



Swingによるデスクトップアプリケーション開発（元本投資）-JavaSE1.8



office・M

2024年12月5日 12:09

...

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）を使って投資アプリケーションを作成します。元本が年複利で一定年数たつと利息が元加されてどれくらいの金額になっているかを計算します。計算ロジックはWebアプリケーション開発で利用した独自クラスを再利用します。

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

GUI設計のポイント

内部設計

処理ロジック

提供クラス

実装準備

プロジェクトの作成

外部classの利用準備

ビルド・パスに追加

実装

ひな形の作成

GUI実装

イベント実装

実装変更

単独起動

実行可能JARファイル

最後に

外部設計

WindowBuilderのSwingデザイナーで図1のようなGUIを作成します。



図1. 元本投資シミュレーション画面

GUI設計のポイント

元本投資の経過をシミュレーションするBeanは履歴をhtml形式でテキストフィールドに保存しています。よって、表示する際はhtmlのタグを解釈できる**JEditorPane**を使います。プロパティの**contentType**を **text/html** に設定することでhtmlによる文字飾り等が利用できるようになります。

シミュレーションを行う場合、運用期間の入力値によって結果の出力量が変わってきます。よって、JEditorPanelは**JScrollPane**内で縦スクロールが可能のように配置します。その際Window枠全体で表示されるよう中央パネルのレイアウトは**BoxLayout**に変更しておきます。

内部設計

元本投資の計算と履歴表示を行う外部クラス（InvestmentSimBean.class）を『MVCモデルによるWebアプリケーション開発（No 5.元本投資）-EE8』の記事より取得してください。Eclipseのビルドパスに追加することで、当該クラスを利用できるようになります。

処理ロジック

- ・元本、利率、運用期間を入力し計算ボタンをクリックすることでアクションイベントを発生させます。
- ・イベント内で入力値から外部クラス（InvestmentSimBean.class）に引数を与えて実体化し元本投資のシミュレーションを行います。
- ・シミュレーション結果は表示領域に張り付けます。

提供クラス

提供される元本投資計算ロジック（InvestmentSimBean.class）の使い方とクラス図は以下のようになります。

（使い方）

- ・ InvestmentSimBeanクラスを**元本と利率と期間**の3つの引数をもつコンストラクタで実体化します。
- ・ simulationメソッドを実行することでsimフィールドに運用収益の履歴がhtml文字列で表組されます。
- ・ 利用側のクラスはgetSim()メソッドで運用収益の履歴を取得します。

（クラス図）

package:jp.ict.aso.model	
InvestmentSimBean	
- ganpon:int	//入力単位は円（元本）
- riritu:double	//入力単位は%の実数値（利率）
- kikan:int	//入力単位は年（期間）
- sim:String	//運用収益の履歴（html形式）
+ InvestmentSimBean(ganpon:int,riritu:double,kikan:int):	
+ simulation():void	//シミュレーション実施
+ getSim():String	//結果の取得
+ getGanpon():int	//設定された元本取得
+ getRiritu():double	//設定された利率取得
+ getKikan():int	//設定された期間取得
-:private +:public #:protected	

図2. InvestmentSimBeanクラス図

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→「SwingMoney」プロジェクトを作成する

→図3の内容で設定する

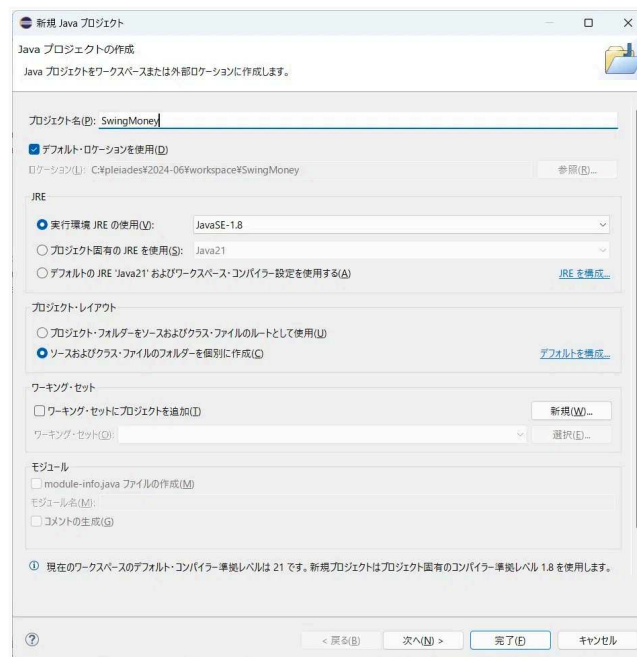


図3. SwingMoneyプロジェクト設定

※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classの利用準備

以下Windows環境を想定しています。

事前に「提供Beanフォルダ」を作成しておきます（例 c:\提供Bean）。

提供Beanフォルダ内に**InvestmentSimBean.class**を保存しておきます。

その際**パッケージの階層**に従ってください。

(例 c: ¥ 提供Bean ¥ jp ¥ ict ¥ aso ¥ model ¥ InvestmentSimBean.class)

ビルド・パスに追加

InvestmentSimBean.classをEclipseのビルド・パスに追加します。

Eclipseパッケージ・エクスプローラより

プロジェクトを右クリック→プロパティ→「Javaのビルド・パス」を選択

→「ライブラリ」タブ→「外部クラス・フォルダーの追加」ボタンクリック

→「提供Bean」を選択→最後に「適用して閉じる」をクリック

※図4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4. 外部クラス・フォルダの追加

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

SwingMoney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 : Investment

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Window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内にGUI部品のJPanelを3つ、パレットから配置します。JPanelのConstraintsプロパティ（配置する位置）はそれぞれ「North」「Center」「South」とします。

④「North」位置のJPanelのLayoutをGridLayout変更し3列の設定にします。

⑤「North」位置のJPanelにGUI部品のJLabel、JTextFieldをパレットから配置します。

⑥「North」位置のJPanelのJLabel部品のプロパティのtextを設定します。設定値は**図1の外部設計を参照**してください。

⑦「Center」位置のJPanelのLayoutを**BoxLayout**変更しGUI部品のJScrollPaneをパレットから配置します。

⑧「Center」位置のscrollPaneの中にGUI部品のJEditorPaneをパレットから配置します。このとき**デザインイメージのViewportの位置**に配置します。

⑨htmlを使った文字装飾が可能なように、「Center」位置のJEditorPaneのcontentTypeプロパティをtext/htmlに変更します。

⑩「South」位置のJPanelにGUI部品のJButtonを2つ、パレットから配置します。あわせて、表示テキストを**図1の外部設計のように変更**します。

⑪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それぞれのボタンをクリックすることで対応付けられます。

イベント実装

ソースタブに変更します。

①「計算ボタン」のイベントのソース部分を以下のヒントを参考に変更します。

```
private class SwingAction extends AbstractAction {
    public SwingAction() {
        putValue(NAME, "計 算");
        putValue(SHORT_DESCRIPTION, "元本投資シミュレーションを実行します");
    }
    public void actionPerformed(ActionEvent e) {
        //元本・利率・運用期間をBeanの仕様通りに変換
        int ganpon = Integer.parseInt(textField.getText())*10000;//単位が万円
        double riritu = Double.parseDouble(textField_1.getText())/100.0;//単位が%
        int kikan = Integer.parseInt(textField_2.getText());

        //一旦表示領域クリア
        sim="";
        editorPane.setText(sim);

        // シミュレーション結果の作成
        InvestmentSimBean isb = new InvestmentSimBean(ganpon, riritu, kikan);
        isb.simulation();
        sim=isb.getSim();
        editorPane.setText(sim);
    }
}
```

図5. イベントソース部分のヒント

②「クリアボタン」のイベントのソース部分を変更します。

③フィールド変数を変更します。

④ローカル変数の宣言になっている部分を変更します。

⑤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6. 起動画面

⑥元本・年率・期間のパラメータを入力して計算ボタンのクリックで投資シミュレーションの結果が最後まで表示されるか確認します。

年目	元本	元金
7年目	¥21,189	¥1,021,189
8年目	¥24,253	¥1,024,253
9年目	¥27,326	¥1,027,326
10年目	¥30,408	¥1,030,408

図7. シミュレーション結果表示

シミュレーション結果は「野村証券のマネーシミュレータ（みらい電卓）」のWebサイトと全く同じ結果になりますので以下のurlで確認してください。

利息元加処理：年複利 元加方式：四捨五入 期間：10年
利率：0.3% 元本：100万円

<https://www.nomura.co.jp/hajimete/simulation/unyou.html>



図8. 野村証券のマネーシミュレータ（みらい電卓）

※サイトのシミュレーションと実行結果がぴったり一致するはずですが

ここまでの実装ではいろいろと不具合があるようです。

とりあえず「タイトルがない」「シミュレーション結果の表示領域が狭い」「画面の大きさが任意に変えられてしまう」「元本・年率・期間のパラメータの初期値を入力しておく」の4点を修正します。

実装変更

①フレームにタイトルを追加し画面の大きさを調整します。追加します。**デザインイメージからでも変更可能です。**

②画面（フレーム）の大きさを固定します。

③起動時に初期値を入力しておきます。**デザインイメージからでも変更可能です。**

④実行して動きを確認します。これで完成しました。

元本投資シミュレータ

運用するお金（元本） 100 万円
 想定利回り（年率） 0.3 %
 運用期間 10 年
 利息元加処理：年複利 元加方式：四捨五入 税抜き

野村証券(元本投資) 年複利 四捨五入 非課税

期間	運用収益	元本+運用収益
1年目	¥3,000	¥1,003,000
2年目	¥6,009	¥1,006,009
3年目	¥9,027	¥1,009,027
4年目	¥12,054	¥1,012,054
5年目	¥15,090	¥1,015,090
6年目	¥18,135	¥1,018,135
7年目	¥21,189	¥1,021,189
8年目	¥24,253	¥1,024,253
9年目	¥27,326	¥1,027,326
10年目	¥30,408	¥1,030,408

計算 クリア

図9. 元本投資シミュレータ完成

単独起動

実行可能JARファイル

①せっかくだので、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
 Swing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

実行可能 JAR ファイル・エクスポート

実行可能 JAR ファイル仕様

実行可能 JAR の作成に使用する 'Java アプリケーション' 起動構成を選択します。

起動構成(L):
 Investment (1) - SwingMoney

エクスポート先(D):
 C:\pleiades\2024-06\元本投資シミュレーション.jar 参照(B)...

ライブラリ処理:
☐ 生成される JAR に必須ライブラリを抽出(E)
☒ 生成される JAR に必須ライブラリをパッケージ(D)
☐ 生成される JAR の隣のサブフォルダーに必須ライブラリをコピー(C)

☐ Ant スクリプトとして保存(S)
 ANT スクリプト・ロケーション(A): C:\pleiades\2024-06\workspace 参照(O)...

? < 戻る(B) 次へ(N) > 完了(F) キャンセル

図10. 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11. 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シミュレーションが起動します。



図12. 実行可能JARファイル

最後に

以上でSwingデザイナーを使って元本投資のシミュレーションを行うデスクトップアプリケーションを作成できました。

ただし、このアプリを実装するにはInvestmentSimBean.classが必要です。このクラスは『MVCモデルによるWebアプリケーション開発（No 5 .元本投資）-EE8』の記事から取得可能です。