



MVモデルによるWebアプリケーション開発（No12.2 進10進変換器）-EE8



office・M

2024年12月15日 15:14

¥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（2進10進変換器）をコントローラを利用しないMVモデルで実装してみます。

「MVCモデルによるWebアプリケーション開発（No11.2 進10進変換器）-EE8」が終了していることが前提です。

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

開発概要

開発方針

設計仕様

外部設計

内部設計

実装手順

1. 2進10進変換ロジック用JavaBeansクラスを作成します

2. 外部アクセス可能なJSPファイルを作成します

実行確認

課題

ソースコード例

1. BinaryValidator.java

2. DecimalValidator.java

3. binaryDecimal.jsp → binaryDecimalCode.jsp

連絡先

開発概要

「MVCモデルによるWebアプリケーション開発（No11.2進10進変換器）-EE8」で作成したWebアプリをJSPから直接Beanを利用する方式に変更します。

JSPからクラスファイル（Bean）を呼び出す方法には、**拡張タグを使う方法**と**Javaのコードを使う方法**の2種類の方法があります。ここでは、新しくJSPから拡張タグを使って直接JavaBeansのインスタンスを使用する方法を採用します。この時のJavaBeansは以下の条件で実装されていることが必要です。

【JSPから拡張タグを使ってJavaBeansを利用するための条件】

作成するクラスは完全に標準的なJavaBeansの考え方に従って作成されていなければなりません。つまり以下のBeanの約束事が守られていないといけません。

- Serializableインターフェースを**implements**する
- 引数無しの**デフォルトコンストラクタ**を使って**インスタンス化**する
- **Setterメソッド**で値を設定して**業務ロジックのメソッド**を起動する
- 結果は**getterメソッド**で受け取る

※この時privateで宣言された**クラス変数をプロパティと呼び**、set○○○ get○○○の○○○はプロパティと**名前が一致**しなければいけません。
（ただし○○○の最初の一文字は**大文字**になる）』

に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。

以上の条件に従いModel（業務ロジック）－View（表示）に分割して実装することになります

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MV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

【2進10進変換器の仕様】

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)

- ・ラジオボタンで2進数から10進数への変換かその逆かを選択します。
- ・数値フィールドに変換したい数値を入力します。
- ・「変換ボタン」をクリックすると同じ画面内に変換結果を表示します。
- ・変換可能な範囲を外れた数値を入力した場合、数字以外の文字を入力した場合はエラーメッセージを表示します。
- ・入力する10進数は整数値です。

(変換範囲)

2進数10進数の相互変換可能な数値はint型の取りうる範囲とします。Javaのint型は**32ビット**になるので**10進数では-2147483648～2147483647**の値を変換することが可能になります。また、Javaでは負数を表現するために先頭ビットが符号ビットとなるため**2進数は31ビット以内で入力**することになります（2進数で負の数の変換は考慮しません）。

この範囲を超えた入力エラーメッセージを出力するようにします。

【Model（処理ロジック）の設計方針】

以下のModelを実装します。

ConversionBeanクラス：←BinaryDecimalBeanから変更します

- ・ Beanの仕様に従い実装する
- ・ 2進数を10進数に変換するメソッドを実装する
- ・ 10進数を2進数に変換するメソッドを実装する

BinaryValidatorクラス：

- ・ 通常クラスで実装する
- ・ 入力値が2進数として適切であるか評価するメソッドを実装する

DecimalValidatorクラス：

- ・ 通常クラスで実装する
- ・ 入力値が10進数として適切であるか評価するメソッドを実装する

外部設計

接続urlは<http://localhost:8080/converter/binaryDecimal.jsp>とします。よってEclipseの動的Webプロジェクトの名前は**converter**となり直接URLアクセス可能な**binaryDecimal.jsp**を用意します。

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→「converter」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。ラジオボタンで変換のタイプを選択し、変換したい数値を入力して「変換ボタン」をクリックすると、図3のように変換結果が表示されます。この時、新たな画面へ遷移はしません。図4、図5のように入力範囲エラーも対処します。

図2. 起動画面

図3. 変換結果画面

図4. 10進数入力範囲エラー

図5. 2進数入力範囲エラー

2進10進変換を行うクラスの仕様とクラス図は以下のようになります。

(仕様)

属性 (フィールド):

- binary: 変換後の2進数の値を保持する文字列 (String)
- decimal: 変換後の10進数の値を保持する文字列 (String)
- number: 変換したい数値を保持する文字列 (String)
- conversionType: 変換方向 (10進数→2進数or2進数→10進数) を指定する文字列 (String)
- message: 変換の成否を知らせる文字列 (String)

コンストラクタ:

- ConversionBean(): フィールドを初期化するデフォルトコンストラクタ

メソッド:

- conver(): 入力値のチェックを行い変換を行うメソッド
- setNumber(number: String): number 属性を設定する setter メソッド
- setConversionType(conversionType: String): conversionType属性を設定する setter メソッド
- getBinary(): String: binary 属性を取得する getter メソッド
- getDecimal(): String: decimal 属性を取得する getter メソッド
- getMessage(): String: message 属性を取得する getter メソッド

(クラス図)

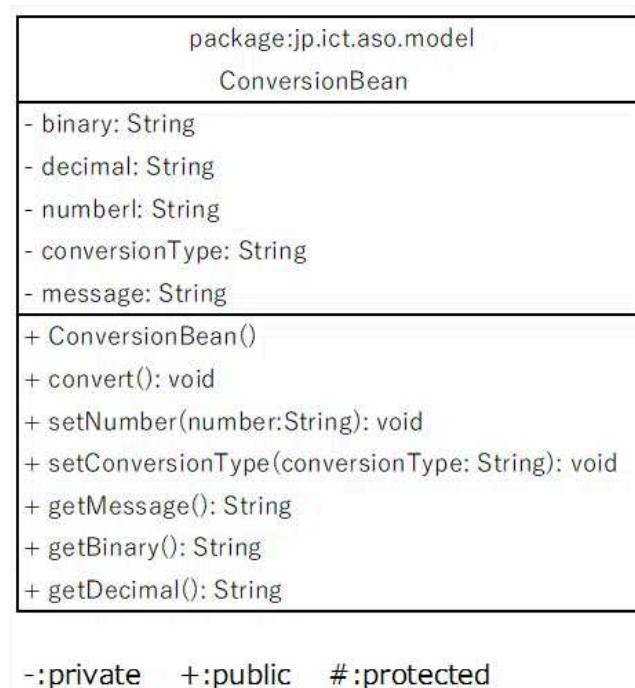


図6. ConversionBeanクラス図

2進数の範囲チェック、数値チェックを行うクラスの仕様とクラス図は以下のようになります。これは「MVCモデルによるWebアプリケーション開発（No11.2進10進変換器）-EE8」や「Swingによるデスクトップアプリケーション開発（2進10進変換器）-JavaSE1.8」で作成したものと同様です。

BinaryValidatorクラス

（仕様）

メソッド:

入力された文字列が**31ビット以内の2進数**かどうかを判定します。

@param binary 入力された2進数を表す文字列

@return true なら有効な31ビット以内の2進数、false なら無効な値

DecimalValidatorクラス

（仕様）

メソッド:

入力された文字列が-2147483648〜2147483647の範囲にある10進数かどうかを判定します。

@param decimal 入力された10進数を表す文字列

@return true なら有効な範囲の10進数、false なら無効な値

実装手順

1. 2進10進変換ロジック用JavaBeansクラスを作成します

- ConversionBean.java

パッケージ : jp.ict.aso.model

名前 : ConversionBean

ソースコード : 入力チェックを含んでいます。以下のようになります。

```
package jp.ict.aso.model;

import java.io.Serializable;

public class ConversionBean implements Serializable{
    private String number;
    private String conversionType;
    private String binary;
    private String decimal;
    private String message;

    public ConversionBean() {
        binary="";
        decimal="";
        message="";
    }

    public void convert() {
        // conversionType に基づいて処理を分ける
        if ("binaryToDecimal".equals(conversionType)) {

            //入力値の範囲チェック
            BinaryValidator bv = new BinaryValidator();
            if(bv.isValidBinary(number)) {                //true範囲内
                // メッセージ(OK)を設定
                message="数値チェックOK";
                // 2進数を10進数に変換
                this.decimal = String.valueOf(Integer.parseInt(number, 2));
            }else {                                        //false範囲外
                // 数値範囲外ならエラーメッセージを設定
                message="入力された値" + number
                    + " は31ビット以内の2進数ではありません。";
            }
        }
    }
}
```

```

    } else if ("decimalToBinary".equals(conversionType)) {

        //入力値の範囲チェック
        DecimalValidator dv = new DecimalValidator();
        if(dv.isValidDecimal(number)) {           //true範囲内
            // メッセージ(OK)をセッションに設定
            message="数値チェックOK";
            //10進数を2進数に変換
            this.binary = Integer.toBinaryString(Integer.parseInt(number));
        }else {                                   //false範囲外
            // 数値範囲外ならエラーメッセージを設定
            message="入力された値" + number
                + " は-2147483648～2147483647の範囲にある10進数ではありません。";
        }
    }
}

public void setNumber(String number) {
    this.number = number;
}

public void setConversionType(String conversionType) {
    this.conversionType = conversionType;
}

public String getBinary() {
    return binary;
}

public String getDecimal() {
    return decimal;
}

public String getMessage() {
    return message;
}

}

```

2. 外部アクセス可能な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
 converter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

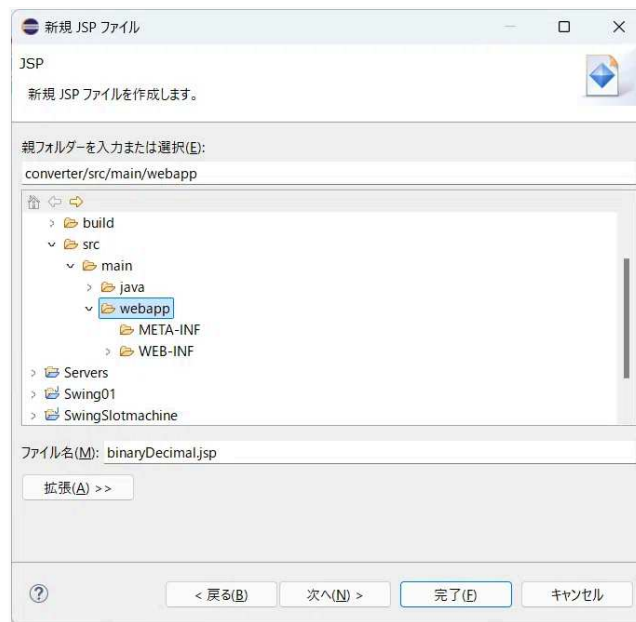


図7. urlから記憶節アクセス可能なJSPファイルの位置

- binaryDecimal.jsp

保存場所 : libcon/src/main/webapp/ ← **注意 !**

ファイル名 : binaryDecimal.jsp

ソースコード : 以下のように実装します

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<jsp:useBean id="bdb" scope="page"
    class="jp.ict.aso.model.ConversionBean" />
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>2進10進変換器</title>
</head>
<body>
    <h1>2進数と10進数の相互変換</h1>
    <form method="POST" action="binaryDecimal.jsp">
        <input type="radio" name="conversionType" value="binaryToDecimal" required>
        2進数を10進数に変換(31bit以内)
        <br>
        <input type="radio" name="conversionType" value="decimalToBinary" required>
        10進数を2進数に変換(-2147483648～2147483647の範囲)
        <br><br>
        変換する数値:
        <input type="number" name="number" required><br><br>
        <button type="submit">送信</button>
    </form><br>

    <jsp:setProperty name="bdb" property="number" param="number" />
    <jsp:setProperty name="bdb" property="conversionType" param="conversionType" />
```

```
<% bdb.convert();%>
<hr>
【結果】<br>
<ul>
<li>メッセージ:<jsp:getProperty name="bdb" property="message" /></li><br>
<li>10進数:<jsp:getProperty name="bdb" property="decimal" /></li>
<li>2進数:<jsp:getProperty name="bdb" property="binary" /></li>
</ul>
</body>
</html>
```

実行確認

JSP (binaryDecimal.jsp) を実行します！！

Eclipseパッケージ・エクスプローラより
binaryDecimal.jspを右クリック→実行→サーバーで実行
→図8のように実行される



図8. 起動画面

課題

今回はJSPの拡張タグを使って、クラスを実体化しメソッドを利用する方法を学びました。一方で、当然のことながら拡張タグ部分はJavaのコードで実現することも可能です。

そこで、binaryDecimal.jspをJavaのコードで処理するよう変更して下さい。
※binaryDecimal.jspをbinaryDecimalCode.jspにコピーして変更します！

binaryDecimalCode.jspのソースコードを提出してください。