MVCモデルによるWebアプリケーション開発（No３.円ドル変換）-EE8

office・M
2024年1月10日 10:33

¥1,000 •••

JavaEE8（JSP・Servlet）の環境でWeb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。今回は入力と出力で画面が遷移しない題材（円ドル変換）をMVCモデルで実装してみます。

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

開発概要

JavaEE8アプリケーション・サーバ（Tomcat）を利用して開発を行うということを前提に実装を進めていきます。開発ツールには統合開発環境のEclipseを用いることにします。

題材として日本円を米ドルに変換するWebアプリケーションを作成します。入力画面に為替レートと日本円を入力すると、同じ画面内に米ドルの金額が表示されます。入力画面と変換結果の出力画面には同じJSPを使うため、GETリクエスト時にもBeanの実体化が必要になります。

JavaEE8環境ですので使用言語は当然Java（JSP・Servlet）となり、設計技法にはMVCモデルを使います。MVCモデルとはWebアプリケーションの構成をModel（業務ロジック）ーView（表示）ーController（制御）に分割して設計する技法です。３つのモデルに役割分担することで部品化が促され、ひいてはチーム開発に貢献します。

実装していく過程でEclipseの利用方法が不明な場合は、別途メール等でお知らせください。この講座では、作成済みの仕様と設計書から「**JavaEE8環境でのMVCモデルの実装の仕組みを学ぶ**」ことを目的とします。

開発方針

Webアプリケーションの開発環境には図１のようにEclipse2024を用います。

図1. 開発環境 Eclipse 2024

Spring等のフレームワークは使用しません。これは、MVCモデルにおけるhttpプロトコルの処理の実装を直に学んでいただきたいためです。

開発に必要な仕様と最小限の設計ドキュメントは提示します。この情報をもとに、まずは、自分で試行錯誤しながら実装してみてください。

設計仕様

【円ドル変換の仕様】
以下の条件を具現化するWebアプリケーションの作成を行います。
（条件）
・urlを指定してサイトにアクセスすると２つの入力欄（為替レート、日本円）と計算ボタンが配置された画面を表示します
・「計算ボタン」をクリックすると同じ画面内に米ドルの金額を表示します
・入力する為替レートは小数点２桁です
・入力する日本円は整数値です
・出力する米ドルは、小数点３桁目の四捨五入で小数点２桁の表示です

【Model（処理ロジック）の設計方針】
以下の仕様で実装します。
（仕様）
・米ドル＝日本円÷為替レート
・米ドルは小数点３桁目を四捨五入します
　　Javaの例：double dollar=((**double**)Math.**round**((yen/rate)*100)/100);
（実装）
・JavaBeansの仕様に基づきYenDollarBean.javaを作成します
・package名は jp.ict.aso.model とします
・作成するクラスはSerializableインターフェースを実装します
・変数はprivate宣言します
・引数なしのコンストラクタを実装します
・適切な引数を与えるコンストラクタを実装し、引数はインスタンス変数に格納します
・処理ロジックを実装します、このとき処理結果はインスタンス変数に格納します

・setterメソッドを実装します
・getterメソッドを実装します

外部設計

接続urlは**http://localhost:8080/libcon/Yen**とします。よってEclipseの動的Webプロジェクトの名前は**libcon**となり、サーブレットのurlパターンは**Yen**となります。

Eclipseのメニューバーより
ファイル→新規→動的Webプロジェクト→「libcon」プロジェクトを作成する
→ 図1.1の内容で設定する

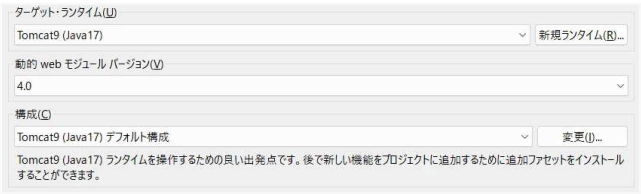

図1.1 動的Webプロジェクトの設定

※作成済みであればこの処理は必要ありません

---

urlにアクセスすると図２のような、起動画面が表示されます。為替レートと日本円を入力して「計算ボタン」をクリックすると、図３のように米ドルの金額が計算され入力値と共に画面下部に表示されます。この時、新たな画面へ遷移はしません。

図２．起動画面

図３．計算結果画面

内部設計

クラス連携図は図４のようになります。

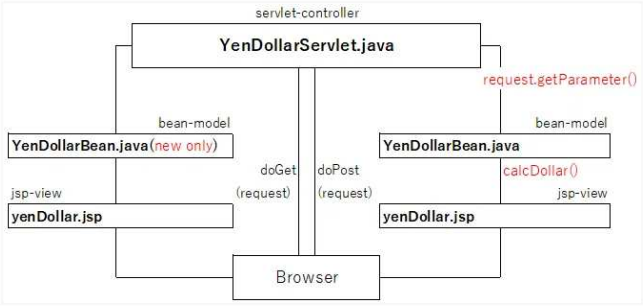

図４．MVCモデル図（クラス連携図）

円ドル変換ロジック（YenDollarBean）のクラス図は図4.1のようになります。

package名：jp.ict.aso.model

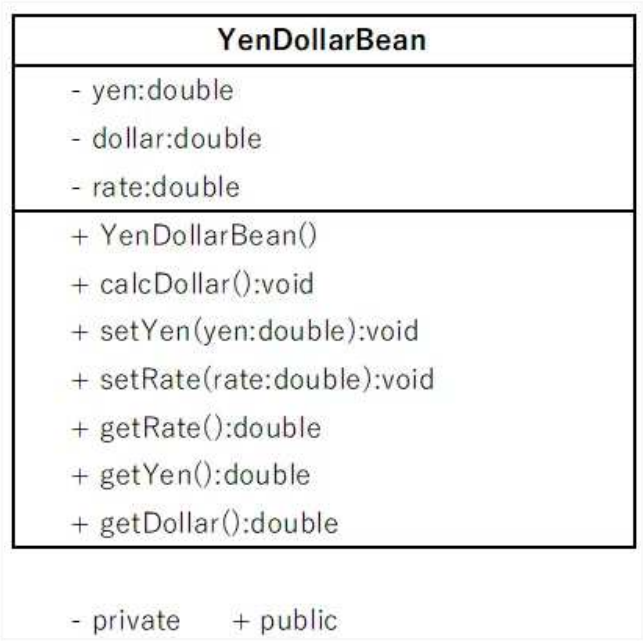

図4.1 クラス図

実装手順

１．円ドル変換ロジック用JavaBeansクラスを作成します

Eclipseパッケージ・エクスプローラより
libconプロジェクトを右クリック→新規→クラス
→以下の内容で作成

・YenDollarBean.java
　　パッケージ：jp.ict.aso.model
　　名前：YenDollarBean
　　ソースコード：考えましょう！

２．リクエストコントロール用Servletクラスを作成します

Eclipseパッケージ・エクスプローラより
libconプロジェクトを右クリック→新規→その他→Web→サーブレット
→以下の内容で作成する

・YenDollarServlet.java
　　パッケージ：jp.ict.aso.controller
　　クラス名：YenDollarServlet
　　ソースコード：考えましょう！　※アノテーションは**/Yen**

３．起動画面を表示するJSPファイルを作成します

Eclipseパッケージ・エクスプローラより
libconプロジェクトを右クリック→新規→その他→Web→JSPファイル
→以下の内容で作成する

・yenDollar.jsp
　　保存場所：libcon/src/main/webapp/**WEB-INF/jsp** ← **注意！**

ファイル名：yenDollar.jsp
ソースコード：考えましょう！

実行確認

サーブレットクラス（YenDollarServlet.java）を実行します！！

Eclipseパッケージ・エクスプローラより
YenDollarServlet.javaを右クリック→実行→サーバーで実行
→図５のように実行される

図５．起動画面

ソースコード例

以下に各プログラムのソースコードの例（本文内では「考えましょう！」になっている部分）を示しますので、実装の参考にしてください。

１．YenDollarBean.java

```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;
  }
  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;
  }
}
```

図６．YenDollarBean.java

２．YenDollarServlet.java

```java
package jp.ict.aso.controller;
import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import jp.ict.aso.model.YenDollarBean;

@WebServlet("/Yen")
public class YenDollarServlet extends HttpServlet {

  protected void doGet(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {

    YenDollarBean ydb=new YenDollarBean();
//    ydb.calcDollar();
    request.setAttribute("yendollar", ydb);
    RequestDispatcher rd=request.getRequestDispatcher("/WEB-INF/jsp/yenDollar.jsp");
    rd.forward(request, response);
  }
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {

    String yenStr=request.getParameter("yen");
    String rateStr=request.getParameter("rate");
    double yen=Double.parseDouble(yenStr);
    double rate=Double.parseDouble(rateStr);

```

図７．YenDollarServlet.java（１）

```java
    YenDollarBean ydb=new YenDollarBean(yen,rate);
    ydb.calcDollar();
    request.setAttribute("yendollar", ydb);
    RequestDispatcher rd=request.getRequestDispatcher("/WEB-INF/jsp/yenDollar.jsp");
    rd.forward(request, response);
  }
}
```

図８．YenDollarServlet.java（２）

３．yenDollar.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="jp.ict.aso.model.*"%>
<%
YenDollarBean ydb=(YenDollarBean)request.getAttribute("yendollar");
%>
<html>
<head>
<title>円ドル変換</title>
</head>
<body>

<h1>円からドルへ為替レート変換を行います</h1>
<hr>
  <form method="POST" action="Yen">
    <p>為替レートで1アメリカドルが <input type="number" step="0.01" name="rate">円のとき <br>
      日本円の<input type="number" name="yen">円はいくらになるか計算します
    </p>
    <input type="submit" value="計算">
  </form><br>

【結果】
<ul>
<li>レート: <%= ydb.getRate() %>円</li>
<li>日本円: <%= ydb.getYen() %>円</li>
<li>米ドル: <%= ydb.getDollar() %>ドル</li>
</ul>

</body>
</html>
```

図９．yenDollar.jsp

表示桁数調整

この状態で実行すると図10のように日本円と米ドルの表示の桁数が仕様とあっていません。それぞれ解決します。

図10．日本円と米ドルの表示桁数ミス

１．日本円の表示変更

jspの出力時にintでキャストします。yenDollar.jspの24行目を図11のように変更します。

```
21 【結果】
22<ul>
23 <li>レート: <%= ydb.getRate() %>円</li>
24 <li>日本円: <%= (int)ydb.getYen() %>円</li>
25 <li>米ドル: <%= ydb.getDollar() %>ドル</li>
26 </ul>
```

図11．yenDollar.jsp

２．米ドルの表示変更

Beanで計算する時にroundメソッドで四捨五入します。YenDollarBean.javaの11行目を図12のように変更します。

```
10  public void calcDollar() {
11    dollar=((double)Math.round((yen/rate)*100)/100);
12  }
```

図12．YenDollarBean.java

練習問題

為替レートをもとに米ドルから日本円に変換するDollarYenアプリを作成してください。

（外部設計）

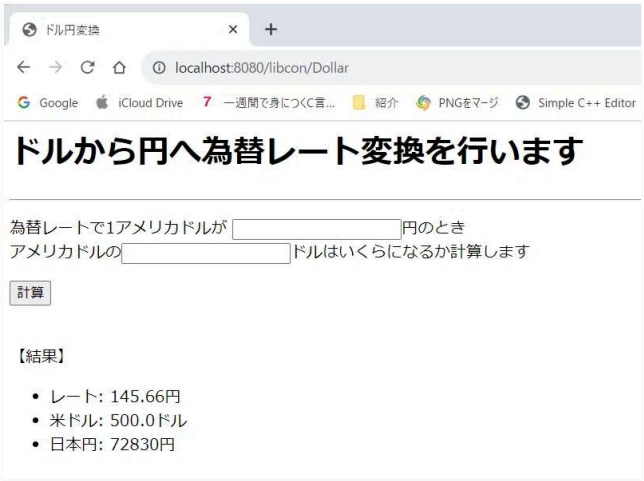

図13. ドル円計算の外部設計

(内部設計)

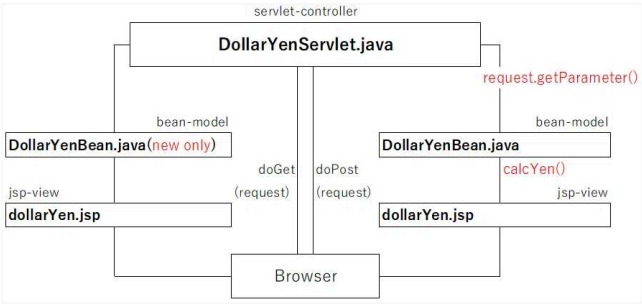

図14. ドル円計算のクラス連携図

DollarYenBean.java → calcYen()で単純に掛け算するだけ
DollarYenServlet.java → urlパターン：/Dollar
dollarYen.jsp → 円の出力は小数点以下切り捨て

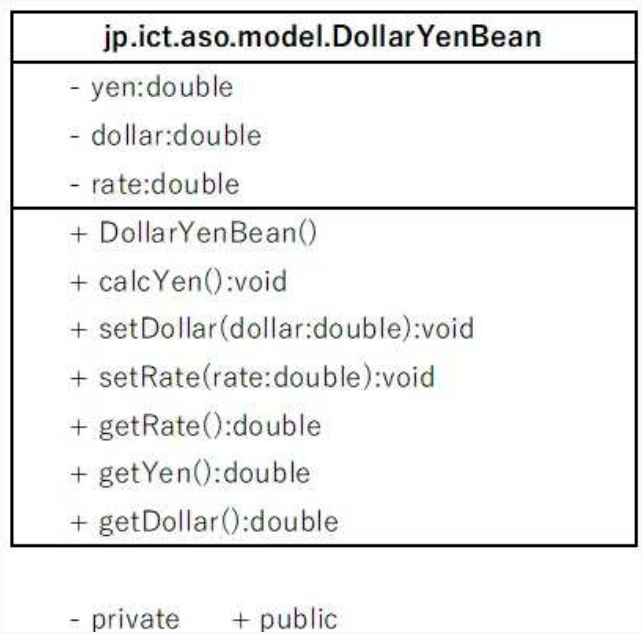

図15. ドル円変換ロジックのクラス図

連絡先

質問や不明な点がある場合は以下のアドレスにメールをください。24時間以内に返信いたします。

info@slowlife.halfmoon.jp