вы делаете новый контейнер, он идет с объектом LayoutManager соответствующего типа. Вы можете установить нового менеджера расположения в любое время методом setLayout(). Например, мы можем установить менеджера расположения так:

```
mycontainer.getContentPane().setLayout(new BorderLayout());
```

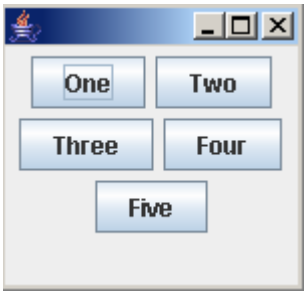
Вызов pack() устанавливает размер окна как можно меньше, с указанием компонентам их привилегированных размеров.

Каждый компонент определяет три важных параметра, используемые менеджером расположения в размещении: минимальный размер, максимальный размер, и привилегированный размер. Об этих размерах сообщают getMinimumSize() , getMaximumSize(), и getPreferredSize() методы.

FlowLayout

FlowLayout - простой менеджер расположения, который пробует устроить компоненты в их привилегированных размерах, слева направо и от вершины к основанию в контейнере.

```
setLayout(new FlowLayout());
add(new JButton("One"));
add(new JButton("Two"));
add(new JButton("Three"));
add(new JButton("Four"));
add(new JButton("Five"));
```

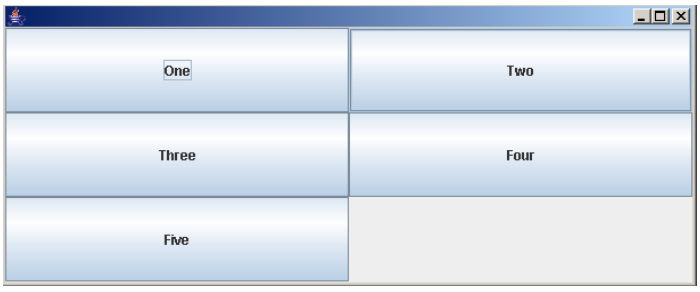


GridLayout

GridLayout устраивает компоненты в раздельные ряды и колонки. Компоненты произвольно изменяются, чтобы соответствовать сетке; их минимум и предпочтенные размеры игнорируются.

GridLayout принимает параметры – число рядов и колонок в своем конструкторе.

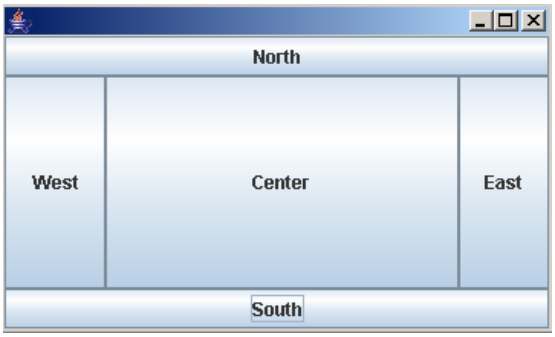
```
setLayout(new GridLayout(3, 2));
add(new JButton("One"));
add(new JButton("Two"));
add(new JButton("Three"));
add(new JButton("Four"));
add(new JButton("Five"));
```



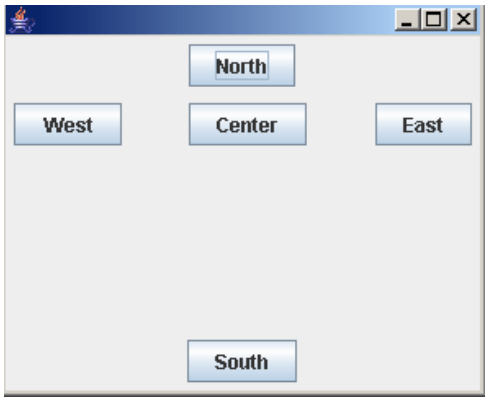
BorderLayout

BorderLayout более интересен. Он пробует устанавливать объекты в одном из пяти географических местоположений, представленных константами в классе BorderLayout: NORTH, SOUTH, EAST, WEST, и CENTER. BorderLayout - расположение по умолчанию для объектов JFrame и JWindow. BorderLayout может управлять самое большее пятью компонентами.

```
setLayout(new BorderLayout());
add(new JButton("North"), BorderLayout.NORTH);
add(new JButton("South"), BorderLayout.SOUTH);
add(new JButton("East"), BorderLayout.EAST);
add(new JButton("West"), BorderLayout.WEST);
add(new JButton("Center"), BorderLayout.CENTER);
```




```
setLayout(new BorderLayout());
JPanel p = new JPanel( );
p.add(new JButton("North"));
add(p, BorderLayout.NORTH);
p = new JPanel();
p.add(new JButton("South"));
add(p, BorderLayout.SOUTH);
p = new JPanel( );
p.add(new JButton("East"));
add(p, BorderLayout.EAST);
p = new JPanel( );
p.add(new JButton("West"));
add(p, BorderLayout.WEST);
p = new JPanel( );
p.add(new JButton("Center"));
add(p, BorderLayout.CENTER);
```



BoxLayout

Swing добавляет несколько новых менеджеров расположения общего назначения в javax.swing пакете; например – BoxLayout. Этот менеджер расположения полезен для того, чтобы создать панели инструментов или вертикальные кнопки на панели. Он располагает компоненты в единственном ряду или колонке.

Хотя Вы можете использовать BoxLayout непосредственно, Swing включает удобный контейнер по имени Box, который заботится обо всех деталях.

Вы можете создать горизонтальную или вертикальную коробку, используя статические методы Box.

```
Container horizontalBox = Box.createHorizontalBox();
Container verticalBox = Box.createVerticalBox();
```

Как только Box создан, Вы можете add() компоненты как обычно:

```
Container box = Box.createHorizontalBox();
box.add(new JButton("In the"));
```

```
Container box = Box.createHorizontalBox( );
box.add(Box.createHorizontalGlue( ));
box.add(new JButton("In the"));
box.add(Box.createHorizontalGlue( ));
box.add(new JButton("clearing"));
box.add(Box.createHorizontalStrut(10));
box.add(new JButton("stands"));
box.add(Box.createHorizontalStrut(10));
box.add(new JButton("a"));
box.add(Box.createHorizontalGlue( ));
box.add(new JButton("boxer"));
box.add(Box.createHorizontalGlue( ));
this.getContentPane( ).add(box, BorderLayout.PAGE_END);
this.pack();
```

CardLayout

CardLayout – специальный менеджер расположения для создания эффекта стека карт.

GridBagLayout

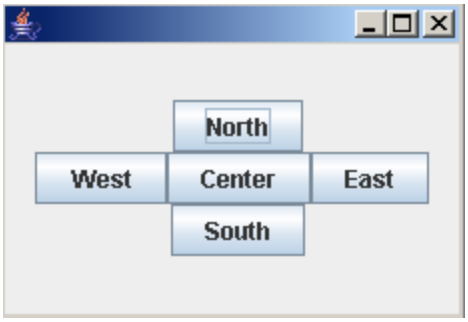
GridBagLayout – очень гибкий менеджер расположения, который позволяет Вам помещать компоненты относительно друг друга, используя ограничения. С GridBagLayout, Вы можете создать почти любое расположение. Компоненты устроены в логических координатах на абстрактной сетке. Ряды и колонки сетки имеют различные размеры, основанные на размерах и ограничениях компонентов, которые они содержат.

Ряд или колонка в GridBagLayout могут расширяться, чтобы приспособить измерения и ограничения наибольшего компонента. Индивидуальные компоненты могут охватить больше чем один ряд или колонку. Компоненты, которые не такого размера, как ячейка сетки, могут быть помещены в одну сторону в пределах ячейки.

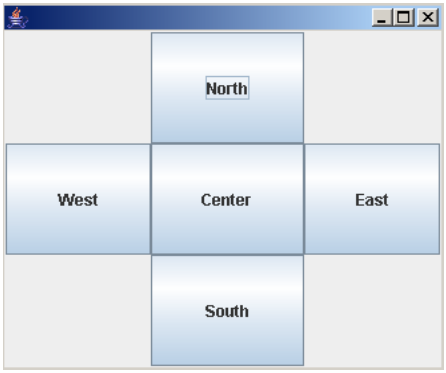
GridBagLayout намного легче использовать в графическом режиме GUI. Короче говоря, GridBagLayout сложен и имеет некоторые причуды.

```
GridBagConstraints constraints = new GridBagConstraints();
void addGB(Component component, int x, int y)
{
      constraints.gridx = x;
      constraints.gridy = y;
      add(component, constraints);
}
```

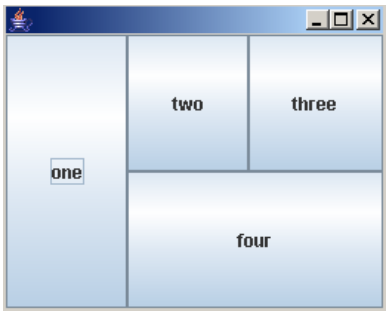
```
setLayout(new GridBagLayout());
int x, y;
addGB(new JButton("North"),  x = 1, y = 0);
addGB(new JButton("West"),   x = 0, y = 1);
addGB(new JButton("Center"), x = 1, y = 1);
addGB(new JButton("East"),   x = 2, y = 1);
addGB(new JButton("South"),  x = 1, y = 2);
```




```
setLayout(new GridBagLayout( ));
constraints.weightx = 1.0;
constraints.weighty = 1.0;
constraints.fill = GridBagConstraints.BOTH;
int x, y;
addGB(new JButton("North"),  x = 1, y = 0);
addGB(new JButton("West"),   x = 0, y = 1);
addGB(new JButton("Center"), x = 1, y = 1);
addGB(new JButton("East"),   x = 2, y = 1);
addGB(new JButton("South"),  x = 1, y = 2);
```




```
setLayout(new GridBagLayout( ));
constraints.weightx = 1.0;
constraints.weighty = 1.0;
constraints.fill = GridBagConstraints.BOTH;
int x, y;
constraints.gridheight = 2;
addGB(new JButton("one"),   x = 0, y = 0);
constraints.gridheight = 1;
addGB(new JButton("two"),   x = 1, y = 0);
addGB(new JButton("three"), x = 2, y = 0);
constraints.gridwidth = 2;
addGB(new JButton("four"),  x = 1, y = 1);
constraints.gridwidth = 1;
```




```
setLayout(new GridBagLayout( ));
constraints.fill = GridBagConstraints.BOTH;
constraints.weighty = 1.0;
int x, y;
constraints.weightx = 0.1;
addGB(new JButton("one"), x = 0, y = 0);
constraints.weightx = 0.5;
addGB(new JButton("two"), ++x, y);
constraints.weightx = 1.0;
addGB(new JButton("three"), ++x, y);
```

Absolute Positioning

Возможно установить менеджера расположения в null: никакой контроль расположения. Можно сделать, чтобы поместить объект в некоторых абсолютных координатах. Это – обычно не правильный подход. Компоненты могут иметь различные минимальные размеры на различных платформах, и ваш интерфейс не будет портативен.