



# Eventとその処理

オブジェクト指向プログラミング特論

2018年度

只木進一：工学系研究科

# Event駆動

## ■ イベント

■ マウス操作、キーボード操作、プログラム構成要素の変化などを検知

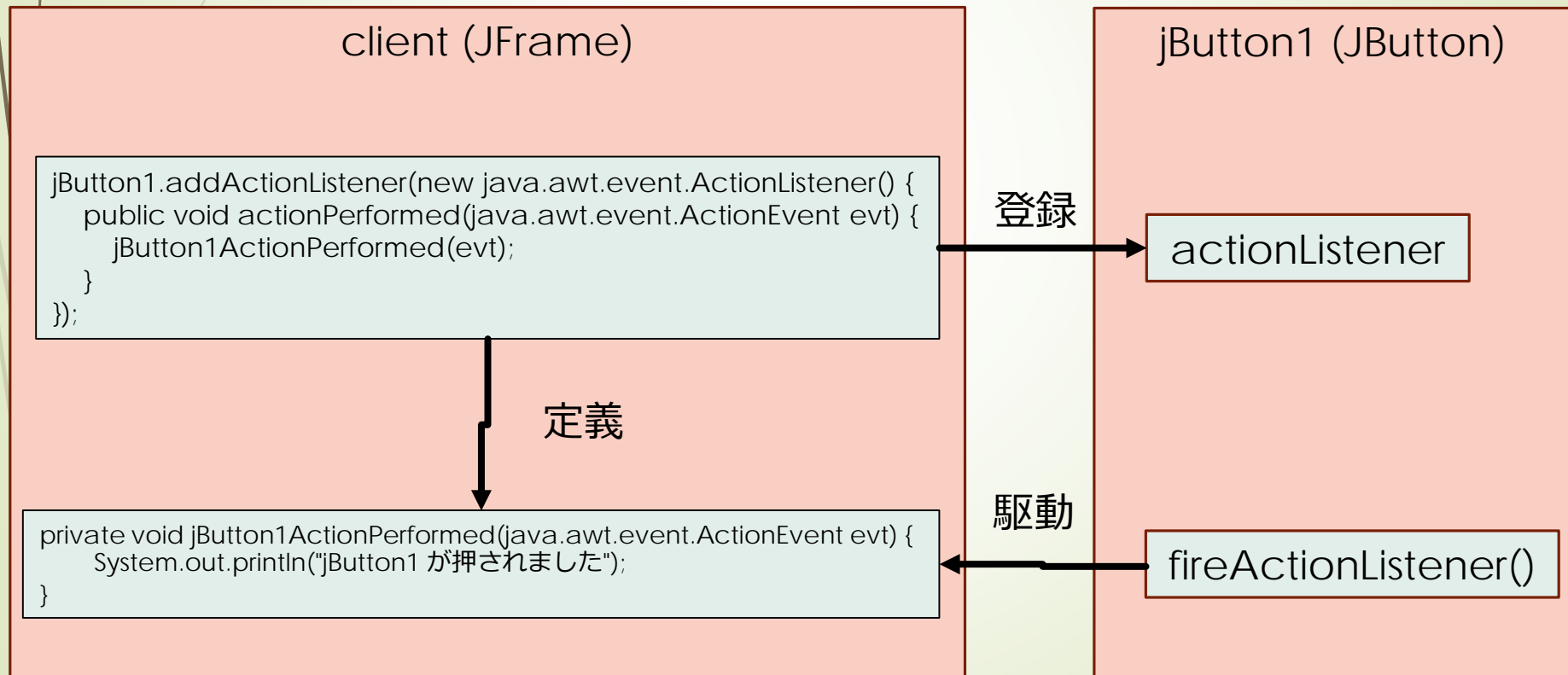
■ イベントを契機に動作することをイベント駆動と呼ぶ

■ GUIなどでは標準的な仕組み

# JavaにおけるEvent処理

- Eventの定義
  - java.util.EventObjectの派生クラス
- Eventを受け取るEventListener
  - java.util.EventListenerの派生クラス
- Event発生時の処理の記述
  - listenerインスタンスで記述

# guiWithActionでの動作：復習



# イベントを発生させる側の動作

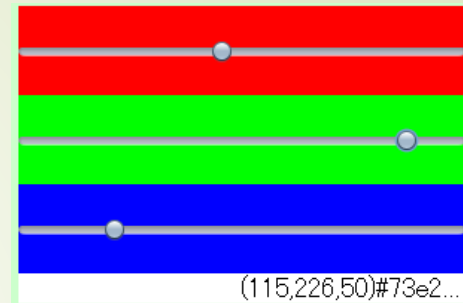
- listener、あるいはそのリストを保持
  - listenerを登録するメソッド
    - addActionListener()
- イベント発生時
  - listenerにイベントを通知
    - fireActionListener()

# イベント活用の例

## RGBをスライダで指定

- 色をRGBで指定する
- JSliderクラス
  - スライダーで数値を指定できる
  - ChangeEventを発生させる
- RGBの三色のスライダーを使いたい
  - JPanelにならべたい
  - JPanelはChangeEventを発生しない

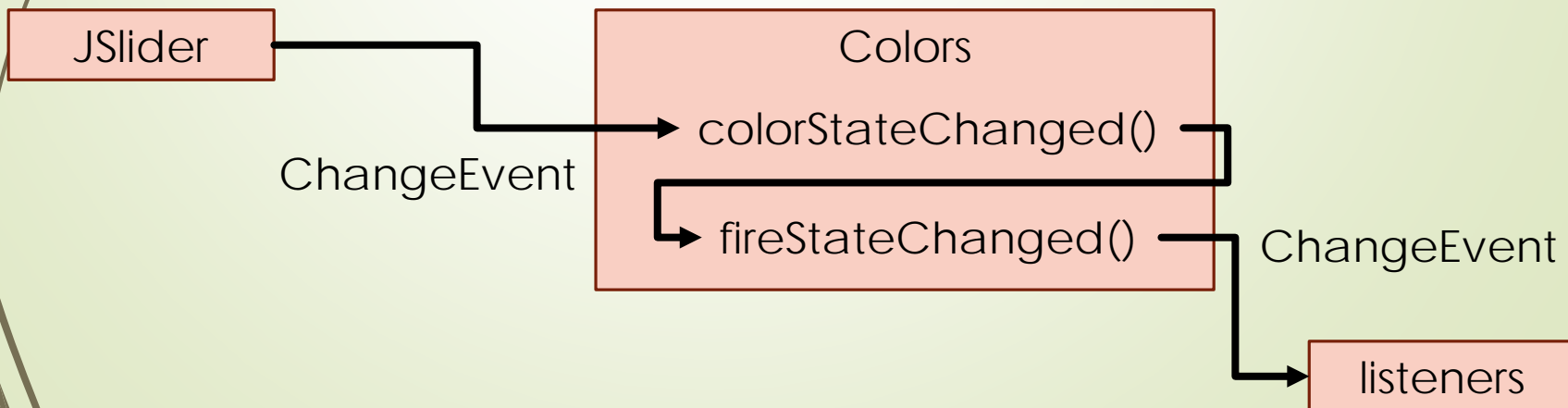
# ColorPanelクラスを設計



- `ChangeListener`のリスト
- `listener`へ変更を通知するメソッド
  - `fireStateChanged()`
- 三つの `JSlider` の `stateChange()` は色変更のメソッド `colorStateChange()` を呼びぶ
  - `fireStateChanged()` で `listener` へ通知

# Colors クラス

- JPanel クラスを拡張
- 三つのスライダーで **RGB** の要素を設定
- **RGB** の変化があったら、**Change** イベントを発生





```
public class Colors extends javax.swing.JPanel {  
  
    private Color rgb;  
    private final List<ChangeListener> listeners;  
    ...  
    public void addChangeListener(ChangeListener listener) {  
        listeners.add(listener);  
    }  
  
    protected void fireStateChanged(ChangeEvent e) {  
        setRgb();  
        //listenerに知らせる  
        listeners.forEach(li -> li.stateChanged(e));  
    }  
    ...  
}
```

# ColorFrame クラス

- JFrameの拡張
  - QUITボタン
  - Colorクラス
  - 色を表示するJPanel
- ColorクラスからのChangeEvent
  - Color.getRGB()
  - JPanel.setBackground()