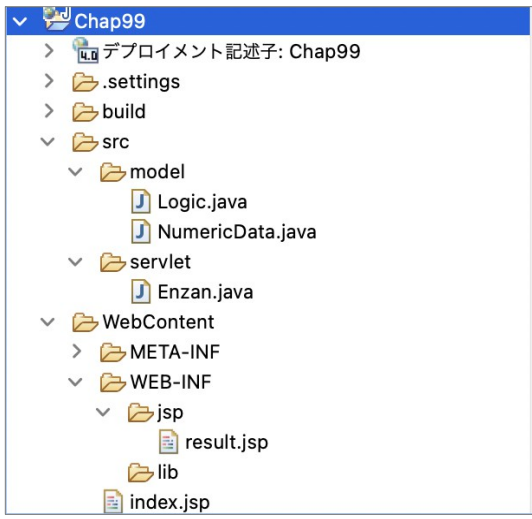


フォルダ構成



```

NumericData.java | Logic.java | index.jsp | *Enzan.java
1 <%@ page language="java" contentType="text/html; charset
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7   <title>四則演算テスト</title>
8 </head>
9 <body>
10  <h1>四則演算</h1>
11  <form action="/Chap99/Enzan" method="post">
12    数値1 : <input type="text" name="num1"><br>
13    数値2 : <input type="text" name="num2"><br>
14    <input type="submit" value="実行">
15  </form>
16 </body>
17 </html>

```

Index.html
ここからスタートという案内postで投げてます。

スコープに入れるためのクラス : NumericData.java

```

NumericData.java
1 package model;
2
3 import java.io.Serializable;
4
5 public class NumericData implements Serializable{
6
7   private int numeric1;
8   private int numeric2;
9   private double numericAdd; //add
10  private double numericDiv; //divide
11  private double numericMul; //multiplie
12  private double numericSub; //subtract
13
14  入力を受けるための整数を2つと結果を
15  入れるためのdoubleを用意。getter と setter
16  はエクリプス任せ。
17  public void setNumeric1(int numeric1) {
18    this.numeric1 = numeric1;

```

計算をするモデル Logic.java

```

NumericData.java | Logic.java | index.jsp | *Enzan.java
1 package model;
2
3 public class Logic {
4
5   public void execute( NumericData num) {
6     double num1 = num.getNumeric1();
7     double num2 = num.getNumeric2();
8
9     double temp = num1 + num2;
10    num.setNumericAdd(temp);
11
12    temp = num1 - num2;
13    num.setNumericSub(temp);
14
15    temp = num1 * num2;
16    num.setNumericMul(temp);
17
18    temp = num1 / num2;
19    num.setNumericDiv(temp);
20
21  }
22 }
23

```

NumericDataのgetterで取り出したデータをローカル変数に取り込み。
num, num1, num2, tempはここでしか使わないので好きな名前でOK
計算結果をNumericDataのsetterを使って numericData に保存

Enzan.java

```

Enzan.java
1
2 import model.Logic;
3 import model.NumericData;
4
5 * Servlet implementation class Enzan
6 @WebServlet("/Enzan")
7 public class Enzan extends HttpServlet {
8   private static final long serialVersionUID = 1L;
9
10  * @see HttpServlet#HttpServlet()
11  public Enzan() {
12
13  * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
14  protected void doPost(HttpServletRequest request, HttpServletResponse response)
15    // リクエストパラメータを取得
16    String num1 = request.getParameter("num1"); //数値1
17    String num2 = request.getParameter("num2"); //数値2
18    // 入力値をプロパティに設定
19    NumericData numericData = new NumericData();
20    numericData.setNumeric1(Integer.parseInt(num1));
21    numericData.setNumeric2(Integer.parseInt(num2));
22    // 健康診断を実行し結果を設定
23    Logic logic = new Logic();
24    logic.execute(numericData);
25    // リクエストスコープに保存
26    request.setAttribute("numericData", numericData);
27    // フォワード
28    RequestDispatcher dispatcher =
29      request.getRequestDispatcher(
30        ("/WEB-INF/jsp/result.jsp");
31    dispatcher.forward(request, response);
32  }
33 }

```

自動で入ると思いますが、名前を変えたときとか要確認。

htmlからのデータは
文字列なので、
整数に変換
使うデータの型によって使い分ける

result.jsp を呼び出す。

result.jsp

```

result.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="model.NumericData"%>
4
5 // リクエストスコープに保存されたHealthを取得
6 NumericData num = (NumericData) request.getAttribute("numericData");
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11   <meta charset="UTF-8">
12   <title>四則演算結果</title>
13 </head>
14 <body>
15   <h1>結果</h1>
16   <p>
17     <%= num.getNumeric1() %> + <%= num.getNumeric2() %> = <%= num.getNumericAdd(
18       num.getNumeric1(), num.getNumeric2()) %>
19     <%= num.getNumeric1() %> - <%= num.getNumeric2() %> = <%= num.getNumericSub(
20       num.getNumeric1(), num.getNumeric2()) %>
21     <%= num.getNumeric1() %> * <%= num.getNumeric2() %> = <%= num.getNumericMul(
22       num.getNumeric1(), num.getNumeric2()) %>
23     <%= num.getNumeric1() %> / <%= num.getNumeric2() %> = <%= num.getNumericDiv(
24       num.getNumeric1(), num.getNumeric2()) %>
25   </p>
26   <a href="/Chap99/">戻る</a>
27 </body>
28 </html>

```

スタートをindex.htmlにしたので
ファイル名を入れなくてもOK。違
う名前からスタートする場合には
ファイル名を忘れずに。