サーブレット課題９

**※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**

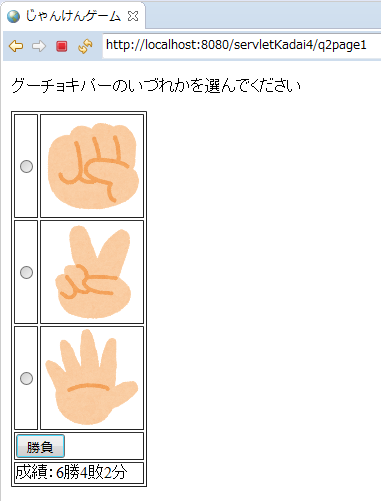
【問題１】以前servletKadai4で作成した、じゃんけんゲームをMVCで書き直してみよう！

　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：

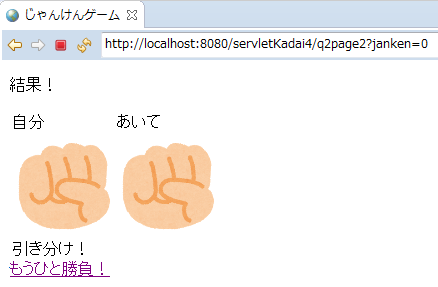


仕様：グーチョキパーのいずれかを選択して、勝負ボタンをクリックすると、２ページ目へ遷移する。成績の初期値は「0勝0敗0分」で、勝負ボタンをクリックして勝敗が決まるたびにカウントアップされていく。

○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】

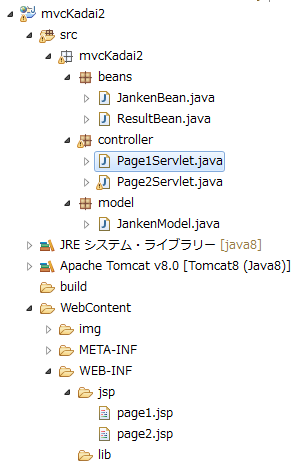
順序としては以下のように考えよう

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、このようになるはずです



ビーンズ

コントローラ

モデル

ビュー

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 | JankenBean | じゃんけんの情報を保持するクラス。  ・自分の手の画像（文字列）  ・相手の手の画像（文字列）  ・表示するメッセージ（文字列）  ・相手の手（数値）  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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| クライアント |  | C |  | M |  | V |
|  | リクエスト  リクエスト | Page2Servlet  Page1Servlet | レスポンス | 転送  リクエスト  JankenBean  Janken  Model  設定  転送 | 設定  q1page2.jsp  レスポンス | 取得  ResultBean  セッション  取得  q1page1.jsp |

※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では、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

　わけには行きません。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

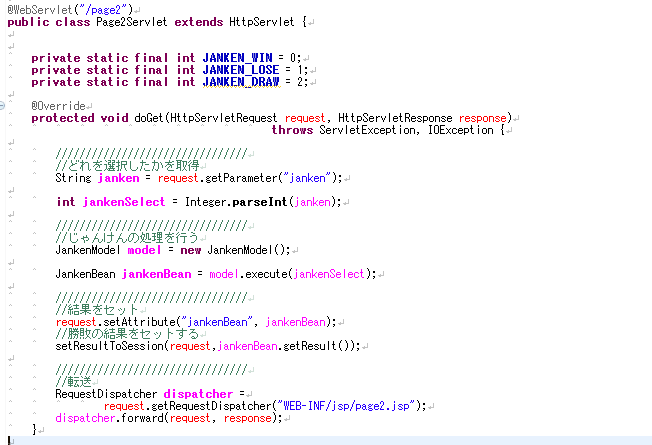
　良いのでアウトプットはJankenBeanとするのが良さそうです。

　→【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、よくわらない人はみてみましょう！！！

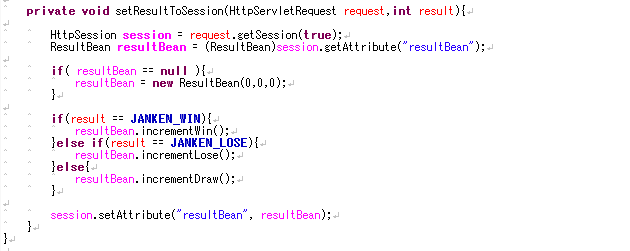
Page2Servlet



リクエストに画像やメッセージをセットする

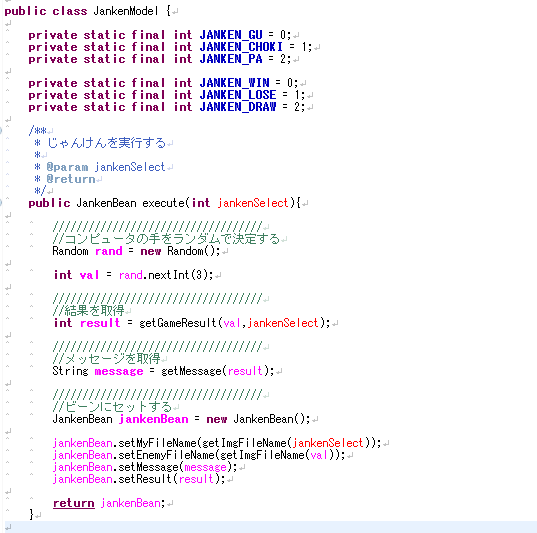
セッションに通算成績をセットする

メソッドを呼び出している



セッションに通算成績をセットする

JankenModel



JankenBeansのsetterを使って

値をセットする

JankenBeansのインスタンスを作成する

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）

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getGameResult

メソッドを呼び出す）

アウトプットは何？

JankenModelは次ページへ続きます・・・！



