

Swingによるデスクトップアプリケーション開発 (円ドル変換) -JavaSE1.8



Swingによるデスクトップアプリケーション開発（円ドル変換） - JavaSE1.8

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）を使って為替レートを使って円とドルの通貨換算を行うアプリケーションを作成します。

2025年8月よりECLIPSEのバージョンを最新版（Version: 2025-06 (4.36.0)）に変更しました。



外部設計

WindowBuilderのSwingデザイナーで図1のようなGUIを作成する。

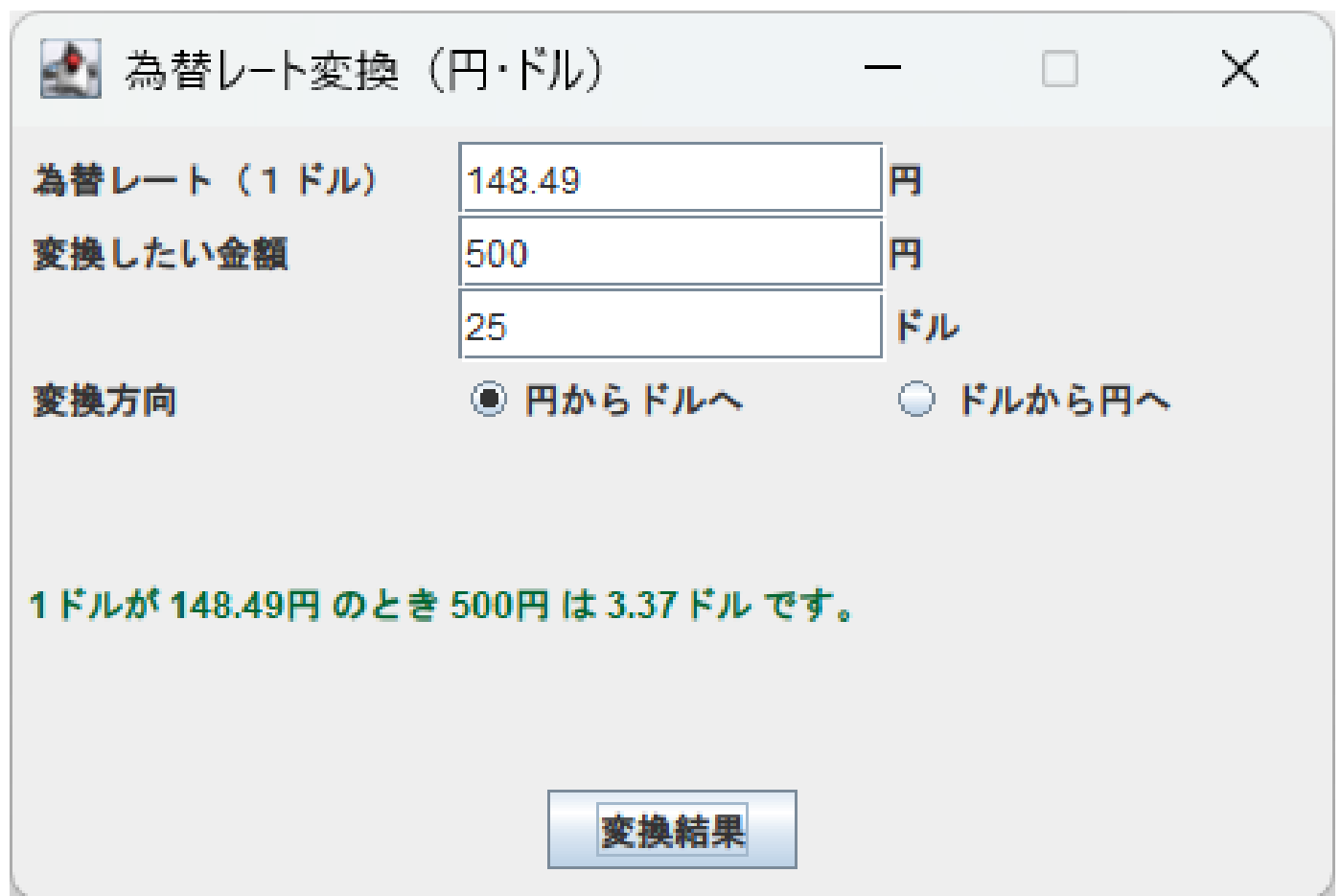


図1. 円ドル変換アプリケーションのGUI画面

内部設計

処理ロジック

変換ボタンをクリックすることでアクションイベントが発生し、為替レートをもとにした円からドルまたはドルから円の変換処理ロジックを実装したメソッドが起動する。円・ドルの変換方向はラジオボタンで選択する。

①変換処理ロジックとクラス図

1.円からドルへ

●計算式

$$\text{ドル} = \frac{\text{円の金額}}{\text{為替レート (円/ドル)}}$$

●YenDollarBean.java (円からドルへ変換クラス)

package名：jp.ict.aso.model

YenDollarBean
- yen:double - dollar:double - rate:double
+ YenDollarBean() + calcDollar():void + setYen(yen:double):void + setRate(rate:double):void + getRate():double + getYen():double + getDollar():double

- private + public

2.ドルから円へ

●計算式

円 = ドルの金額 × 為替レート（円/ドル）

●DollarYenBean.java（ドルから円へ変換クラス）

jp.ict.aso.model.DollarYenBean
- yen:double - dollar:double - rate:double
+ DollarYenBean() + calcYen():void + setDollar(dollar:double):void + setRate(rate:double):void + getRate():double + getYen():double + getDollar():double

- private + public

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→「SwingYen」プロジェクトを作成→図2の内容で設定する

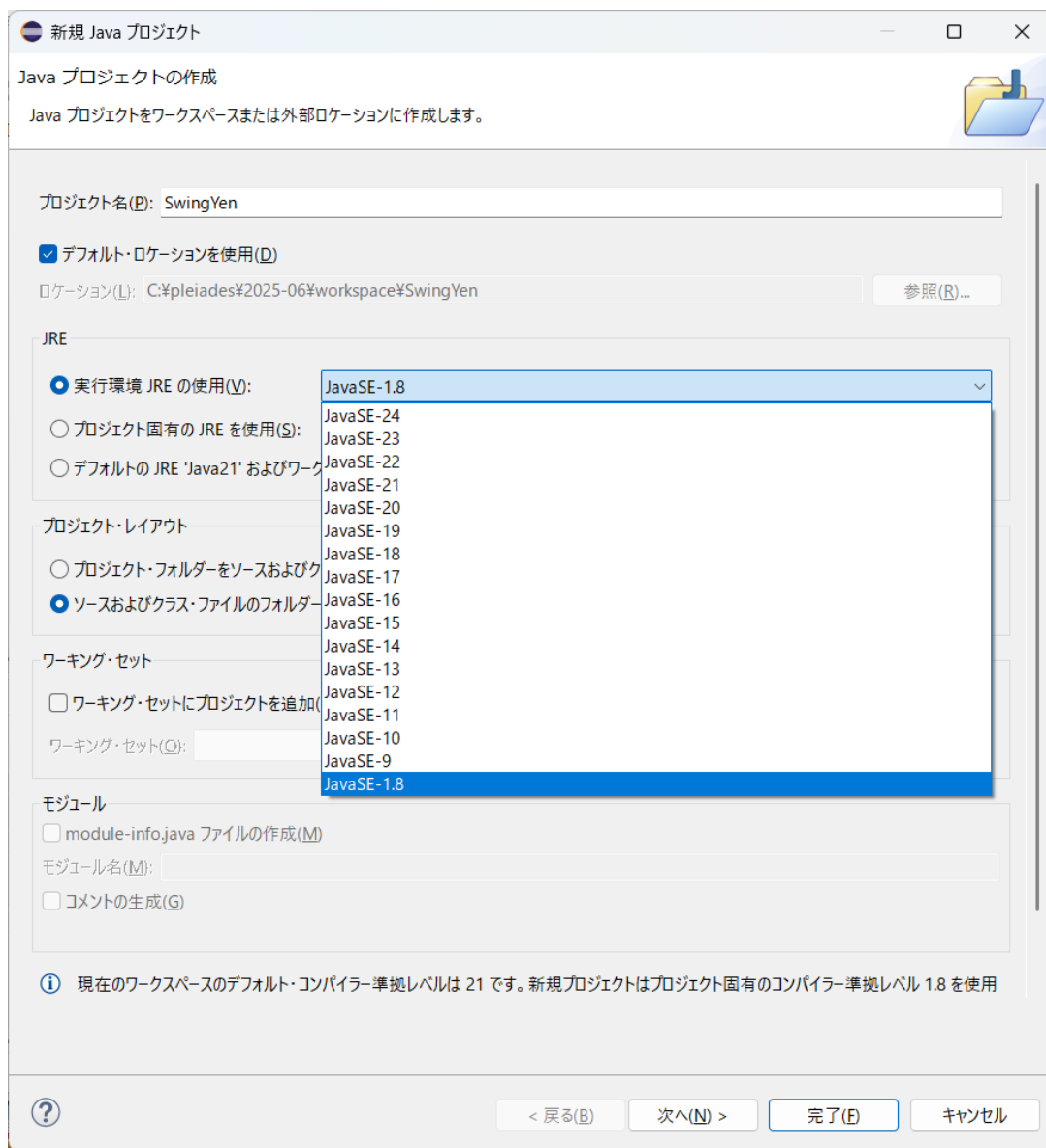


図2. SwingYenプロジェクト設定

※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

実装

modelの準備

為替変換の計算処理を行うJavaBeans（クラス）をプロジェクト内に用意しないといけません。

以下の2つのクラス（Webアプリケーションで作成するものと同じBean）をSwingYenプロジェクトに保存（コンパイル）しておいてください。

YenDollarBean.java

```
package jp.ict.aso.model; ↓
import java.io.Serializable; ↓
public class YenDollarBean implements Serializable{ ↓
    private double yen,dollar,rate; ↓
    public YenDollarBean() {} ↓
    public YenDollarBean(double yen,double rate) { ↓
        this.yen = yen; ↓
        this.rate= rate; ↓
    } ↓
    public void calcDollar() { ↓
        // dollar=yen/rate; ↓
        dollar=((double)Math.round((yen/rate)*100)/100); ↓
    } ↓
    public double getRate() { ↓
        return rate; ↓
    } ↓
    public void setRate(double rate) { ↓
        this.rate = rate; ↓
    } ↓
    public double getYen() { ↓
        return yen; ↓
    } ↓
    public void setYen(double yen) { ↓
        this.yen = yen; ↓
    } ↓
    public double getDollar() { ↓
        return dollar; ↓
    } ↓
}
```

DollarYenBean.java

```
package jp.ict.aso.model; ↓
import java.io.Serializable; ↓
public class DollarYenBean implements Serializable{ ↓
    private double yen,dollar,rate; ↓
    public DollarYenBean() {} ↓
    public DollarYenBean(double dollar,double rate) { ↓
        this.dollar = dollar; ↓
        this.rate= rate; ↓
    } ↓
    public void calcYen() { ↓
        yen=((double)Math.round((dollar*rate)*100)/100); ↓
    } ↓
    public double getRate() { ↓
        return rate; ↓
    } ↓
    public void setRate(double rate) { ↓
        this.rate = rate; ↓
    } ↓
    public double getYen() { ↓
        return yen; ↓
    } ↓
    public void setDollar(double dollar) { ↓
        this.dollar = dollar; ↓
    } ↓
    public double getDollar() { ↓
        return dollar; ↓
    } ↓
}
```

GUI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

SwingYen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

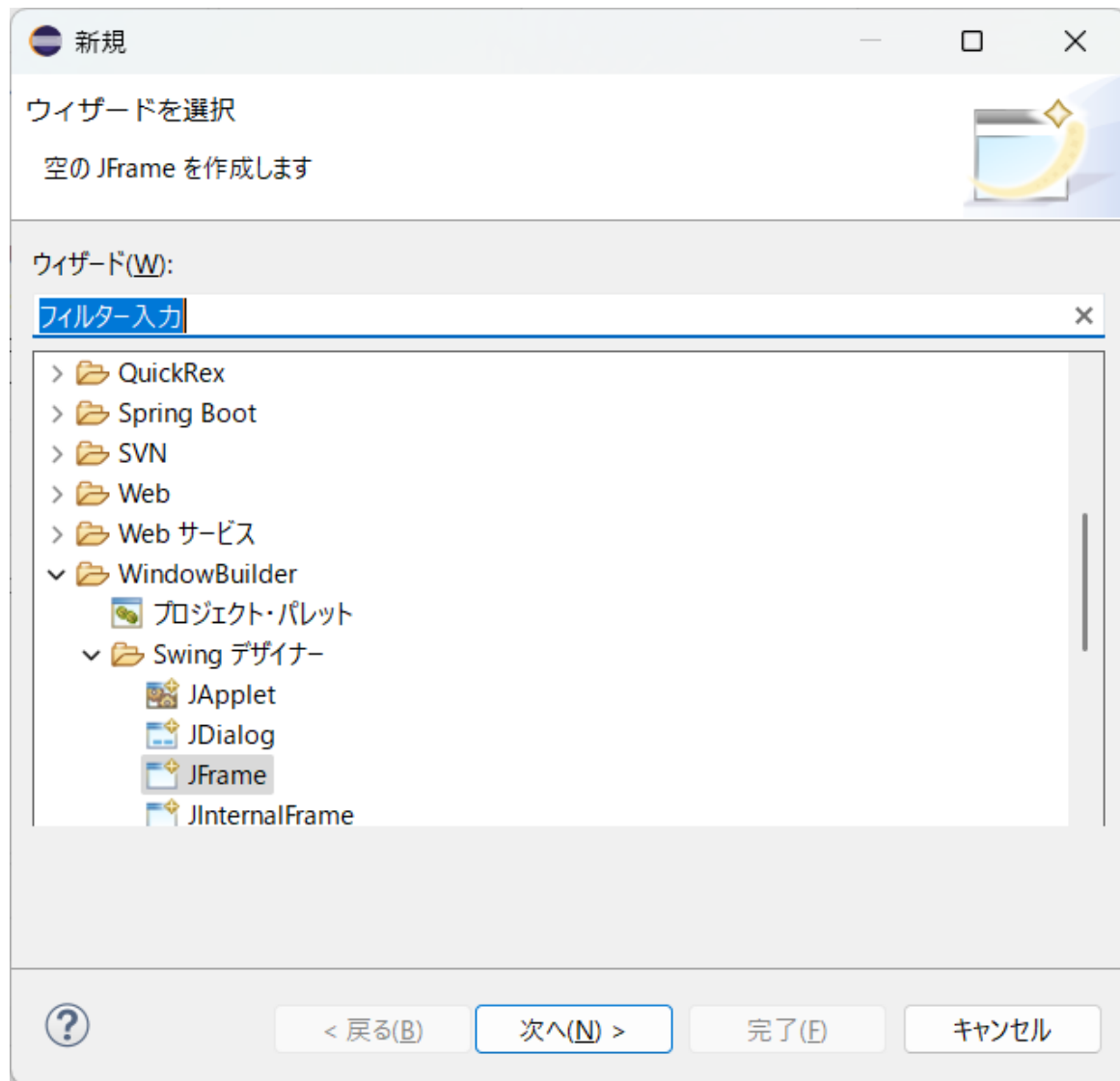


図3. JFrameウィザードの選択

以下の内容で作成

パッケージ：jp.ict.aso.swing

名前：KawaseHenkan

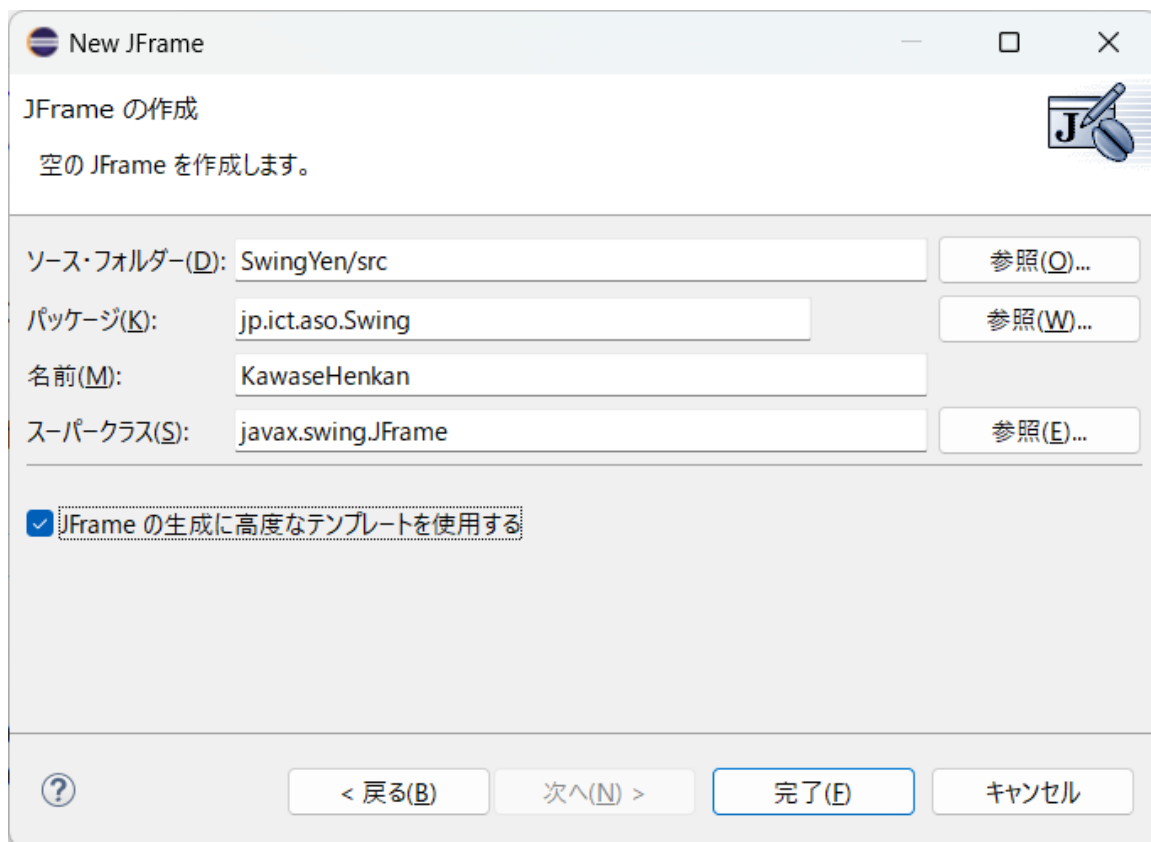


図4. Swingアプリケーションの生成

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。

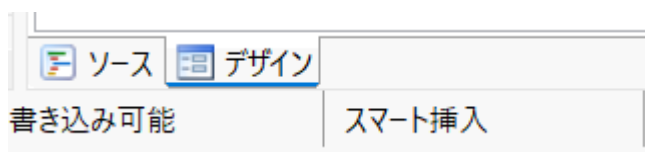


図5. Swingデザイナーに切り替え

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

③contentPaneの「North」の位置にGUI部品のJPanelをパレットから配置します。JPanelのLayoutプロパティはGridLayoutとします。

④JPanelのLayoutプロパティを + マークから展開しcolumnsを3に設定します（図6. 参照）。

⑤JPanelにGUI部品のJLabel、JTextField、JRadioButtonを順番にパレットから配置します。それぞれのtextプロパティを図6.のように変更します。

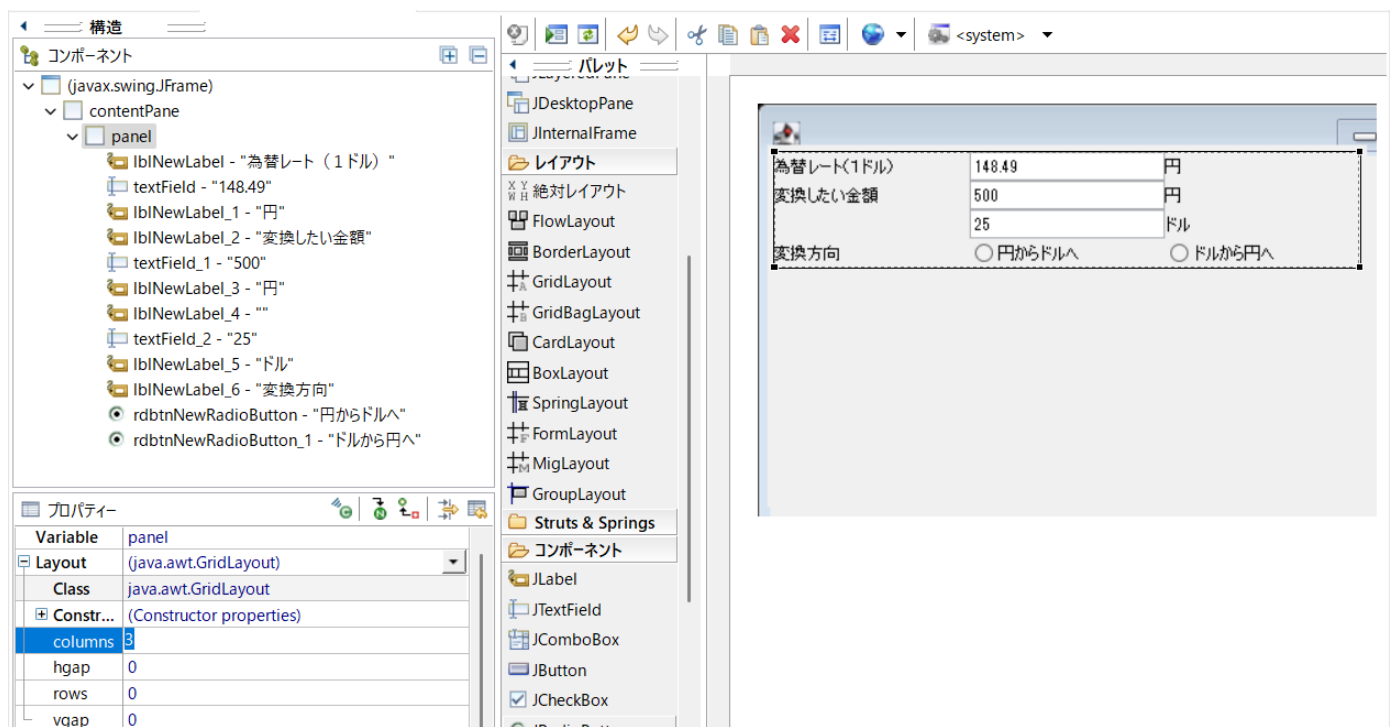


図 6. 「North」位置のGUI部品

⑥contentPaneの「Center」の位置にGUI部品のJPanelをパレットから配置します。あわせてLayoutプロパティをBoxLayoutに変更します。

⑦JPanelにGUI部品のJLabelをパレットから配置します。textプロパティを図7.のように変更します。

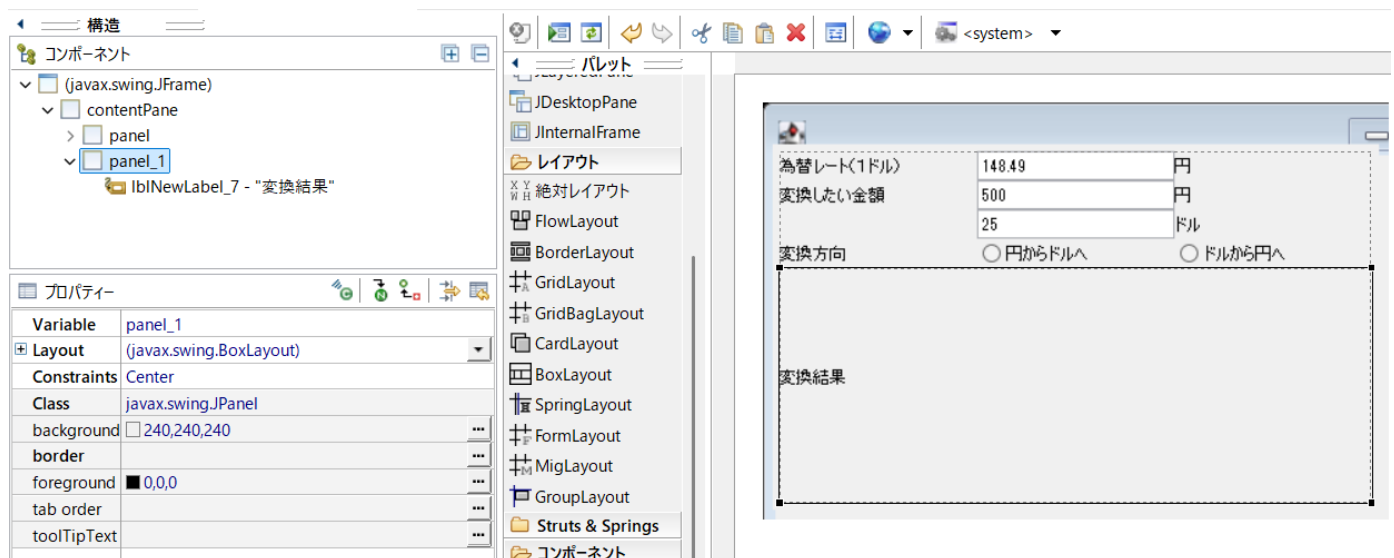


図 7. 「Center」位置のGUI部品

⑧contentPaneの「South」の位置にGUI部品のJPanelをパレットから配置します。

⑨JPanelにGUI部品のJButtonをパレットから配置します。

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。

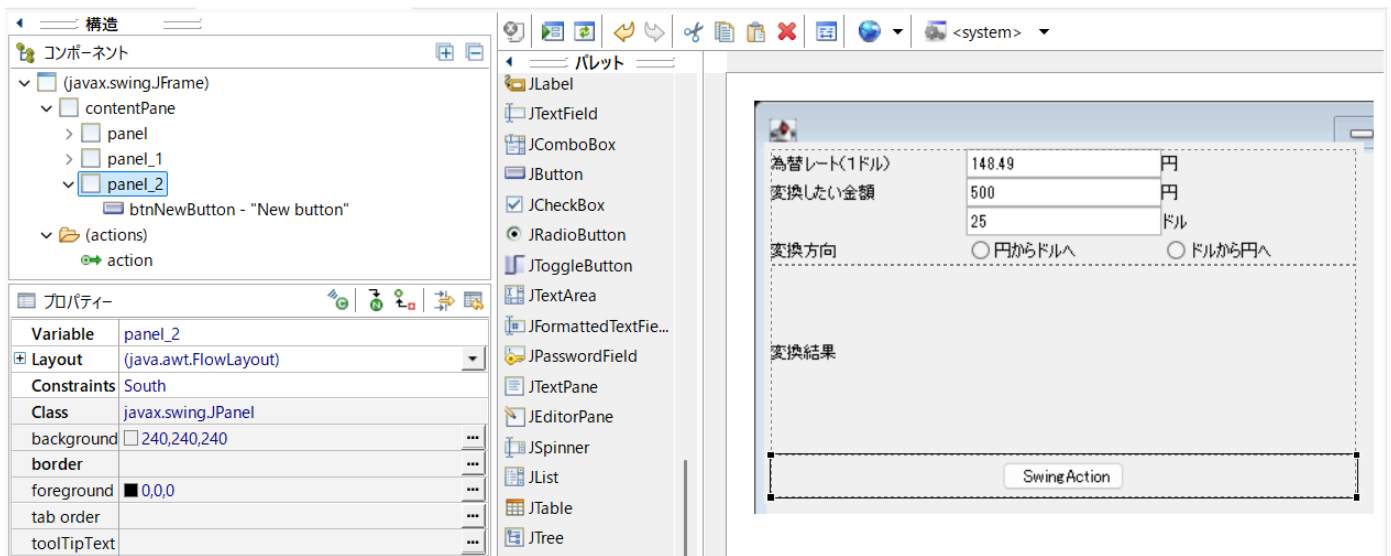


図8. 「South」位置のGUI部品

イベント実装

ソースタブに変更します。

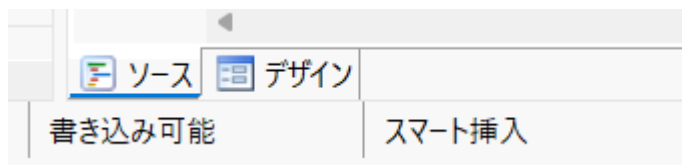


図9. ソースタブに変更

①「変換結果」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, "変換結果");
        putValue(SHORT_DESCRIPTION, "為替レート変換を行います");
    }
    public void actionPerformed(ActionEvent e) {
        try {
            double rate = Double.parseDouble(textField.getText());
            double yen = Double.parseDouble(textField_1.getText());
            double dollar = Double.parseDouble(textField_2.getText());
            lblNewLabel_7.setForeground(Color.BLACK);
            if (rdbtnNewRadioButton.isSelected()) {
                YenDollarBean yd = new YenDollarBean(yen, rate);
                yd.calcDollar();
                lblNewLabel_7.setForeground(new Color(0, 102, 51)); // 深い緑
                lblNewLabel_7.setText("1ドルが "+rate+"円 のとき "+(int)yen+"円 は "
                    + yd.getDollar() + "ドル です。");
            } else {
                DollarYenBean dy = new DollarYenBean(dollar, rate);
                dy.calcYen();
                lblNewLabel_7.setForeground(new Color(0, 170, 51)); // 深い緑
                lblNewLabel_7.setText("1ドルが "+rate+"円 のとき "+dollar+"ドル は "
                    + (int)dy.getYen() + "円 です。");
            }
        } catch (NumberFormatException ex) {
            lblNewLabel_7.setForeground(Color.RED);
            lblNewLabel_7.setText("数値を正しく入力してください!");
        }
    }
}
```

図10. 為替レート変換の実行結果ボタンのイベント内容

②フィールド変数を変更します。//kokoの部分を追加します。

```
public class KawaseHenkan extends JFrame {  
    private static final long serialVersionUID = 1L;  
    private JPanel contentPane;  
    private JTextField textField;  
    private JTextField textField_1;  
    private JTextField textField_2;  
    private final Action action = new SwingAction();  
  
    private JLabel lblNewLabel_7; //koko  
    private JRadioButton rdbtnNewRadioButton; //koko  
    private JRadioButton rdbtnNewRadioButton_1; //koko
```

図11. フィールド変数の変更

③ローカル変数の宣言になっている部分を変更します

あわせてラジオボタンをグループ化します。//kokoの部分を変更します。

```
rdbtnNewRadioButton = new JRadioButton("円からドルへ"); //koko  
panel.add(rdbtnNewRadioButton);  
  
rdbtnNewRadioButton_1 = new JRadioButton("ドルから円へ"); //koko  
panel.add(rdbtnNewRadioButton_1);  
  
ButtonGroup buttongp = new ButtonGroup(); //koko  
buttongp.add(rdbtnNewRadioButton); //koko  
buttongp.add(rdbtnNewRadioButton_1); //koko  
  
JPanel panel_1 = new JPanel();  
contentPane.add(panel_1, BorderLayout.CENTER);  
panel_1.setLayout(new BoxLayout(panel_1, BoxLayout.X_AXIS));  
  
lblNewLabel_7 = new JLabel("変換結果"); //koko  
panel_1.add(lblNewLabel_7);
```

図12. ローカル変数の宣言変更

④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。

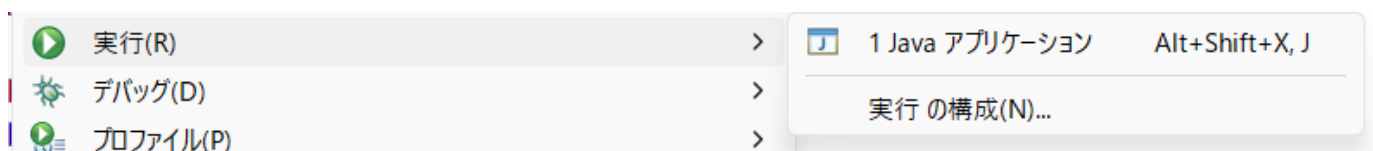


図13. 実行方法

⑤変換結果ボタンのクリックで結果が表示されるか確認します。

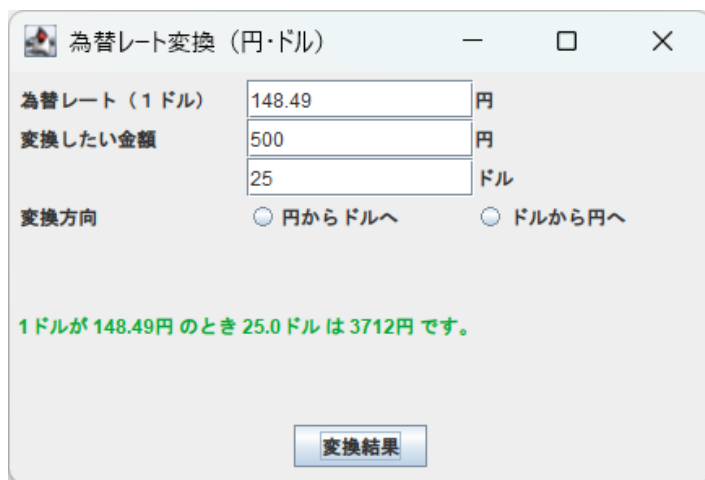


図14. 為替レート変換の実行結果の表示

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「ラジオボタンが初期選択されていない」「画面の大きさが任意に変えられてしまう」の2点を修正します。

実装変更

①デザインタブのコンポーネントのrdbtnNewRadioButtonのプロパティのselectedをチェックしてtrueに変更します。

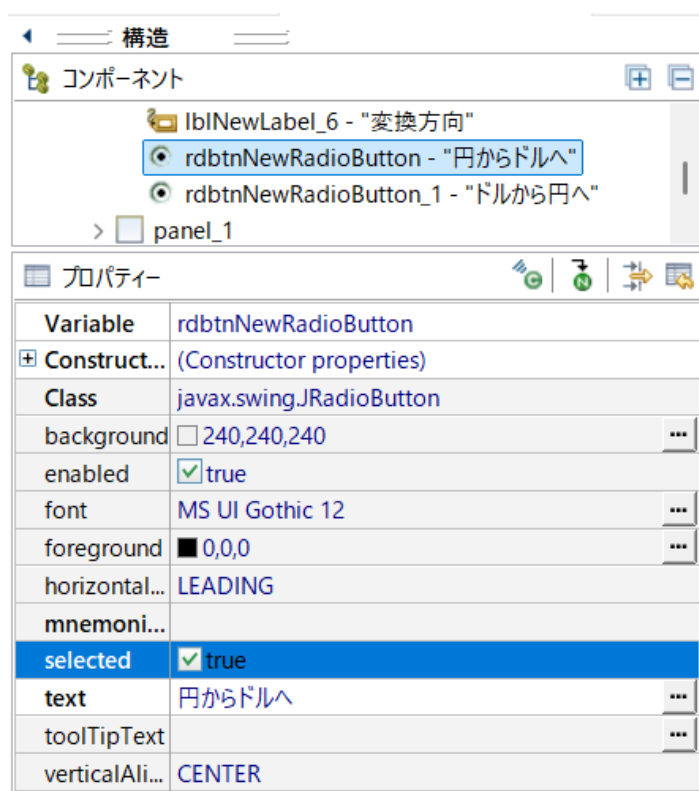


図15. ラジオボタンの初期選択

②ソースタブより画面（フレーム）の大きさを固定します。//kokoの部分を追加します。

```

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                KawaseHenkan frame = new KawaseHenkan();
                frame.setVisible(true);
                frame.setResizable(false); //koko
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

```

図16. 画面サイズの固定

③実行して動きを確認します。これで一応完成ですが、、、

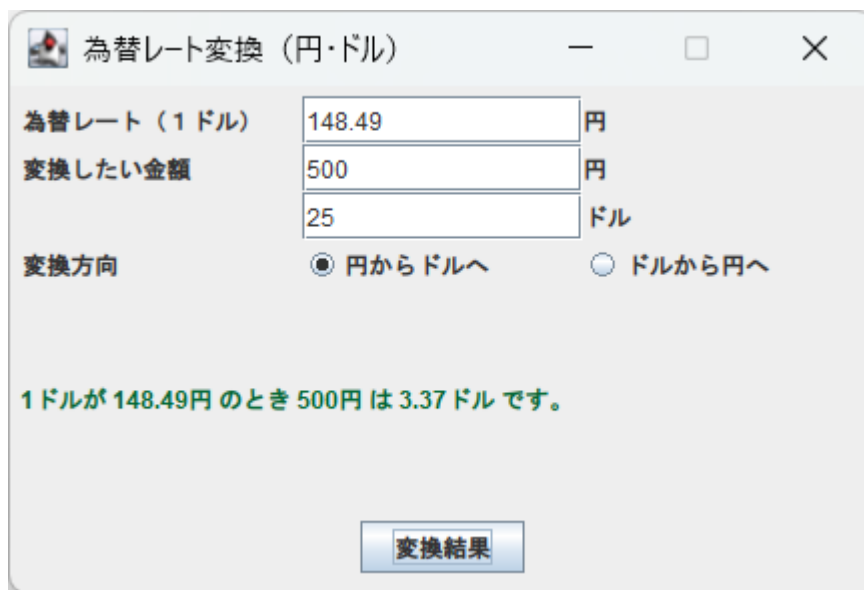


図17. 為替レート変換アプリケーション完成

演習

①例によってGUIを少しかっこよくしましょう。

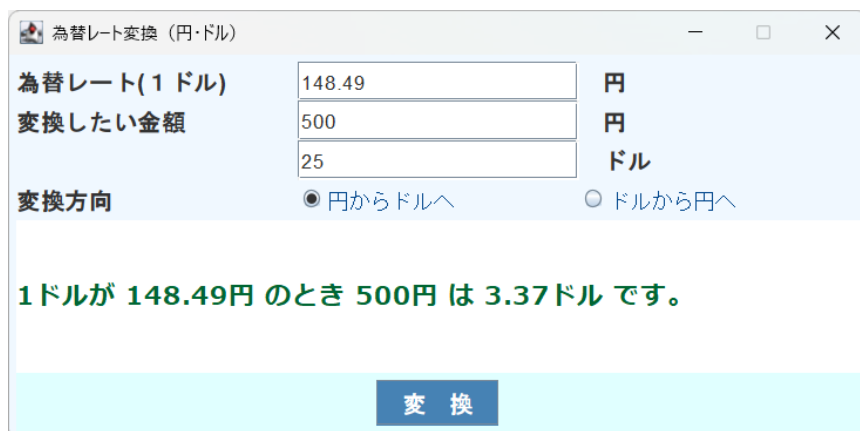


図18. 為替レート変換アプリケーションGUI装飾版

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

SwingYen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

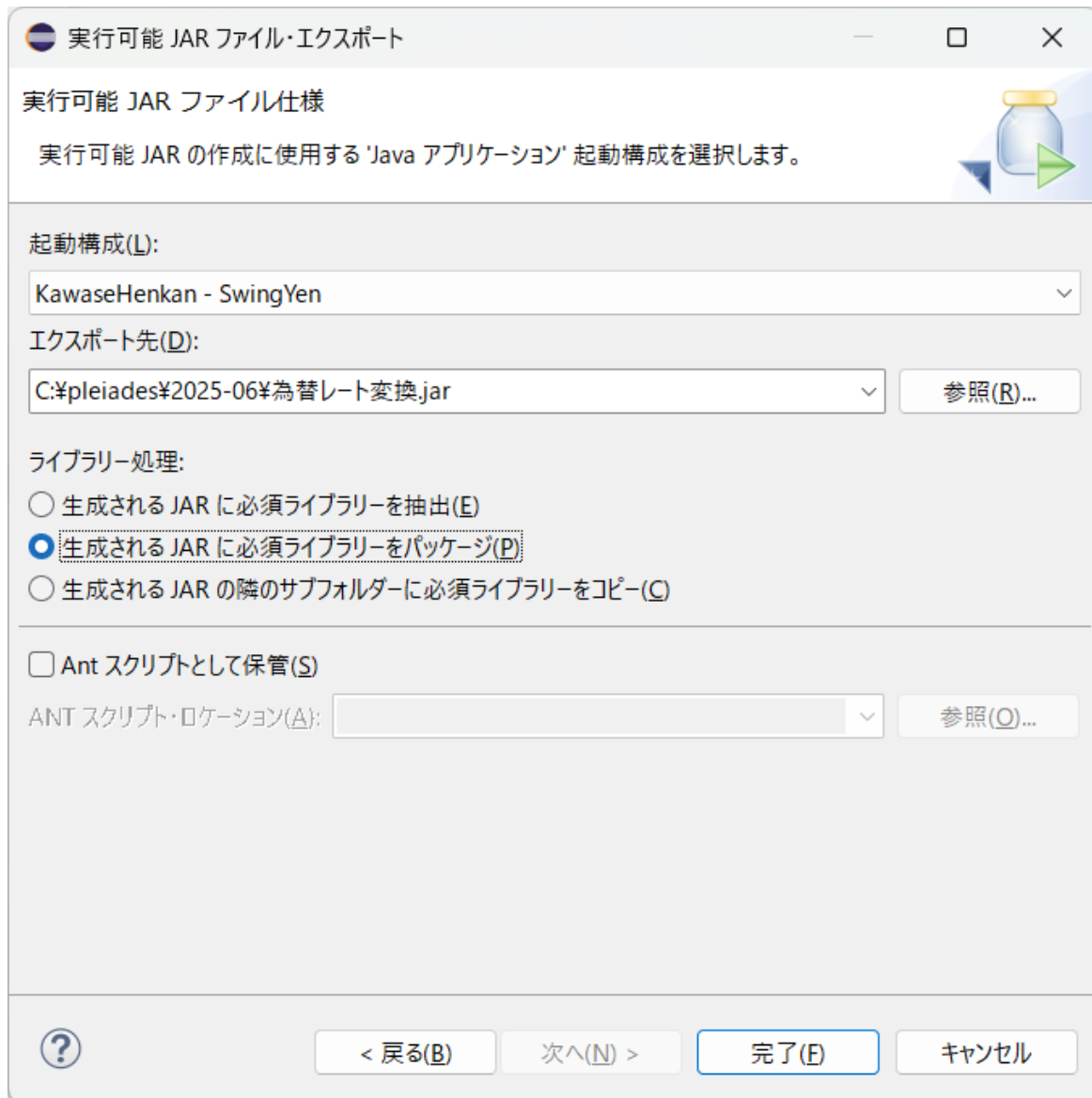


図19. 実行可能JARファイルの設定

以下のような警告ダイアログが出る場合がありますが無視します。

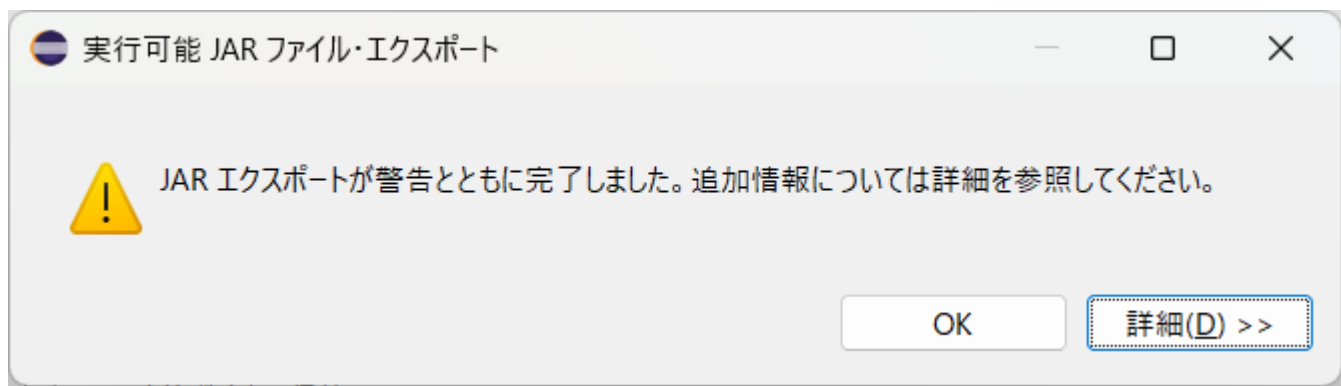


図20. 警告ダイアログ

作成されたjarファイルをダブルクリックすると為替レート変換アプリが起動します。



図21. 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円・ドルの為替レート変換を行うデスクトップアプリを作成できました。