

慶応義塾大学文学部
応用情報処理V 2017

ボタンに反応させる

Chapter 12
2017-12-22

新居雅行 Masayuki Nii Ph.D. in Engineering

 nii@msyk.net  [msyknii](https://www.facebook.com/msyknii)  [msyk_nii](https://twitter.com/msyk_nii)

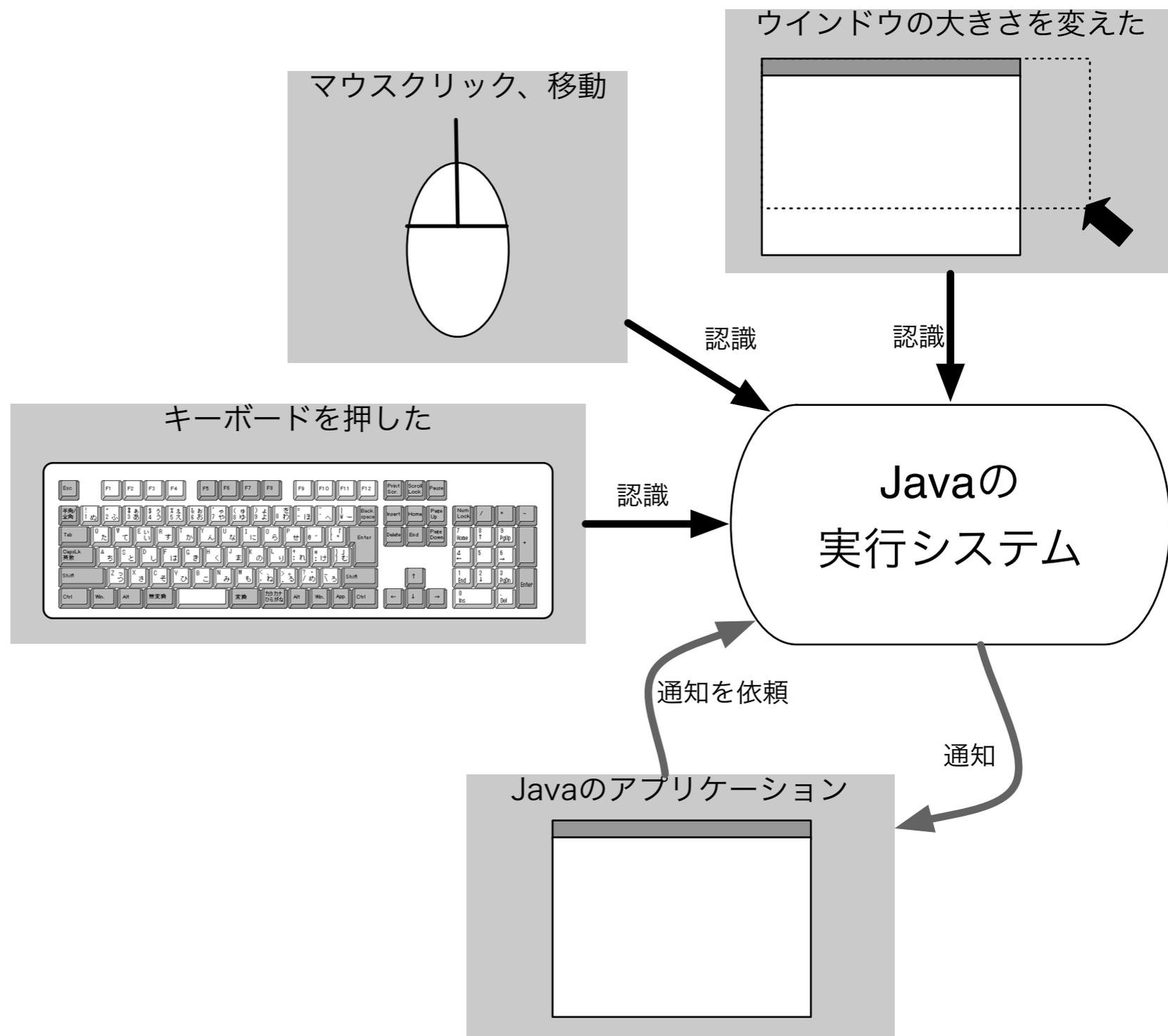
イベントの考え方

原則

- ・ 通知の依頼
- ・ 通知の受付

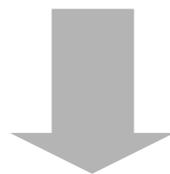
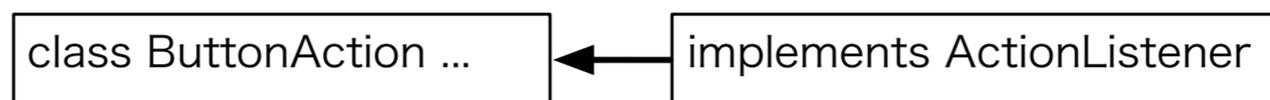
依頼や受付可能なイベント

- ・ システム側で定義されている



イベントを受け付ける

- ・ イベントの受付 = 決められたメソッドの定義
 - ・ 考え方としては「イベントが発生すると、決められたメソッドが呼び出される」でOK
 - ・ 例：actionPerformed(ActionEvent)メソッド
- ・ 呼び出されるメソッドはインタフェースとして定義されているので、クラスでそれを利用することを宣言すれば、必要なメソッドは実装が必須となる

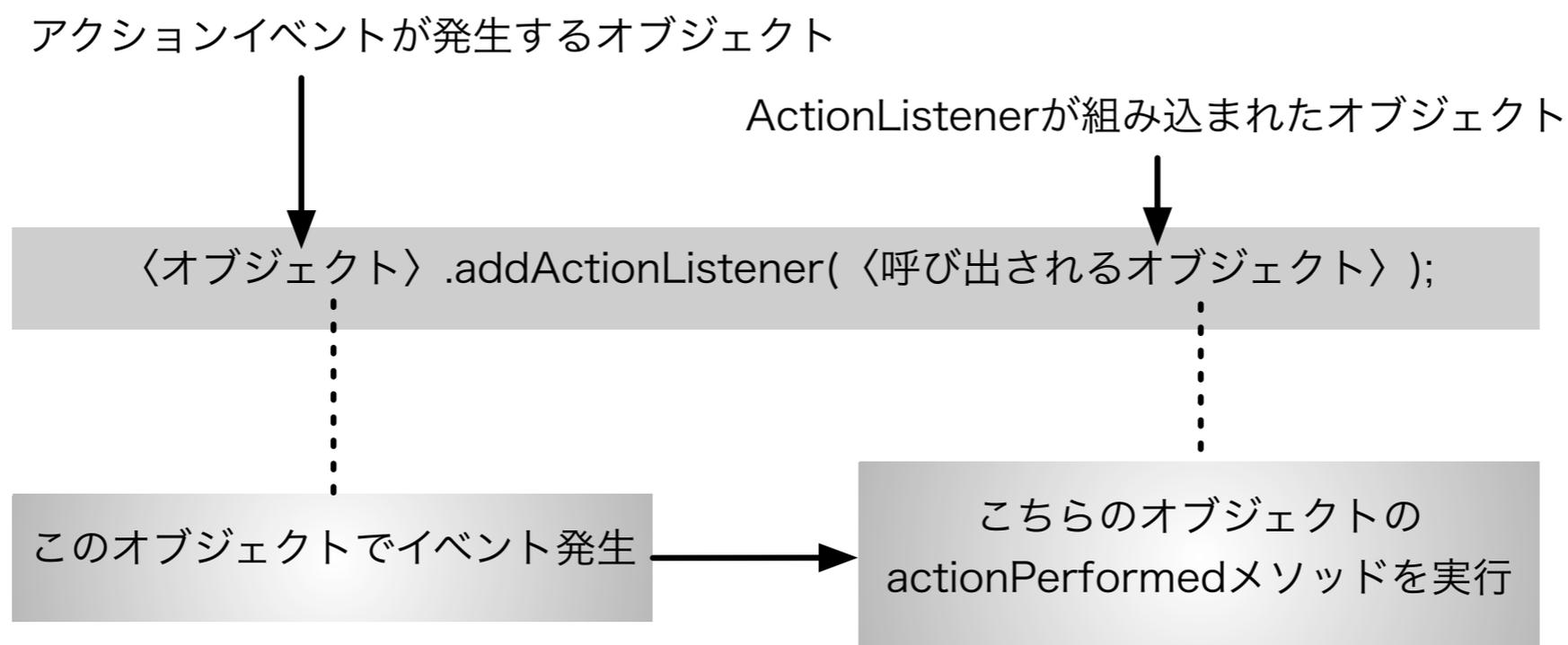


```
class ButtonAction ... implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        ....  
    }  
}
```

通知依頼

- ・ **あらかじめ依頼する**

- ・ イベントが発生するオブジェクトに対して「イベントリスナー」を追加する
- ・ イベントが発生したときに呼び出されるオブジェクトを引数で指定する
- ・ オブジェクトのクラスは特定のインターフェースが実装されている必要がある



イベント実装のパターン

- **ウィンドウが通知を受ける**
 - (テキスト:ButtonAction1)
 - 通知を受けるオブジェクトが増えたときに処理が複雑化
- **処理するプログラムを登録時に記述する**
 - (テキスト:ButtonAction2)
 - 直接記述できる感じは分かりやすい
 - 処理が長くなるとき、いろいろ不便
 - Java8以降ではラムダ式を使えば次のように記述できる
 - ```
this.activeButton.addActionListener(e -> {
 ButtonAction.this.clickButton(e);
});
```
- **通知を受けるボタンを定義する**
  - (テキスト:ButtonAction3)
  - オブジェクト指向的な方法

# 講義のまとめ

---

- ・ 状況に応じた処理は「イベント処理」と呼ばれる
  - ・ Javaのシステムに多数のイベントが定義されている
- ・ イベント処理を行うには、あるオブジェクトでイベントが発生した場合に、どのオブジェクトに通知するかを登録しておく必要がある
- ・ 登録しておけば、通知=メソッド呼び出しとなる
  - ・ イベントの種類ごとに決められたメソッドがある
  - ・ そのメソッドを実装しておくことで、イベント対応処理が記述できる