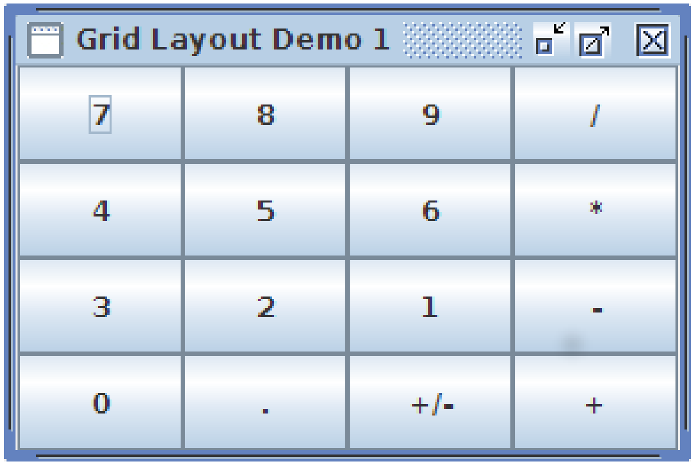
8. GUI Programming :: Java Layout Managers :: *GridLayout* Demo 1

The final Java layout manager we will look at is *java.awt.GridLayout* which arranges the components in a rectangular grid of rows and columns. For example, consider this calculator GUI using a *Grid Layout* with four rows and four columns:



We start by creating a *JPanel* that uses *GridLayout*:

```
JPanel panel = new JPanel();
panel.setLayout(new GridLayout(4, 4));
```

The first argument to the constructor is the number of rows and the second is the number of columns.

8. GUI Programming :: Java Layout Managers :: *GridLayout* Demo 1 (continued)

To add the *JButtons* we call *addButton*() sixteen times:

```
addButton(panel, "7");
addButton(panel, "8");
addButton(panel, "9");
addButton(panel, "/");
addButton(panel, "4");
addButton(panel, "5");
addButton(panel, "6");
addButton(panel, "*");
addButton(panel, "3");
addButton(panel, "2");
addButton(panel, "1");
addButton(panel, "-");
addButton(panel, "0");
addButton(panel, ".");
addButton(panel, "+/-");
addButton(panel, "+");
```

```
private void addButton(JPanel pPanel, String pText) {
  JButton button = new JButton(pText);
  pPanel.add(button);
}
```

Note that components are added in **row major** order.

8. GUI Programming :: Java Layout Managers :: *GridLayout* Demo 1 (continued)

```
// GridLayoutDemo.java
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
public class GridLayoutDemo {
  public static void main(String[] args) { new GridLayoutDemo().run(); }
  public void run() {
    JFrame.setDefaultLookAndFeelDecorated(true);
    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(4, 4));
    addButton(panel, "7");
    addButton(panel, "8");
    addButton(panel, "9");
    addButton(panel, "/");
    addButton(panel, "4");
    addButton(panel, "5");
    addButton(panel, "6");
    addButton(panel, "*");
    addButton(panel, "3");
    addButton(panel, "2");
    addButton(panel, "1");
    addButton(panel, "-");
    addButton(panel, "0");
```

8. GUI Programming :: Java Layout Managers :: *GridLayout* Demo 1 (continued)

```
        addButton(panel, ".");
        addButton(panel, "+/-");
        addButton(panel, "+");
        JFrame frame = new JFrame("Grid Layout Demo");
        frame.setSize(400, 300);
        frame.add(panel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
    private void addButton(JPanel pPanel, String pText) {
        JButton button = new JButton(pText);
        pPanel.add(button);
    }
}
```