

## ※課題は「mvcKadai3」という

### 名前で新しくプロジェクトを作成して行うこと

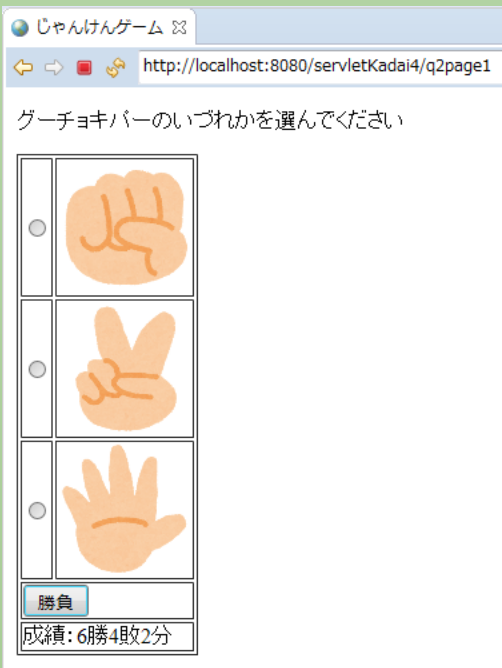
【問題1】以前 servletKadai4 で作成した、じゃんけんゲームを MVC で書き直してみよう！

次のように画面遷移する Web アプリケーションを作りなさい

#### ○1 ページ目

URL: <http://localhost:8080/servletKadai4/q2page1>

画面：

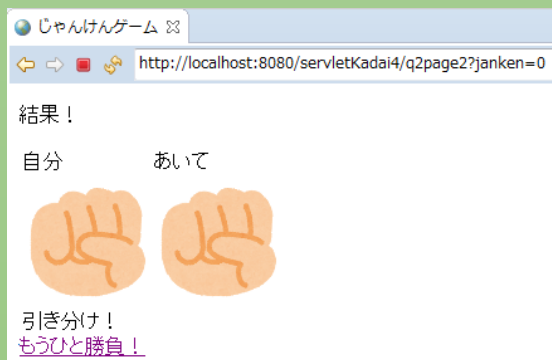


仕様：グーチョキパーのいずれかを選択して、勝負ボタンをクリックすると、2ページ目へ遷移する。成績の初期値は「0 勝 0 敗 0 分」で、勝負ボタンをクリックして勝敗が決まるたびにカウントアップされていく。

#### ○2 ページ目

URL: <http://localhost:8080/servletKadai4/q2page2>

画面：



仕様：自分が選んだ手の画像と、ランダムで選ばれたあいての手の画像を表示して、その下に、結果を表示する。結果のメッセージは以下の3種類

勝った場合：勝ち！

負けた場合：負け！

あいこの場合：引き分け！

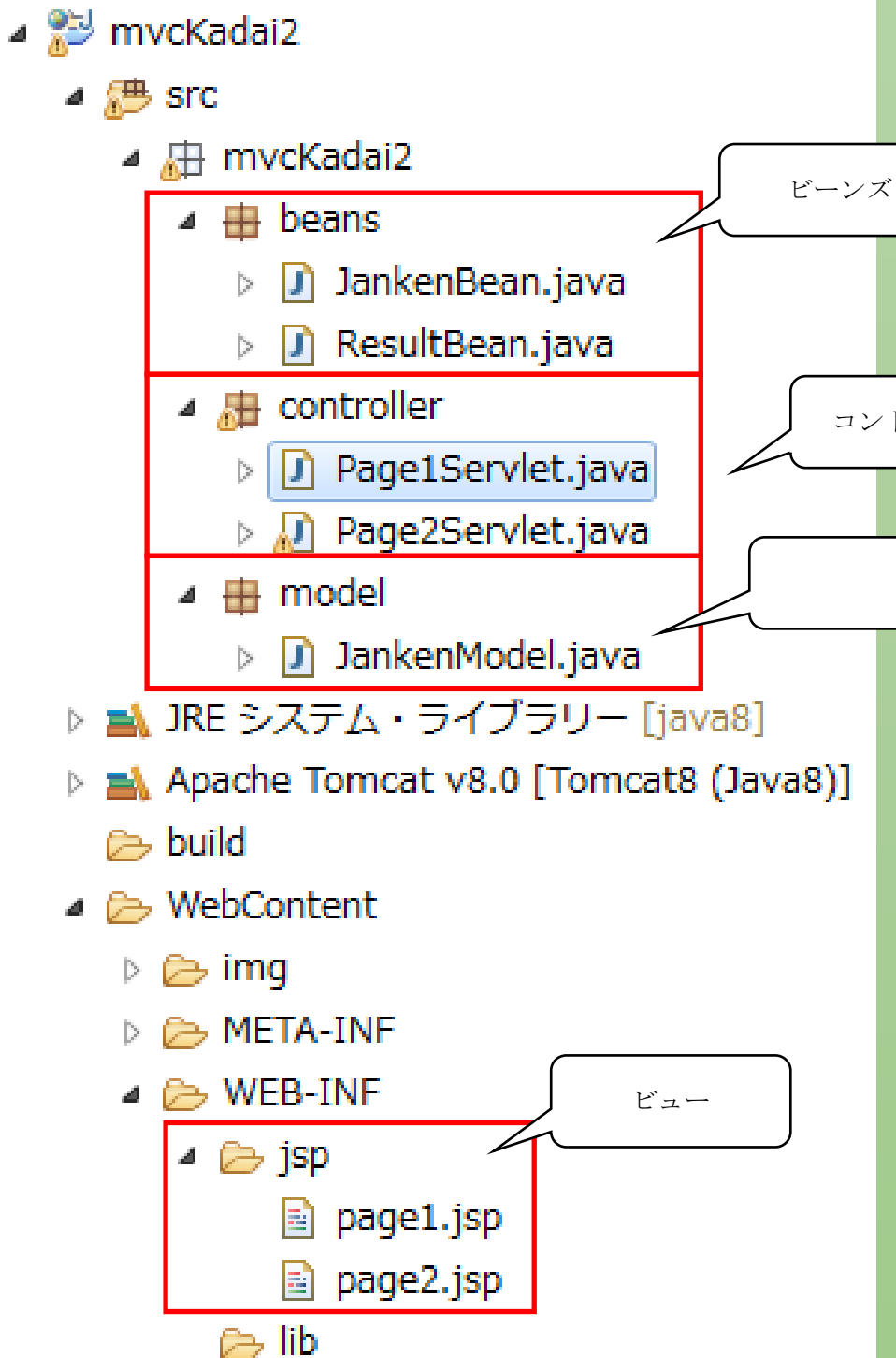
「もうひと勝負！」のリンクをクリックすると1ページ目へ戻る

#### 【ヒント】

順序としては以下のように考えよう

1. どの部分を **Model** に抜き出すかを考える
2. **Model** のインターフェース（引数=インプット、戻り値=アウトプット）を考える
3. **Beans** を使用するかどうかを考える

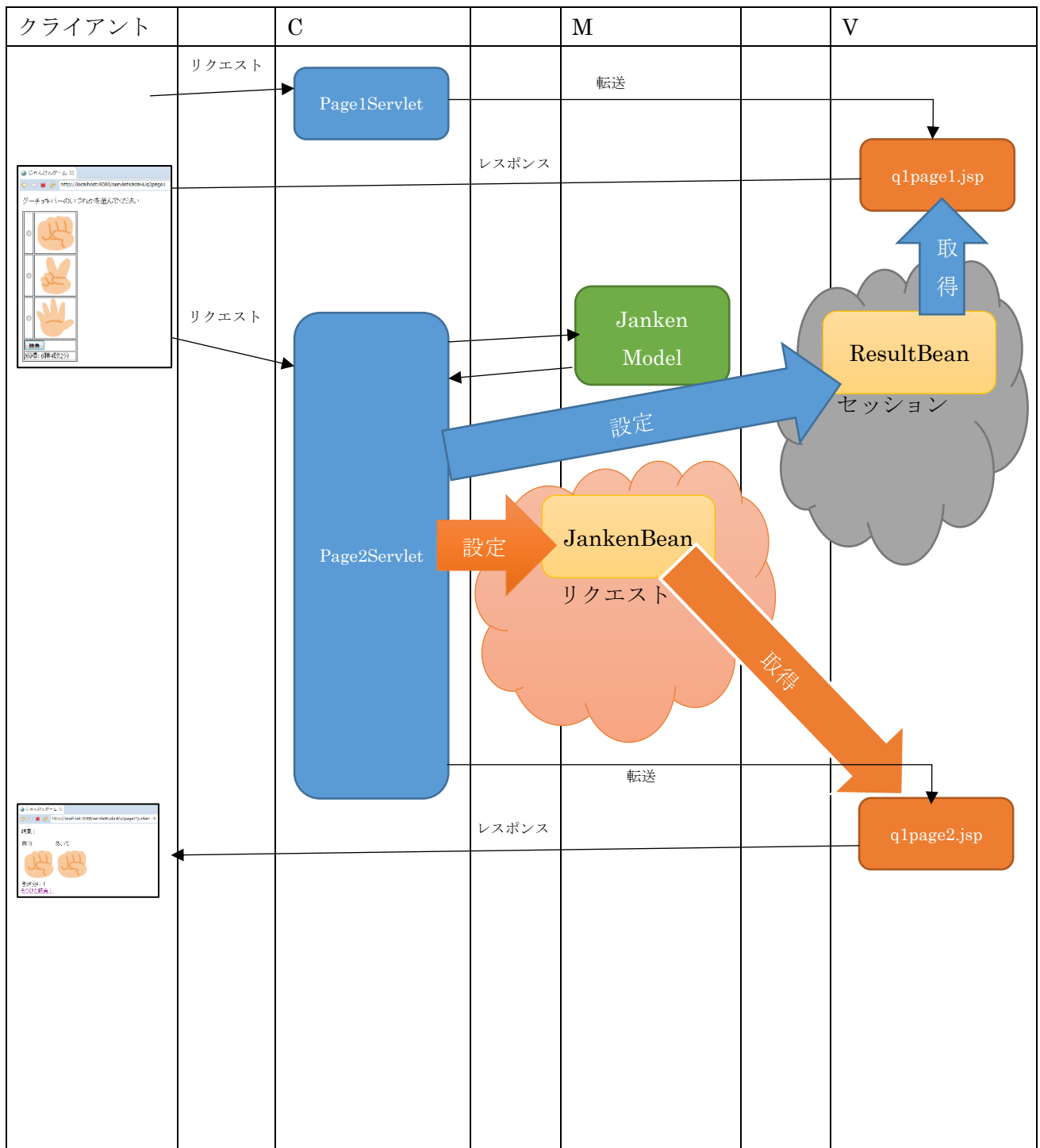
今回のパッケージはビーンズが追加になるので、このようになるはずです



それぞれのクラスでやることは以下のとおり

種類	ファイル名	処理概要
ビーンズ (M)	JankenBean	じゃんけんの情報を保持するクラス。 <ul style="list-style-type: none"><li>• 自分の手の画像（文字列）</li><li>• 相手の手の画像（文字列）</li><li>• 表示するメッセージ（文字列）</li><li>• 相手の手（数値）</li></ul> リクエストに保存して、2 ページ目に表示する
	ResultBean	じゃんけんの結果を保持するクラス。 勝ち数、負け数、引き分け数 を保持する。セッションに保存する
モデル (M)	JankenModel	乱数で相手の手を決定して、勝敗を決定する。 また、自分と相手の手の画像を決定して、JankenBean に格納する
コントローラ (C)	Page1Servlet	1 ページ目を表示する (VIEW (jsp) に転送のみ)
	Page2Servlet	page1.jsp から送られた自分の手を取得する。 JankenModel のインスタンスを生成して、じゃんけん の処理を行い、勝敗を取得しリクエストに JankenBean をセットする。また、JankenBean の勝敗情報よりセッ ションにある、勝ち数、負け数、引き分け数の情報を更 新する
ビュー (V)	q1page1.jsp	1 ページ目の表示 セッションから ResultBean を取得し、通算成績を表示 する
	q1page2.jsp	結果の表示画面 リクエストより JankenBean を取得して、画像とメッ セージを表示する

シーケンスは以下のようになる



※次ページよりヒントがあります！

## 【ヒント】

### ○ResultBean の使いどころ

基本的に **Model** には、**Web** の操作を行いません。

つまり、「セッションに **ResultBean** をセットする」というプログラムは **Controller**（サーブレット）に記載するのが一般的です。

### ○Page2Servlet の実装のヒント

**Page2Servlet** では、以下のような処理が必要です。

1. **page1** で選択された手を取得する
2. じゃんけんの処理を行う
3. リクエストに相手の手の画像やメッセージをセットする
4. セッションに通算成績をセットする

このうち、**Model** に抜き出すのは 2. の部分のみです。

残りの部分は、**Web** の操作が絡むので、**Controller** 側で実装します。

### ○Model のインターフェースの考え方

じゃんけんの処理をするには何が必要かを考えます。

- ・自分の手
- ・相手の手

これがあれば、勝敗が決定できます。このうち、相手の手はランダムで決定するので **Model** 側で決めてあげればよい。つまり **Controller** 側から入力値としては不要です。

しかし、自分の手は 1 ページ目で選択した項目なので、これは **Model** 側で勝手に決めるわけにはいきません。**Page2Servlet** で 1 ページ目の値を取得しているので、この値を **Controller** から貰う必要があります。

→【結論】**Model** のインプット（**Controller** から受け取る値）は「自分の手を表す値」

アウトプットは、じゃんけんの結果となります。これと一緒に画像のファイル名も返すと良いのでアウトプットは **JankenBean** とするのが良さそうです。

→【結論】**Model** のアウトプット（**Controller** へ返す値）は「じゃんけんの結果（と画像）」

※次ページよりさらなるヒントがあります。

ここまで見て、よくわからない人はみてみましょう！！

## Page2Servlet

```
@WebServlet("/page2")  
public class Page2Servlet extends HttpServlet {  
  
    private static final int JANKEN_WIN = 0;  
    private static final int JANKEN_LOSE = 1;  
    private static final int JANKEN_DRAW = 2;  
  
    @Override  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        //////////////////////////////////////  
        //どれを選択したかを取得  
        String janken = request.  
          
        int jankenSelect = Integer.parseInt(janken);  
  
        //////////////////////////////////////  
        //じゃんけんの処理を行う  
        JankenModel model =   
          
        JankenBean jankenBean =   
          
        //////////////////////////////////////  
        //結果をセット  
        request.  
          
        //勝敗の結果をセットする  
        setResultToSession(request, jankenBean.getResult());  
  
        //////////////////////////////////////  
        //転送  
        RequestDispatcher dispatcher =  
            request.getRequestDispatcher("WEB-INF/jsp/page2.jsp");  
        dispatcher.forward(request, response);  
    }  
}
```

リクエストに画像やメッセージをセットする

セッションに通算成績をセットする  
メソッドを呼び出している

```
private void setResultToSession(HttpServletRequest request, int result){  
  
    HttpSession session = request.getSession(true);  
    ResultBean resultBean = (ResultBean)session.getAttribute("resultBean");  
  
    if( resultBean == null ){  
        resultBean = new ResultBean(0,0,0);  
    }  
  
    if(result == JANKEN_WIN){  
        resultBean.incrementWin();  
    }else if(result == JANKEN_LOSE){  
        resultBean.incrementLose();  
    }else{  
        resultBean.incrementDraw();  
    }  
  
      
}
```

セッションに通算成績をセットする

## JankenModel

```
public class JankenModel {  
  
    private static final int JANKEN_GU = 0;  
    private static final int JANKEN_CHOKI = 1;  
    private static final int JANKEN_PA = 2;  
  
    private static final int JANKEN_WIN = 0;  
    private static final int JANKEN_LOSE = 1;  
    private static final int JANKEN_DRAW = 2;  
  
    /**  
     * じゃんけんを実行する  
     * @param jankenSelect  
     * @return  
     */  
    public execute(int jankenSelect){  
  
        //////////////////////////////////////  
        //コンピュータの手をランダムで決定する  
        Random rand = new Random();  
  
        int val = rand.nextInt(3);  
  
        //////////////////////////////////////  
        //結果を取得  
        int result = ;  
  
        //////////////////////////////////////  
        //メッセージを取得  
        String message = ;  
  
        //////////////////////////////////////  
        //ビーンにセットする  
        JankenBean jankenBean = ;  
  
        jankenBean.  
        jankenBean.  
        jankenBean.  
        jankenBean.  
  
        return jankenBean;  
    }  
}
```

アウトプットは何？

相手の手と自分の手の組み合わせで勝敗を決める  
(同じクラス内にある getResult メソッドを呼び出す)

結果によるメッセージを取得する  
(同じクラス内にある getMessage メソッドを呼び出す)

JankenBeans のインスタンスを作成する

JankenBeans の setter を使って  
値をセットする

JankenModel は次ページへ続きます・・・！



```

/**
 * @param val
 * @param jankenSelect
 * @return
 */
private int getGameResult(int val, int jankenSelect){
    if( val == jankenSelect ){
        //引き分け！
        return JANKEN_DRAW;
    }

    int result = JANKEN_LOSE;

    //勝ちのパターン
    if( val == JANKEN_GU &&
        jankenSelect == JANKEN_PA){
        result = JANKEN_WIN;
    }else if( val == JANKEN_CHOKI &&
        jankenSelect == JANKEN_GU){
        result = JANKEN_WIN;
    }else if( val == JANKEN_PA &&
        jankenSelect == JANKEN_CHOKI){
        result = JANKEN_WIN;
    }

    return result;
}

/**
 * @param result
 * @return
 */
private String getMessage(int result){
    String message = "";

    if( result == JANKEN_WIN ){
        message = "勝ち！";
    }else if( result == JANKEN_LOSE ){
        message = "負け！";
    }else{
        message = "引き分け！";
    }

    return message;
}

```

```

/**
 * @param select
 * @return
 */
private String getImgFileName(int select){
    String fileName = "";

    if( select == JANKEN_GU){
        fileName = "janken_gu.png";
    }else if(select == JANKEN_CHOKI ){
        fileName = "janken_choki.png";
    }else{
        fileName = "janken_pa.png";
    }

    return fileName;
}

```