Swingによるデスクトップアプリケーション開発（マネーシミュレータ）-JavaSE1.8

office・M
2024年9月29日 10:05

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）を使ってマネーシミュレータを作成します。元本投資・積立投資・貸付返済の３つのシミュレーション機能を持っています。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。

『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』を終了していることが前提条件です。

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

『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』の外部設計をもとにしてWindowBuilderのSwingデザイナーで図1のようなGUIを作成します。

図1. マネーシミュレータ画面

以下のように元本投資、積立投資、貸付返済のシミュレーション機能を持たせます。

図1.1 元本投資シミュレーション画面

図1.2 積立投資シミュレーション画面

図1.3 貸付返済シミュレーション画面

GUI設計のポイント

実行経過の履歴はhtml形式でテキストフィールドに保存されています。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（or返済金額）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、JEditorPaneは**JScrollPane**内で縦スクロールが可能なように配置します。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。

『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』をもとに上部パネルのテキスト領域をラジオボタンに変更してシミュレーションの機能を選択するようにします。入力のテキストフィールドは変更せずに、機能ごとに意味を持たせることにします。

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（No５.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。

積立投資の計算と履歴表示を行う外部クラス（AccumulationSImBean.class）を『MVCモデルによるWebアプリケーション開発（No６.積立投資）-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。

貸付返済の計算と履歴表示を行う外部クラス（LoanSImBean.class）を『MVCモデルによるWebアプリケーション開発（No７.貸付返済）-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

・ラジオボタンの選択で元本投資、積立投資、貸付返済のシミュレーション機能を選択します。
・元本（借入金）、利率（手数料率）、運用期間（返済金額）の入力領域はそれぞれの機能で共有します。
・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それぞれの機能の外部クラスに引数を与えて実体化し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）、積立投資計算ロジック（AccumulationSimBean.class）、貸付返済計算ロジック（LoanSim.class）の使い方とクラス図は以下のようになります。

元本投資計算ロジック（InvestmentSimBean.class）

（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）

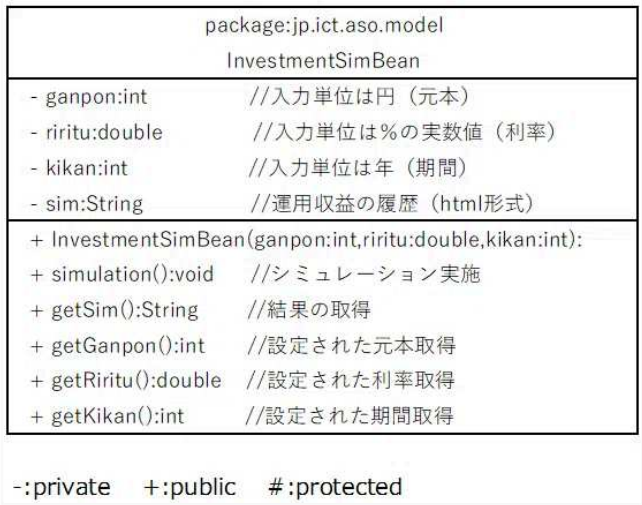

図2．InvestmentSimBeanクラス図

積立投資計算ロジック（AccumulationSimBean.class）

（使い方）
・AccumulationSimBeanクラスを**積立金額**と**想定利回り**と**積立期間**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）

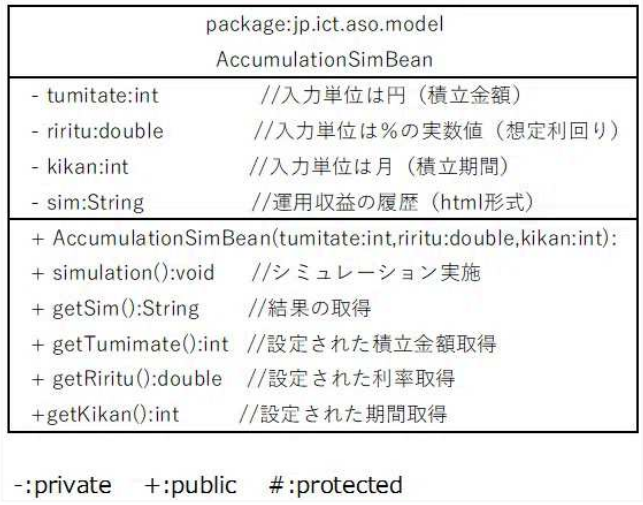

図2．AccumulationSimBeanクラス図

貸付返済計算ロジック（LoanSim.class）

（使い方）
・LoanSimBeanクラスを**借入残額**と**手数料率（年利率）**と**返済金額**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フィールドに借入残金の履歴がhtml文字列で表組されます。
・利用側のクラスはgetSim()メソッドで借入残金の履歴を取得します。

（クラス図）

図3．LoanSimBeanクラス

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４の内容で設定する

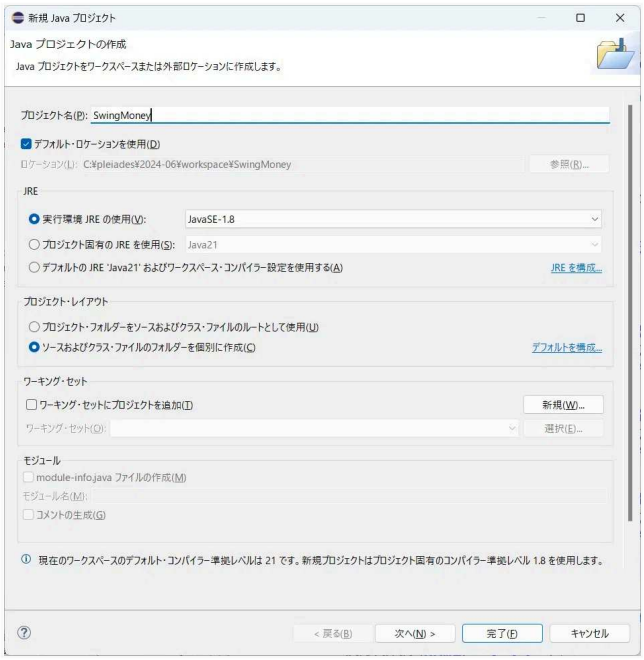

図4．SwingMoneyプロジェクト設定

※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、AccumulationSimBean.class、LoanSim.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￥〇〇〇.class）

ビルド・パスに追加

InvestmentSimBean.class、AccumulationSimBean.class、LoanSim.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5．外部クラス・フォルダの追加

実装

GUI実装

『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』で作成したプログラムを変更して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図6のデザインイメージようにGUIを変更します。

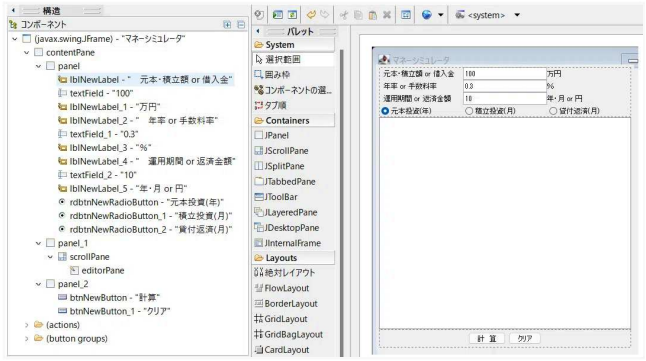

図6．変更後の画面デザイン

イベント実装

ソースタブに変更します。

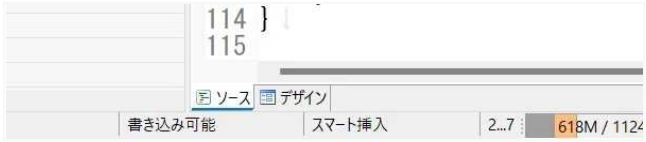

図7．ソースタブに変更

①「計算ボタン」のイベントのソース部分を変更します。ラジオボタンの選択状況により実体化するBeanを変更します。
以下の元本投資のソースコードをヒントに考えましょう！

```
if(rdbtnNewRadioButton.isSelected()) {
    //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,ririt u,kikan);
            isb.simulation();
            sim=isb.getSim();
            editorPane.setText(sim);
        }
```

元本投資のシミュレーション結果は参照先１に示す「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。

（参照先１）

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　野村証券
※実行結果とぴったり同じになります
https://www.nomura.co.jp/hajimete/simulation/unyou.html

積立投資のシミュレーション結果は参照先２に示す「カシオ」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。

（参照先２）

利息元加処理：月複利　元加方式：四捨五入　期間：１年
利率：0.3%　積立額：５万円　非課税　カシオ
※実行結果とぴったり同じになります
https://keisan.casio.jp/exec/system/1254841870

また、金融庁のwEBサイトでも確認可能です。

（参照先３）

金融庁　資産運用シミュレーション
https://www.fsa.go.jp/policy/nisa2/moneyplan_sim/index.html

貸付返済のシミュレーション結果は参照先４に示す「アプラス」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。

（参照先４）

アプラス（新生銀行）　元利定額リボルビング払いシミュレーション
https://www.aplus.co.jp/creditcard/revo/simulation/index.html

②ラジオボタンの選択状況によりシミュレーションが変更されることを確認して完成です。

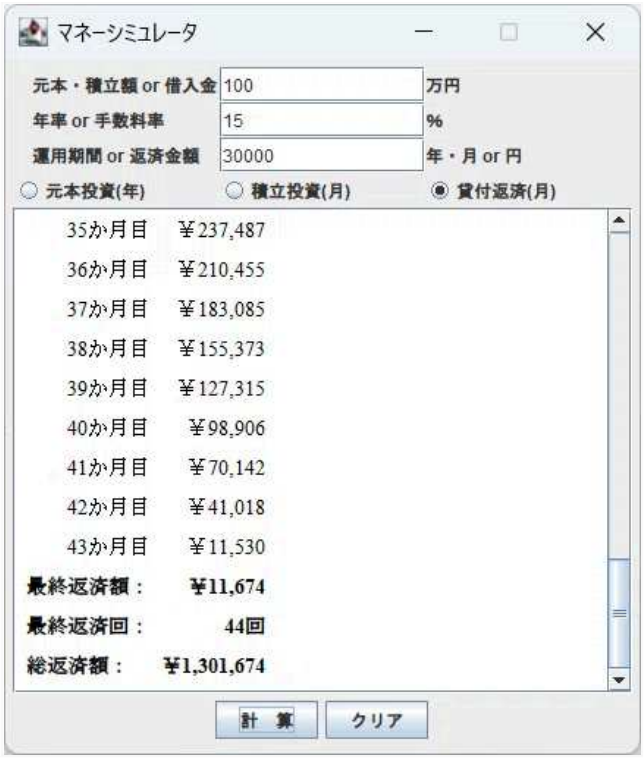

図8．マネーシミュレータ完成

提出課題

以下の条件は「NISAで長期積み立てを行うシミュレーション」の例です。
条件に従い今回のプログラムでシミュレーションを行い結果を提出してください。
（条件）
毎月積み立て金額：5,000円(0.5万円)
想定利回り(年率)：3.0%
積立期間：20年(240カ月)

※金融庁の「つみたてシミュレータ」によるとざっくりと164万円になるようですが、今回のプログラムを使って１円の単位まで計算してください。

図9．金融庁つみたてシミュレータの例

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
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

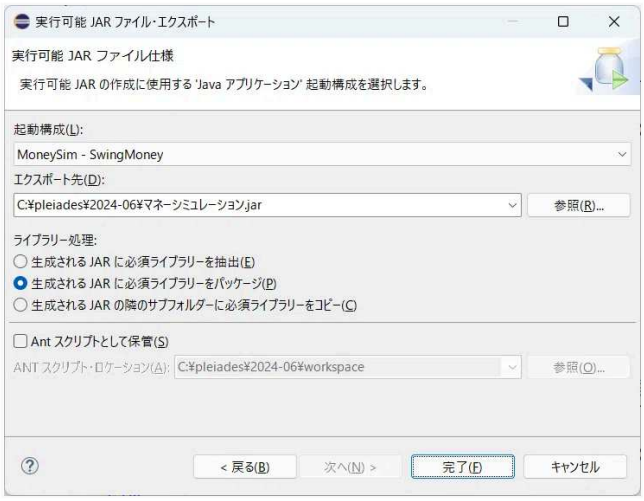

図10．実行可能JARファイルの設定

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。

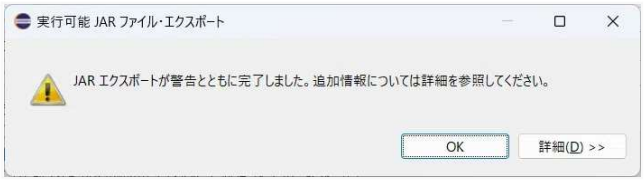

図11．警告ダイアログ

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、積立投資、貸付返済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それぞれのシミュレーションを実行するクラスファイルが必要です。このクラスは
『MVCモデルによるWebアプリケーション開発（No５.元本投資）-EE8』『MVCモデルによるWebアプリケーション開発（No６.積立投資）-EE8』『MVCモデルによるWebアプリケーション開発（No７.貸付返済）-EE8』
の３つの記事から取得可能です。

また、『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』を終了していることが前提条件です。