

プログラミング基礎

GUI (1)

GUI (Graphical User Interface)

- ウィンドウ、アイコン、メニューなどで表示
- ポインティングデバイスで操作
 - マウス、タッチパネルなど
- 対比
 - 情報・機能を文字で表示し、コマンド入力で操作
 - CUI: Character User Interface
 - CLI: Command-Line Interface

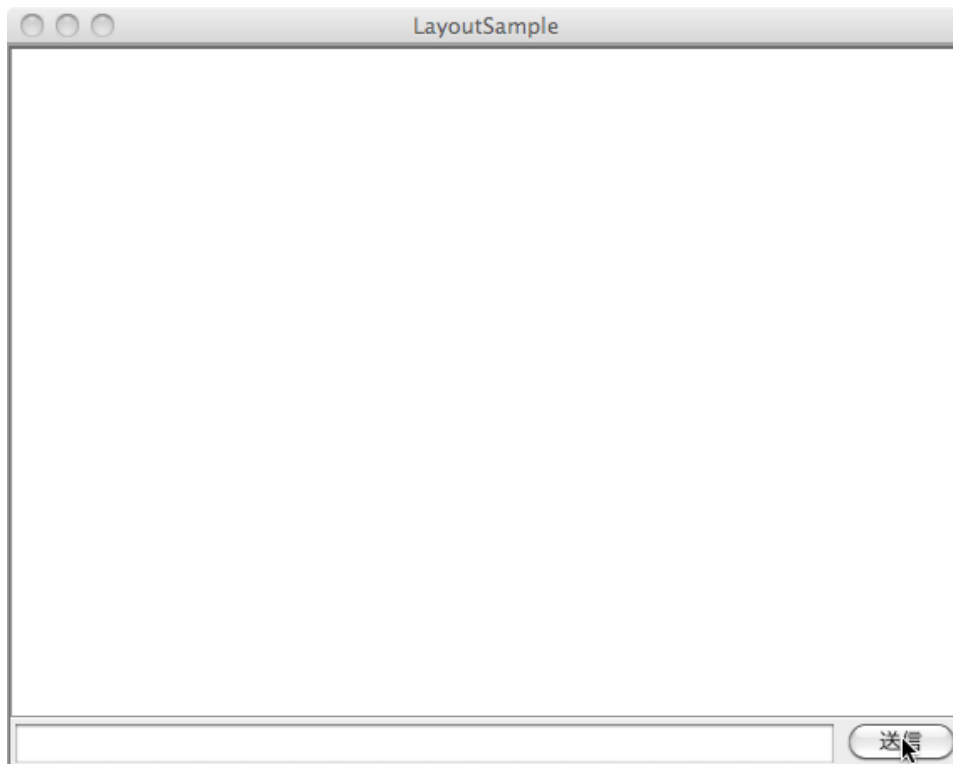
JavaでGUIを作るためのAPI

- Swing (javax.swing パッケージ)
- 他にも AWT, SWT などの API もある

ウィンドウと部品の配置

部品配置の階層構造

- HTML と似ている



[イメージ]

```
<window>
```

```
    <textarea>
```

```
    <textfield>
```

```
    <button>送信</button>
```

```
</window>
```

ウィンドウを表示する

- 以下のプログラムを書いて実行してみよう

```
public class LayoutSample {  
    public static void main(String[] args) {  
        // window を作成する  
        JFrame frame = new JFrame("LayoutSample");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true); // window を表示する  
    }  
}
```

ウィンドウに部品を追加

- ボタン(JButton)を追加
 - 青い部分が増えた部分

```
JFrame frame = new JFrame("LayoutSample");  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
JButton button = new JButton("送信");  
frame.add(button);  
frame.pack(); // window の大きさを中身に合わせる  
frame.setVisible(true); // window を表示する
```

部品を複数追加して、レイアウトする

- JTextArea
(複数行テキスト)
- JTextField
(1行テキスト)
- Box
(配置用の箱)
 - 縦方向配置
 - 横方向配置

```
// 中身の部品を作成
JTextArea textArea = new JTextArea(40, 40);
JTextField textField = new JTextField(40);
JButton button = new JButton("送信");

// 中身の部品を配置
Box box1 = Box.createVerticalBox();
Box box2 = Box.createHorizontalBox();
box1.add(textArea);
box1.add(box2);
box2.add(textField);
box2.add(button);
frame.add(box1);

frame.pack(); // window の大きさを中身に合わせる
```

次回予告：部品同士の連携

- 例
 - button をクリックすると、textField の内容が textArea に追加される
 - button をクリックすると、（中略）送信されて、相手の textArea に追加される
- 今日は新しい課題なし
遅れている分を取り戻しましょう！