

デザイナーを使った実装2(パスワード表示とスクロールバー)

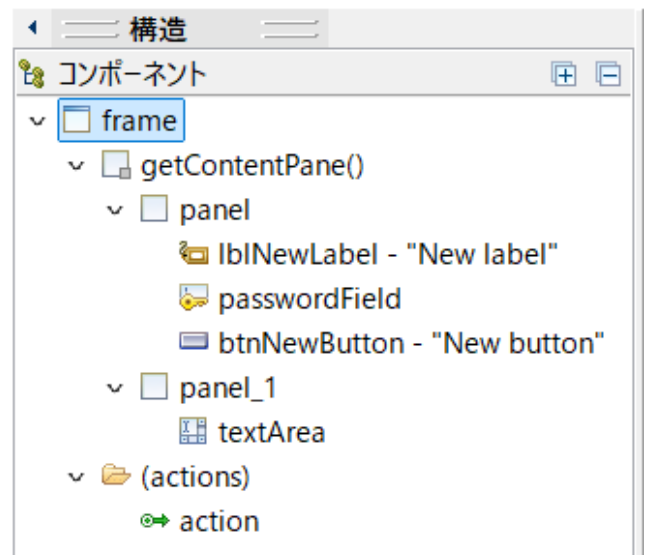
ボタンをクリックするとパスワードフィールドに入力したパスワードがテキストエリアに出力される GUI アプリケーションを作成します。(プロジェクト名:Swing)

1. クラスファイル作成

メニューバー → ファイル → 新規 → その他 → WindowBuilder
→ Swing デザイナー → アプリケーション・ウィンドウ
→ パッケージ名 :jp.ict.aso.swing クラス名 :Exec310

2. GUI デザイン作成(デザインタブ)

【パレット】		【配置場所】
① BorderLayout	→	FRAME 全体
② JPanel	→	NORTH 位置
③ JLabel	→	NORTH 位置
④ JPasswordField	→	NORTH 位置
プロパティ	→	columns 10
⑤ JButton	→	NORTH 位置
⑥ JPanel	→	CENTER 位置
⑦ FlowLayout	→	CENTER 位置
⑧ JTextArea	→	CENTER 位置
プロパティ	→	columns 30 rows 10
⑨ SwingActions 新規	→	JButton 位置



BorderLayout は、画面を上下左右中の 5つのブロックに分け、それぞれに部品を配置します。5つのブロックはそれぞれ、東西南北中央(East, West, South, North, Center)で示されます。

FlowLayout クラスはコンポーネントを左から右へ順に配置するレイアウト・マネージャーです。1つの行に表示できなくなると、自動的に次の行にコンポーネントを配置します。

3. ソース変更(ソースタブ)

① //koko の範囲を追加(イベント処理の記述)

```
private class SwingAction extends AbstractAction {  
    public SwingAction() {  
        putValue(NAME, "SwingAction");  
        putValue(SHORT_DESCRIPTION, "Some short description");  
    }  
    public void actionPerformed(ActionEvent e) {  
        //koko/////////////////////////////////////  
        char[] password = passwordField.getPassword();  
        String passwordstr = new String(password);  
        textArea.append(passwordstr+"\n");  
        frame.setVisible(true);  
        System.out.println(passwordstr);  
        //koko/////////////////////////////////////  
    }  
}
```

② //koko の部分を追加(インスタンス変数の宣言追加)

```
public class Exec310 {  
    private JFrame frame;  
    private JPasswordField passwordField;  
    private final Action action = new SwingAction();  
    private JTextArea textArea; //koko  
  
    /**  
     * Launch the application.  
     */  
}
```

③ //koko の部分を変更(ローカル変数の宣言削除)

```
JPanel panel_1 = new JPanel();  
frame.getContentPane().add(panel_1, BorderLayout.CENTER);  
panel_1.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));  
  
textArea = new JTextArea(); //koko  
textArea.setRows(10);  
textArea.setColumns(30);  
panel_1.add(textArea);
```

4. 実行

ソースコード内を右クリック → 実行 → Java アプリケーション

実行結果の例

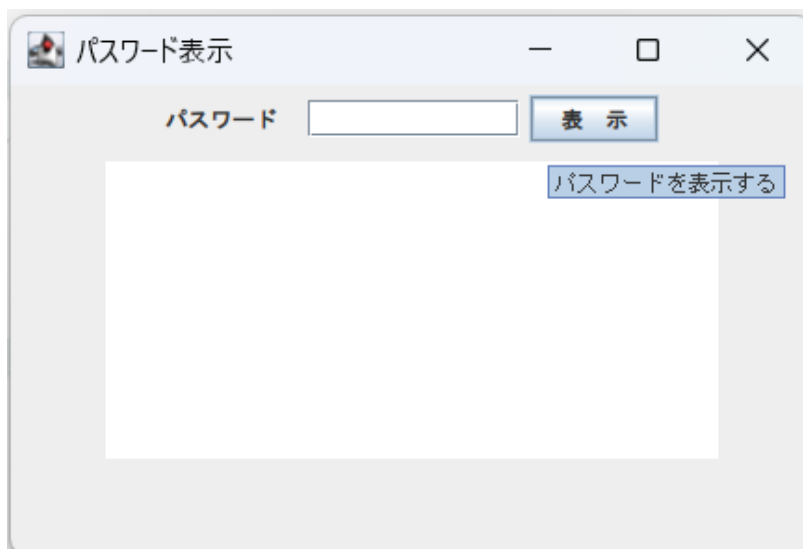


5. 表示名の変更

以下のように各オブジェクトの内容を変更してください

- ・タイトルを付けます
- ・ラベルの表示を変更します
- ・ボタンの表示を変更します あわせてツールチップ(注釈補足情報)も設定します

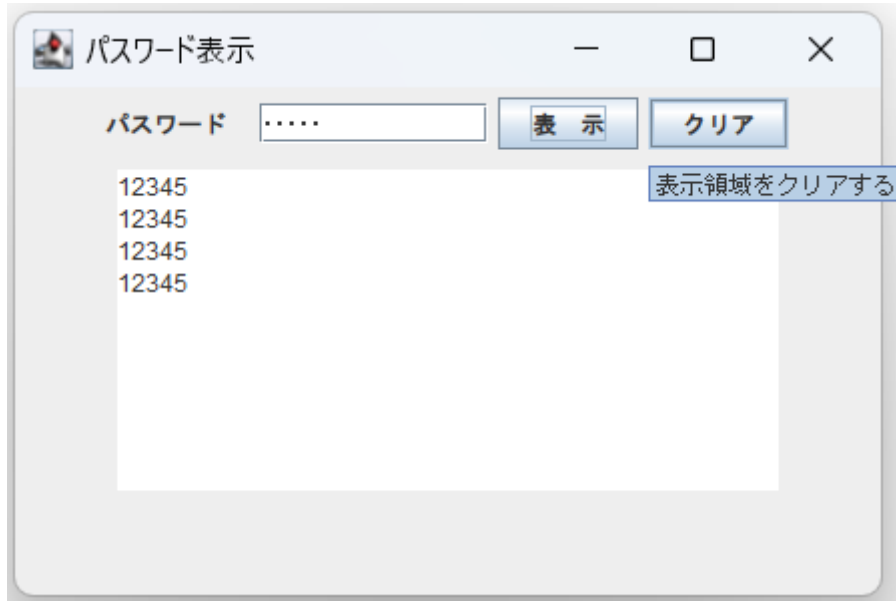
ソースタブでもデザインタブでも変更可能ですが、**デザインタブのプロパティだけでは設定できない項目がありますのでその時はソースを直接変更します。**



【演習1】

上記プログラムにテキスト領域のクリアボタンを付けてください
追加位置は表示ボタンの横です

テキスト領域のクリア処理は次のようにして実装できます
フィールド変数名.setText(null);

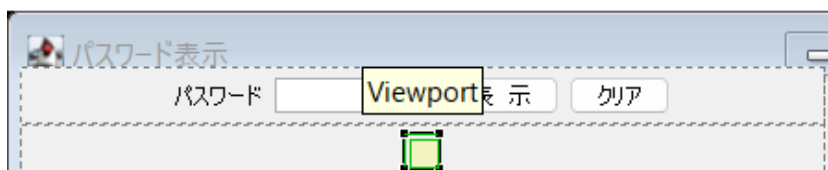


【演習2】

上記プログラムを
テキスト領域が縦横のスクロール可能なように常にスクロールバーが出るよう変更し
てください

(手順1 デザインタブより)

構造のコンポーネント → panel_1 内の textArea を右クリック → 削除
→ パレットから JScrollPane をクリック→構造のコンポーネントから panel_1 をクリック
構造のコンポーネント → panel_1 内の scrollPane をクリック
→ プロパティで HORIZONTAL_SCROLLBAR_ALWAYS を選択
→ プロパティで VERTICAL_SCROLLBAR_ALWAYS を選択
パレットから JTextArea をクリック → デザインイメージの scrollPane 領域の
ViewPoint が表示される位置でをクリック



→ textArea のプロパティで columns を 30、row を 10 に設定



(手順2 ソースタブより)

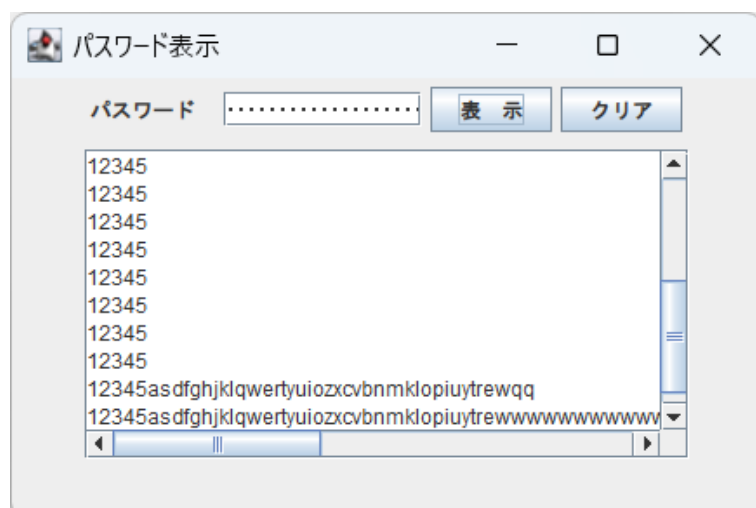
①フィールドに JTextArea textArea の宣言を追加

```
public class Exec310 {  
    private JFrame frame;  
    private JPasswordField passwordField;  
    private final Action action = new SwingAction();  
    private final Action action_1 = new SwingAction_1();  
    private JTextArea textArea; //koko
```

②メソッド内の JTextArea のローカル変数の宣言を変更

```
textArea = new JTextArea(); //koko  
textArea.setRows(10);  
textArea.setColumns(30);  
scrollPane.setViewportView(textArea);
```

(手順3 実行して確認)



(最終的な構造)

