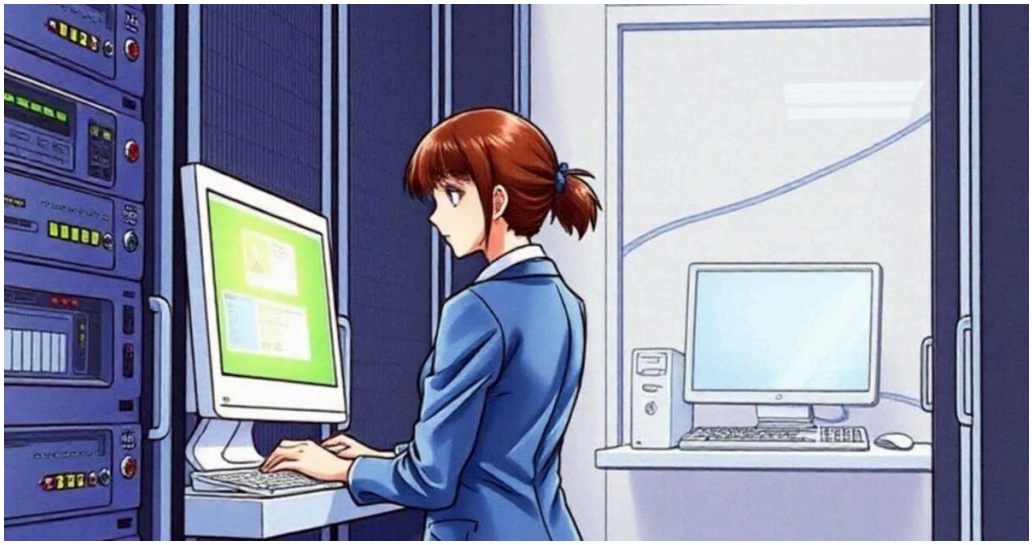

Swingによるデスクトップアプリケーション開発（図書貸出検索）-JavaSE1.8

office・M
2024年11月4日 13:54

Java8のSwing環境でデスクトップアプリケーションの開発方法を学ぶ講座をシリーズで提供しています。今回はWindowBuilder（Swingデザイナー）を使ってPostgreSQLデータベースに接続して図書の貸出履歴を検索するシステムを作成します。

2024年9月よりECLIPSEのバージョンを最新版（Version: 2024-06 (4.32.0)）に変更しました。

目次

外部設計

WindowBuilderのSwingデザイナーで図１・図２のようなGUIを作成します。
・サーバのIPアドレス、データベース名はデフォルトでの設定値を表示します。
・認証用に登録済みの電話番号とパスワードを入力し検索実行ボタンのクリックで認証が成功すると、電話番号をもとにデータベースから図書の貸出履歴を検索し表示します。
・認証できないときは「貸出し履歴を取得することができません」のメッセージを表示します。
・クリアボタンをクリックすることで「認証電話番号」「パスワード」「検索結果」の各領域をクリアします。

図1．図書貸出履歴検索システム起動画面

図2．図書貸出履歴検索システム検索結果表示画面

内部設計

DB設計

「ハッシュログイン機能付き図書の貸出履歴検索システム-JavaEE8,
PostgreSQL」で定義されたDB設計を用います。当該Webページの認証方式、論理設計、物理設計、を参照して実装してください。
また、認証用のT_USERテーブルも同様に当該Webページを参照して実装してください。

処理ロジック

DBに接続して図書検索を行う業務ロジック（**DataAccessObject**）を実装したBeanは「ハッシュログイン機能付き図書の貸出履歴検索システム-JavaEE8,
PostgreSQL」で作成したクラスファイルを流用します。当該Webページを参照して仕様を確認してください。流用するクラスファイルは、以下の３点となります。

- **PostgresConnectionBean.class**
- **BookLibraryUserCheckBean.class**
- **BookLibraryArrayListBean.class**

実装準備

プロジェクトの作成

Eclipseのメニューバーより
ファイル→新規→Javaプロジェクト→「SwingLibrary」プロジェクトを作成
→図３の内容で設定する

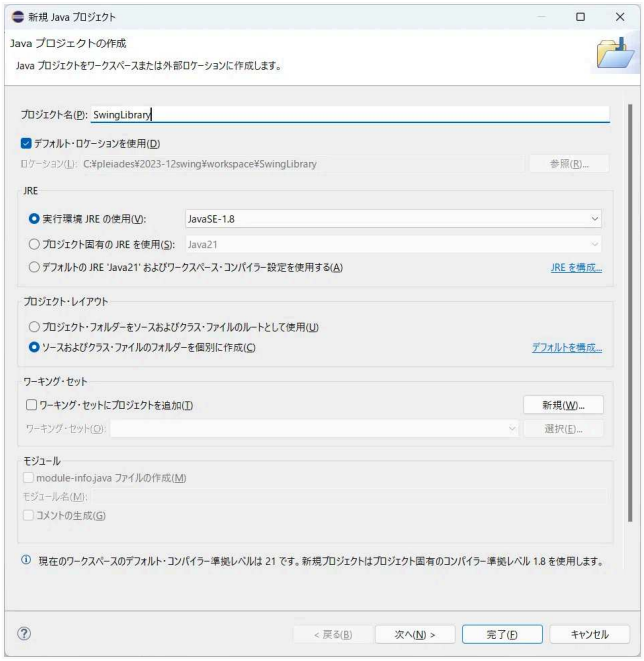

図8．SwingLibraryプロジェクト設定

※作成済みであればこの処理は必要ありません。

以下画面のスクリーンショットはライトテーマで取得します。
（ライトテーマの設定方法）
Eclipseのメニューバーより
　ウィンドウ → 設定 → 一般 → 外観 → ルック＆フィール → ライト
　→ 適用して閉じる → Eclipseの再起動がかかります

外部ファイルの利用準備

(1)jarファイルの準備をします
以下Windows環境を想定しています。
事前に「**提供Library**」フォルダを作成しておきます。
（例　c:￥提供Library）

図３のように提供Libraryフォルダ内に以下のの**２つのjarファイル**を保存しておきます。
・**ハッシュライブラリ：commons-codec-1.15.jar**
・**jdbcドライバ：org.postgresql.driver.jar**

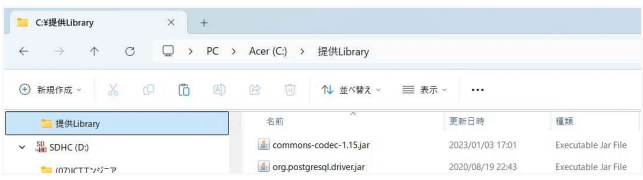

図3．Hashライブラリとjdbcドライバの保存

(2)クラスファイルの準備をします
図４のように提供Libraryフォルダ内に以下の**３つのクラスファイル**を保存しておきます。
（**PostgresConnectionBean.clas、BookLibraryUserCheckBean.class、BookLibraryArrayListBean.class**）
※これらのクラスは「ハッシュログイン機能付き図書の貸出履歴検索システム-JavaEE8,PostgreSQL」で作成したものをコピーして保存します。

その際**パッケージの階層**に従ってください。
（例　c:￥提供Library￥jp￥ict￥aso￥model￥〇〇〇.class）

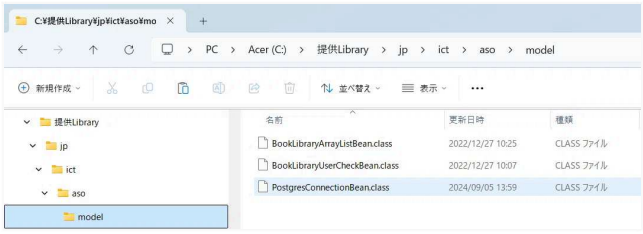

図4．クラスファイルの保存

ビルド・パスに追加

(1)jarファイルを登録します
２つのjarファイルをビルド・パスに追加します。

Eclipseパッケージ・エクスプローラより
SwingLibraryプロジェクトを右クリック→ビルド・パス
→ビルド・パスの構成→「ライブラリ」タブ→外部JARの追加
→Ctrlキーを押しながら**commons-codec-1.15.jar**と**org.postgresql.driver.jar**をクリック→「開く」をクリックします
→最後に「適用して閉じる」をクリック

※**Eclipse2024の場合** → SwingLibraryプロジェクトを右クリック→プロパティ
　→Javaのビルド・パスで設定します。

※図５のように一度設定されていれば再度設定する必要はありません。

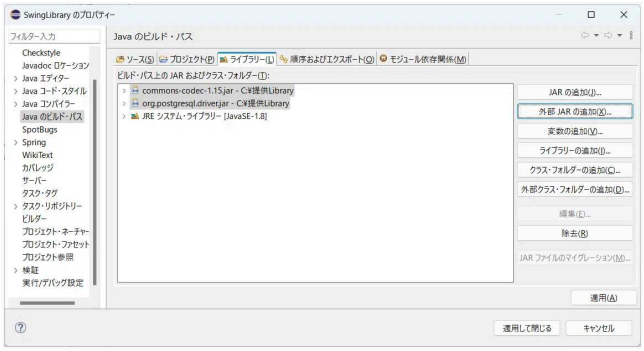

図5．外部JARファイルをビルド・パスに追加

(2)クラスファイルを登録します
３つのクラスファイルをビルド・パスに追加します。

Eclipseパッケージ・エクスプローラより
SwingLibraryプロジェクトを右クリック→ビルド・パス
→ビルド・パスの構成→「ライブラリ」タブ
→外部クラス・フォルダーの追加→「提供Library」のフォルダを選択
→最後に「適用して閉じる」をクリック

※**Eclipse2024の場合** → SwingLibraryプロジェクトを右クリック→プロパティ
→Javaのビルド・パスで設定します。

※図６のように一度設定されていれば再度設定する必要はありません。

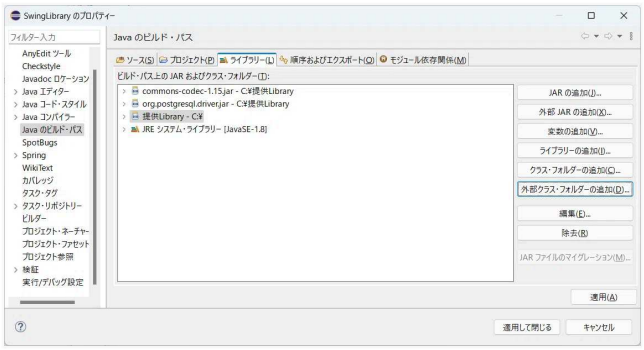

図6．外部クラスファイルをビルド・パスに追加

この時点で「参照ライブラリ」の状況は図7．のようになっています。

図7．参照ライブラリの状況

実装

ひな形の作成

WindowBuilderを用いてSwingアプリケーションのスケルトン（骨格）を自動生成させます。

Eclipseパッケージ・エクスプローラより
SwingLibraryプロジェクトを右クリック→新規→その他
→ WindowBuilder → Swingデザイナー → JFrameを選択 → 次へ

以下の内容で作成
　　パッケージ：jp.ict.aso.swing
　　名前：Library

GUI実装

自動生成されたプログラム（スケルトン）からGUIのデザインを実装します。

パレットと構造（コンポーネント、プロパティ）のViewを利用するのがコツです。デザインイメージは設定反映の参考としてとらえた方が良いでしょう。

①画面中央下部にあるデザインタブでソースコード編集画面からSwingデザイナーに切り替えます。

②contentPaneのLayoutプロパティをBorderLayoutに設定します。

③contentPaneの「North」「South」「Center」の順番でGUI部品のJPanelをパレットから配置します。各パネルのプロパティのConstrainsは「**panel**が**North**」「**panel_1**が**South**」「**panel_2**が**Center**」となります。

④「**Center**」位置のJPanel(**panel_2**)のプロパティのLayoutを**BoxLayout**に変更し、Layoutの＋を展開しConstructorの＋を展開した**axisの設定をY_AXIS**にします。

⑤「**North**」位置のJPanel(**panel**)のプロパティのLayoutを**GridLayout**に変更し、Layoutの＋を展開した**columsの設定を４**にします。

⑥「**North**」位置のJPanel(**panel**)にGUI部品のJLabelを４つ、JTextFieldを３つ、JPasswordFieldを１つ、図８の順番でパレットから配置します。各部品のtextプロパティも図８のように変更します。

⑦「**South**」位置のJPanel(**panel_1**)にGUI部品のJButtonを２つ、パレットから配置します。各部品のtextプロパティを図８のように変更します。

⑧「**Center**」位置のJPanel(**panel_2**)にGUI部品のJScrollPaneを配置しプロパティのholizontalScrollBarPolicyとverticalScrollBarPolicyをそれぞれ
　HORIZONTAL_SCROLLBAR_ALWAYS

VERTICAL_SCROLLBAR_ALWAYS
に変更します。

⑨「**Center**」位置のJPanel(**panel_2**)にGUI部品のJEditorPaneをパレットから配置します。このとき**デザインイビュー内でViewportの位置**に配置します。さらに、プロパティの**contentType**を**text/html**に変更します。

ここまでのGUIの設計状況（部品配置）は図８のようになります。

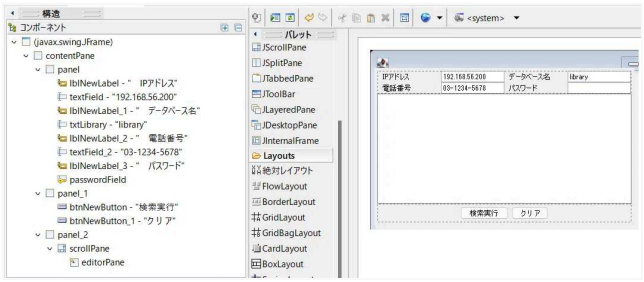

図8．GUI設計状況（部品配置）

⑩ボタンにイベントリスナーを対応付けます。パレットのSwingActions内にある「新規」のリスナーを選択してボタンをクリックすることで対応付けられます。

最終的なデザインは図９のようになります。

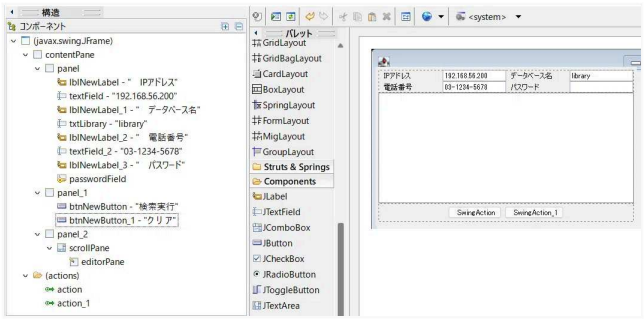

図9．最終デザイン

イベント実装

ソースタブに変更します。

①「検索実行」ボタンのイベントのソース部分を変更します。

```
private class SwingAction extends AbstractAction {
    public SwingAction() {
        putValue(NAME, "検索実行");
        putValue(SHORT_DESCRIPTION, "図書の貸出状況を検索します");
    }
    public void actionPerformed(ActionEvent e) {
        String ip = textField.getText();
        String db = txtLibrary.getText();
        String tel = textField_2.getText();

        char[] password = passwordField.getPassword();
        String pass = new String(password);
        String hashPass = DigestUtils.sha256Hex(pass);  //ハッシュ値を求める

        // DB接続用コネクション作成
         PostgresConnectionBean pcb = new PostgresConnectionBean(ip,db);
            Connection conSearch=pcb.getConnection();
            // ユーザチェック
            BookLibraryUserCheckBean blu = new BookLibraryUserCheckBean();
            blu.setConnection(conSearch);
            blu.setTel(tel);
            blu.userCheck();
         // パスワードチェックOK
          if(hashPass.equals(blu.getPassword())) {
            // DB検索（電話番号）
            BookLibraryArrayListBean bla = new BookLibraryArrayListBean();
            bla.setConnection(conSearch);
            bla.setTel(tel);
            bla.bookSearch();
            // DB接続破棄
            pcb.releaseConnection(conSearch);//DB接続断
            //一旦表示領域クリア
            editorPane.setText("");
            //検索結果出力
            editorPane.setText(bla.getResultset());
         // パスワードチェックNG
          }else {
            // DB接続破棄
            pcb.releaseConnection(conSearch);//DB接続断
            // エラーメッセージを表示
            editorPane.setText("貸出し履歴を取得することができません");
          }
    }
}
```

図10．検索実行ボタンのイベント内容

②「クリア」ボタンのイベントのソース部分を変更します。

```
private class SwingAction_1 extends AbstractAction {
    public SwingAction_1() {
        putValue(NAME, "ク リ ア");
        putValue(SHORT_DESCRIPTION, "検索表示結果をクリアします");
    }
    public void actionPerformed(ActionEvent e) {
        editorPane.setText(null);
        textField_2.setText(null);
        passwordField.setText(null);
    }
}
```

図11．クリアボタンのイベント内容

④フィールド変数を変更します。//kokoの部分を追加します。

```
private static final long serialVersionUID = 1L;
private JPanel contentPane;
private JTextField textField;
private JTextField txtLibrary;
private JTextField textField_2;
private JPasswordField passwordField;
private final Action action = new SwingAction();
private final Action action_1 = new SwingAction_1();

private JEditorPane editorPane;    //koko

/**
 * Launch the application.
 */
```

図12．フィールド変数の変更

⑤ローカル変数の宣言になっている部分を変更します。**JEditorPaneのローカル宣言を削除**します。//kokoの部分を変更します。

```
    JScrollPane scrollPane = new JScrollPane();
    panel_2.add(scrollPane);

    editorPane = new JEditorPane();        //koko
    editorPane.setContentType("text/html");
    scrollPane.setViewportView(editorPane);
}
```

図13．ローカル変数の変更

⑥実行確認します。エディタの画面内で右クリック → 実行 → Javaアプリケーションで実行されます。

図14．初期起動画面

⑦パスワードを入れて検索実行ボタンをクリックします。フレームを広げて検索結果の全体を確認してください。

利用者電話番号	氏名	所属	書名	貸出日	返却日
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	1999-12-12	1999-12-13
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	1999-12-12	
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	2001-10-03	2002-10-23
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	2002-10-26	2002-10-25
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	2002-10-26	
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	2002-10-26	
03-1234-5678	山田隆	営業	構造化分析とシステム仕様	2002-10-26	
03-1234-5678	山田隆	営業	WindowsNT4.0システム管理入門（基礎編）	1998-07-07	1998-08-07
03-1234-5678	山田隆	営業	総合化MRPシステム－設計と導入-	1998-07-07	1998-08-07
03-1234-5678	山田隆	営業	わかりやすいデータベース設計技法	2000-02-02	2001-02-02

図15．検索実行画面

実装変更

電話番号「045-666-7777」や「045-123-4567」のユーザで検索結果が問題なければ以下の変更を行ってください
・起動時のフレームの大きさを検索結果の全体が表示できるように変更
・フレームにタイトルを設定
・表示フォントを見やすく変更

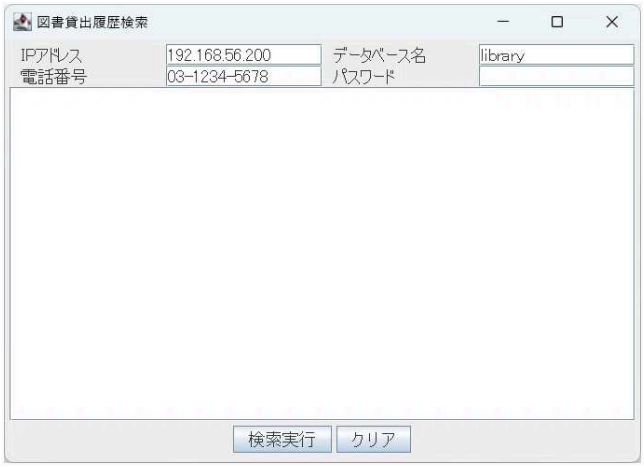

図16．完成画面

これで、完成です。

単独起動

実行可能JARファイル

①せっかくですので、単独で起動できるアプリケーションにエクスポートしましょう。Java1.8以上のJREの環境がWindowsのPCにインストールされていればダブルクリックで起動できます。

Eclipseパッケージ・エクスプローラより
SwingLibraryプロジェクトを右クリック → エクスポート
→ Java → 実行可能JARファイル → 次へ
→ 以下のように設定する → 完了

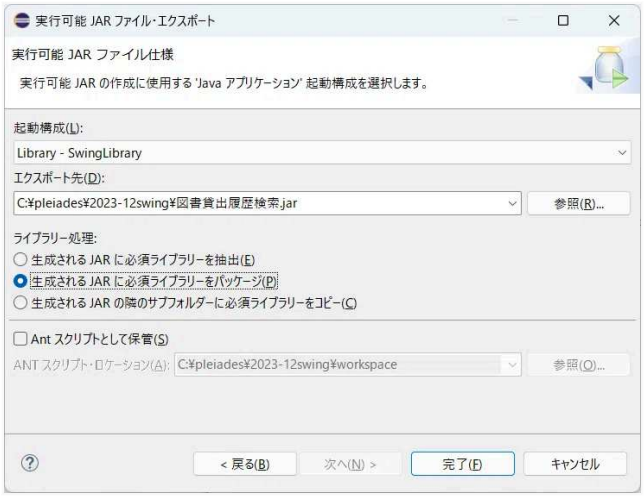

図17．実行可能JARファイルの設定

以下のような警告が出る場合がありますが、気にしません。

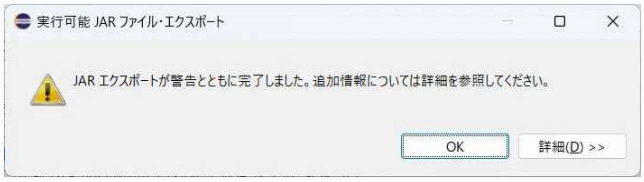

図18．警告ダイアログ

作成されたjarファイルをダブルクリックするとアプリが起動します。

図19．実行可能JARファイル

最後に

今回は、GUIデザイン部分の設定の詳細は提示していません。また、実装変更部分のソースコード例も載せていません。自由に工夫してみてください！