

プログラミングII

<http://bit.ly/Prog3i>

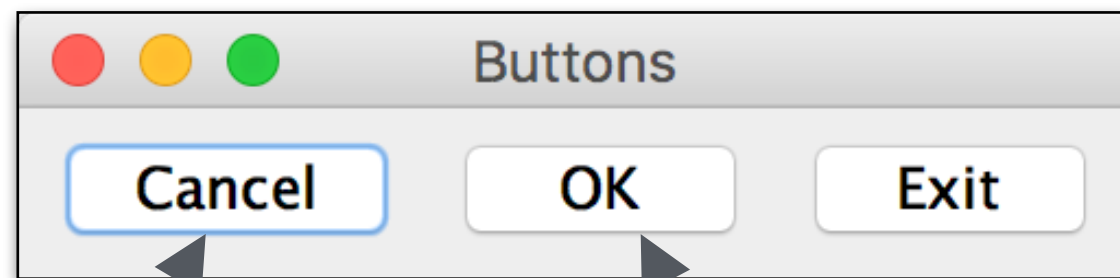
GUI (4)

後期 第13週

2019/12/25

今回は・・・

複数のイベントソース（ボタンなど）で、
異なる動作をするアプリケーションを作ります。



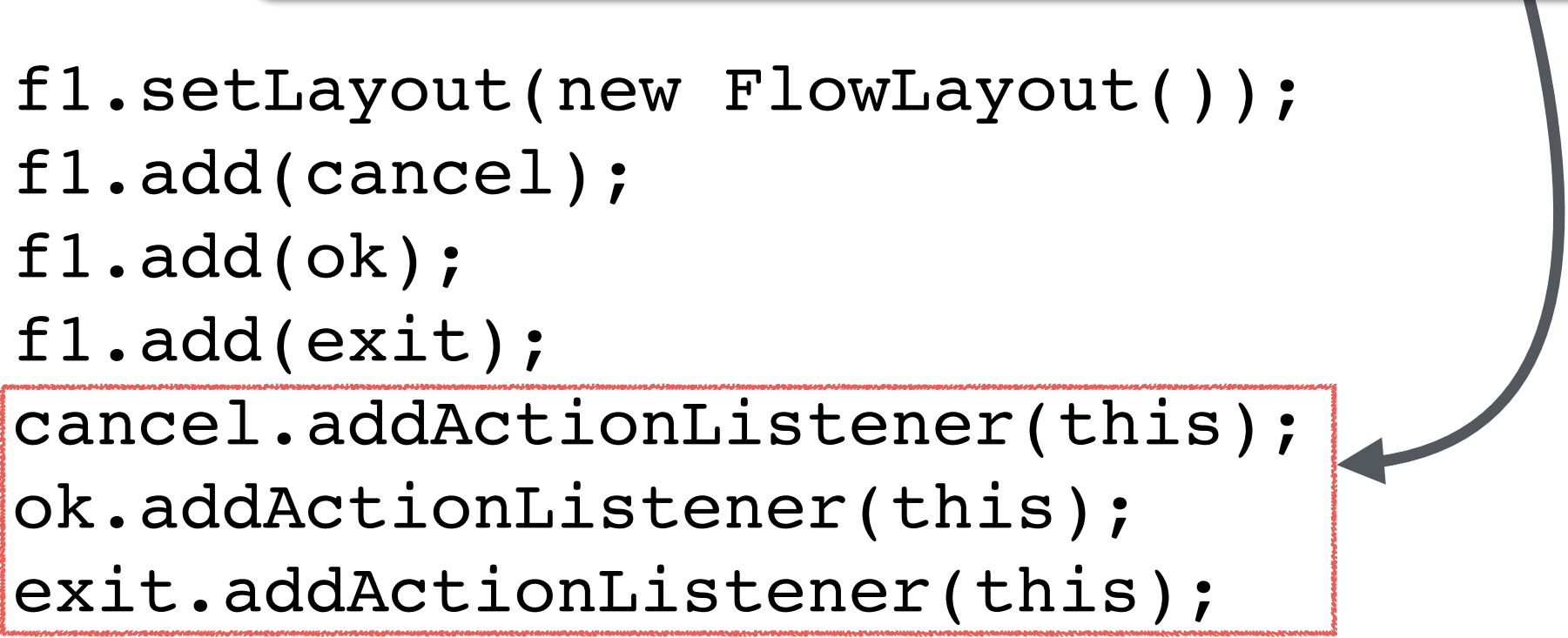
これを押すとキャンセルの処理

これを押すとOKの処理

```
1: import java.awt.*;
2: import java.awt.event.*;
3:
4: class MyFrame implements ActionListener {
5:     private Frame f1;
6:     private Button ok, cancel, exit;
7:     MyFrame() {
8:         f1 = new Frame("Buttons");
9:         cancel = new Button("Cancel");
10:        ok = new Button("OK");
11:        exit = new Button("Exit");
```

3つのボタンに自身をリスナとして登録する

```
12:      f1.setLayout(new FlowLayout());
13:      f1.add(cancel);
14:      f1.add(ok);
15:      f1.add(exit);
16:      cancel.addActionListener(this);
17:      ok.addActionListener(this);
18:      exit.addActionListener(this);
19:      f1.pack();
20:      f1.setVisible(true);
21:  }
```



イベントに関する様々な情報が渡される

「e.getSource()」でイベントソースのインスタンスを得られる
「イベントソースのインスタンスがcancelならば」つまり
「cancelボタンが押されたならば」という意味になる

```
22:      public void actionPerformed(ActionEvent e) {  
23:          if(e.getSource()==cancel) {  
24:              System.out.println("キャンセル");  
25:          }  
26:          if(e.getSource()==ok) {  
27:              System.out.println("OK");  
28:          }  
29:      }  
30:      public static void main(String[] args) {  
31:          MyFrame obj = new MyFrame();  
32:      }  
33:  }
```

「イベントソースのインスタンスがcancelならば」
つまり「okボタンが押されたならば」

【補足】 文字列と整数の相互変換

**String型の文字列からint型の整数へ変換する場合
(テキストフィールドから得た文字列を整数に変換するなど)**

```
String s1;
```

```
int n1;
```

```
//テキストフィールドtfの文字列を取得
```

```
s1 = tf.getText();
```

```
//parseIntメソッドで整数に変換された値がn1に代入される
```

```
n1 = Integer.parseInt(s1);
```

【補足】 文字列と整数の相互変換

**int型の整数からString型の文字列へ変換する場合
(計算した整数を文字列に変換してラベルにセットするなど)**

```
String s2;  
int n2;  
  
n2 = 100;    //n2に100という整数が入っているとする  
//valueOfメソッドで変換された文字列がs2に代入される  
s2 = String.valueOf(n2);  
  
//整数から変換した文字列s2をラベルlbにセットする  
lb.setText(s2);
```

【課題の準備】

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone 「自分のGitHubユーザ名」  
$ cd prog3i-ユーザ名  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

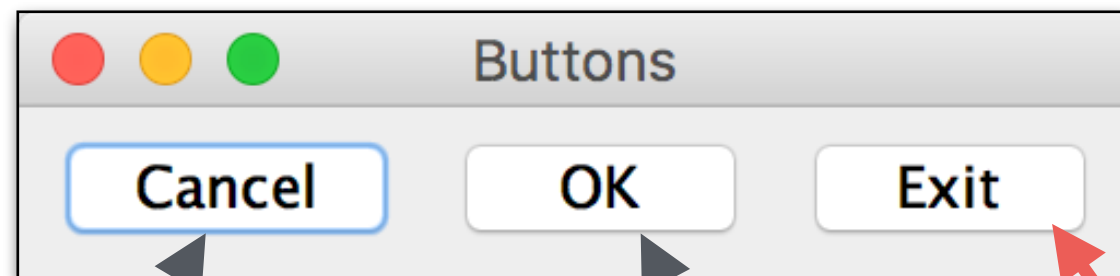
1. 端末を起動して、以下のコマンドを実行して後期第13週のフォルダを作る
\$ cd prog3i-ユーザ名 (←既に移動しているなら不要)
\$ mkdir week213
\$ cd week213

【練習13-1】

サンプルプログラム「2_13_MyFrame.java」を
コンパイルして、実行結果を確認しましょう。

【課題13-1】

サンプルプログラム「2_13_MyFrame.java」を、
「Exitボタンを押したらアプリケーションが終了する」機能を追加してください。



これを押すとキャンセルの処理

これを押すとアプリ終了

これを押すとOKの処理

【課題の提出】

以下の流れで、作ったプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題13-1提出”
```

```
$ git push origin master
```

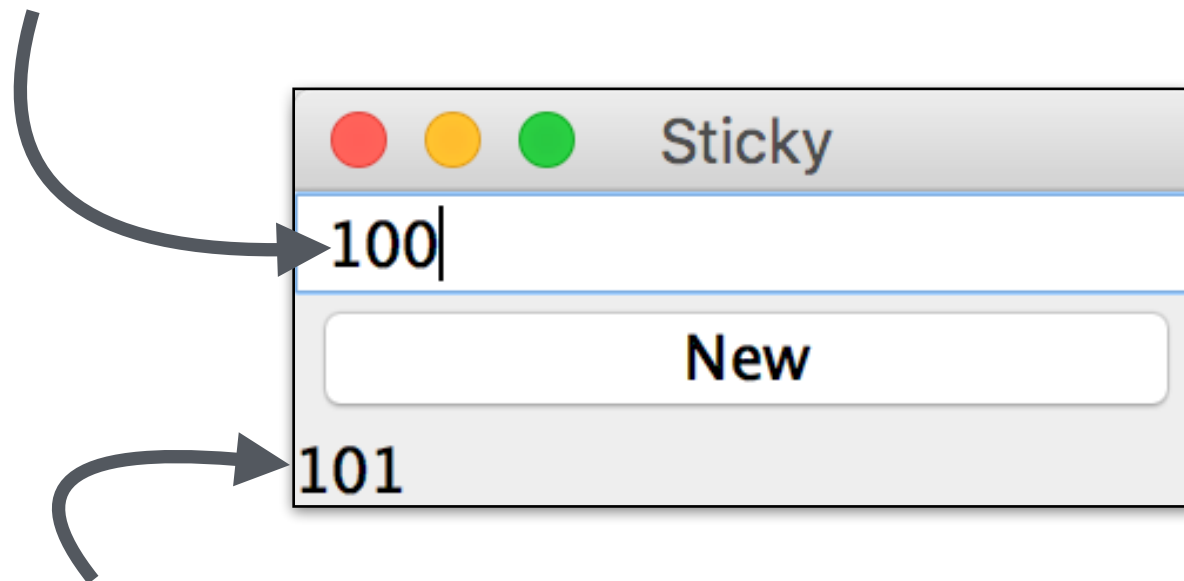
2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題13-2】

「2_13_Sticky.java」に「テキストフィールドに整数を入力してEnterキーを押したら、ラベルに**1加算した結果**が表示される」動作を追加してください。

(1) ここに整数を入れて、Enterキーを押すと・・・



※必ず整数が入力されるとする
(整数以外を入れると
Exceptionが発生する)

(2) ラベルに1加算された結果が表示される

【課題の提出】

以下の流れで、作ったプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題13-2提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

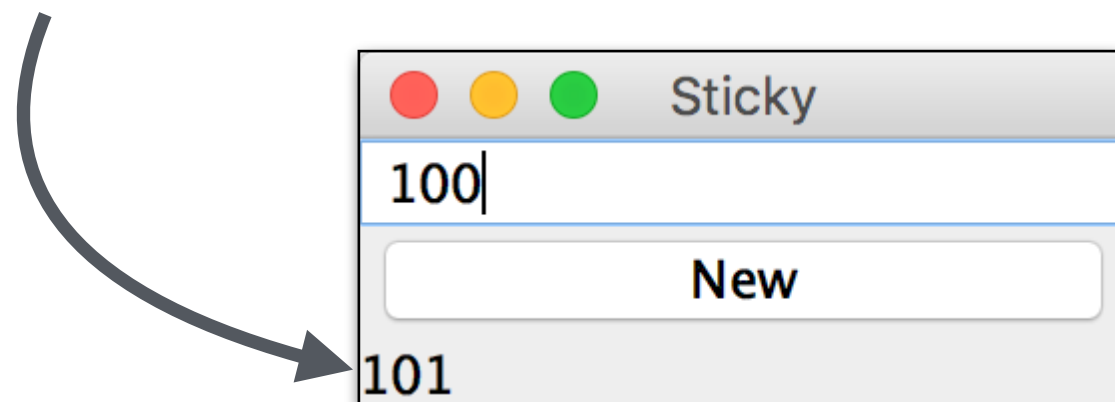
[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題13-3】

課題13-2のプログラムを次のような動作となるように改良してください。

- ▶ テキストフィールドでEnterを押すと課題13-2の動作
- ▶ Newボタンを押すと課題12-3の動作
(つまり、新規フレームに付箋を作る)

Enterキーを押すと (つまりイベントソースがテキストフィールドだと) こっちの動作



Newボタンを押すと (つまりイベントソースがNewボタンだと) こっちの動作



【課題の提出】

以下の流れで、作ったプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題13-3提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

小テストについて

小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）
- プログラムの提出はGitHubを使用する。

小テストについて

小テスト中に参照できるもの

- 教科書, 参考書, 配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- 小テストでは紙媒体のものは参照可能
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
Webブラウザ、ネットワークを介した情報の参照
自分のPCを使用する、など