



# MVCモデルによるWebアプリケーション開発（No.10スロットマシン）-EE8



office · M

2024年9月9日 10:43

¥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。今回はセッションパラメータで情報を受け渡す方法とgifのアニメーションを作成する方法を学びます。チーム開発での役割分担を想定しています。

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

開発方針

設計仕様

外部設計

内部設計

実装準備

1. 画像とアニメーションを準備します

実装手順

1. 配当計算ロジック用JavaBeansクラスを作成します
2. リクエストコントロール用Servletクラスを作成します
3. 初期状態の画面のJSPファイルを作成します
4. 回転開始状態のJSPファイルを作成します
5. 持ち点確定状態のJSPファイルを作成します

実行確認

ソースコード例と画像例

1. SlotmachineBean.java
2. SlotmachineServlet.java
3. slotmachineView.jsp
4. slotmachineStart.jsp
5. slotmachineResult.jsp

連絡先

## 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。

**題材としてスロットマシンを取り上げます。数字（絵文字）の組み合わせにより配当の点数が増減するゲームをWebアプリケーションで作成します。**

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）－View（表示）－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。

**今回、業務ロジックであるBeanを含むすべての処理と回転アニメーションを実装することにします。**

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 Eclipse2024

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

### 【スロットマシンの仕様】

以下の仕様を実現するWebアプリケーションの作成を行います。

#### （概要）

3つの乱数を発生させて当たりを判定するスロットマシンを作成します。  
数字あるいは絵文字の画像の回転のアニメーションをつけます。

#### （仕様）

##### ①メッセージ表示

- ・ 3つの乱数を発生させそれぞれインスタンス変数に格納する
- ・ 3組揃いでVerryGood、2組揃いでGood、のメッセージを表示する（ばらばらでは表示なし）

##### ②持ち点と配当

- ・ 掛け点、持ち点のインスタンス変数を用意する
- ・ 初期値で持ち点は10点、掛け点は1点とする
- ・ 3組揃いで掛け点の5倍を持ち点にプラスする
- ・ 2組揃いで掛け点の2倍を持ち点にプラスする
- ・ ばらばらで掛け点を持ち点よりマイナスする
- ・ 回転結果、掛け点、持ち点、メッセージ の順にヘッダー部に表示する

# 外部設計

接続urlは<http://localhost:8080/slotmachine/Slotm>とします。よってEclipseの動的Webプロジェクトの名前は**slotmachine**となり、サーブレットのurlパターンは**Slotm**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→「slotmachine」プロジェクトを作成する → 図2の内容で設定する



図2. 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3のような起動画面が表示されます。スタートボタンで数字（絵文字）の回転が始まり、図4のようにストップボタンが押された時点での数字（絵文字）の組み合わせにより配当が決まり持ち点が増減します。



図3. スロットマシン起動画面



図4. スロットマシン配当決定画面

# 内部設計

クラス連携図は図5 のようになります。

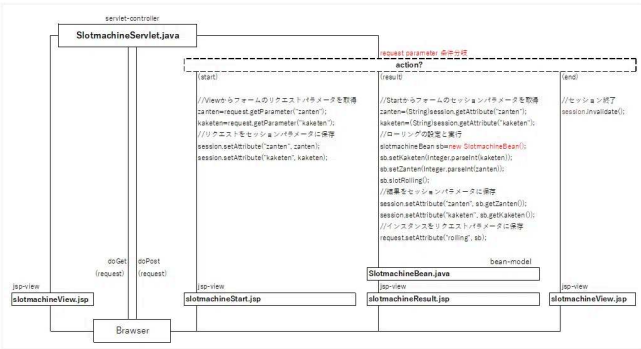


図5. MVCモデル図（クラス連携図）

配当を計算するロジック（SlotmachineBean）のクラス図は図6 のようになります。

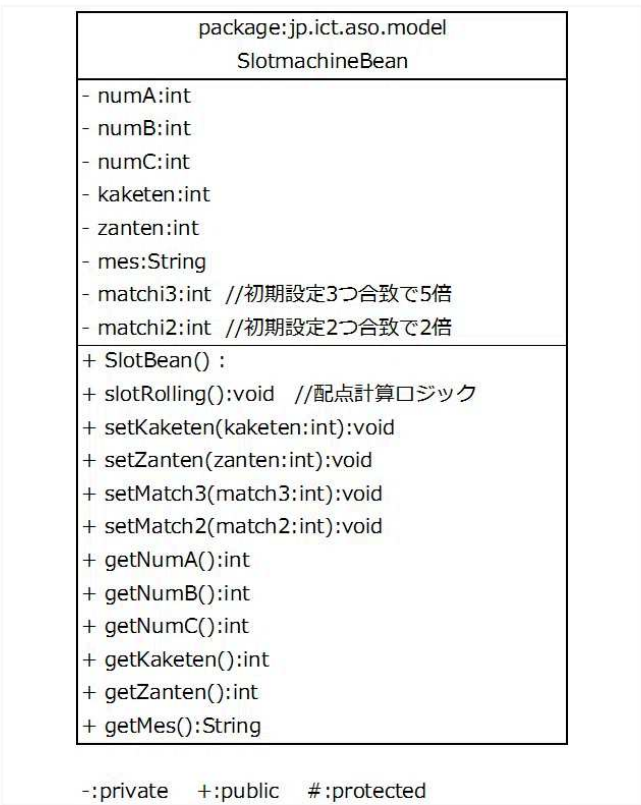


図6. SlotmachineBeanクラス図

# 実装準備

## 1. 画像とアニメーションを準備します

imagesフォルダ内に0～9に対応した画像と回転アニメーションの画像を保存します。

フリー画像サイト → <https://pixabay.com/illustrations/search/>

## アニメーションの作り方 (gimpのインストールが必要です)

→ <https://slowlife.halfmoon.jp/pdf/GIF%E3%82%A2%E3%83%8B%E3%83%A1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%81%AE%E4%BD%9C%E3%82%8A%E6%96%B9.pdf>

今回、特に数字に対応したイラストを以下のWebサービスから使用させていただきました。

<https://loosedrawing.com/>

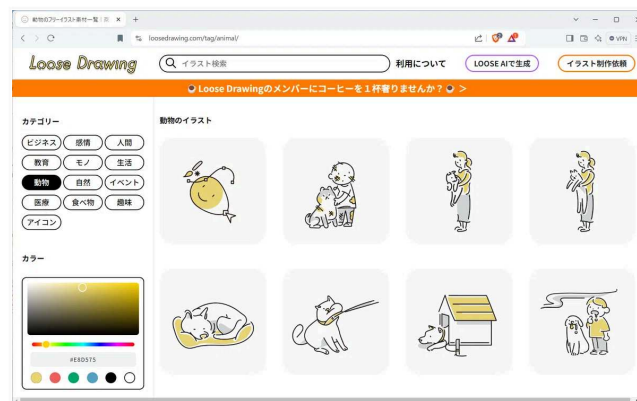


図7. フリーイラストダウンロードサイト

作成した数字回転GIFアニメーションの例を提示します。今回は3桁の数字を10枚重ねたパラパラアニメーションを作成していますが1桁目、2桁目、3桁目を別々に作成した方が良いかもしれません。



図7.1 数字回転GIFアニメーションの例

imagesフォルダはwebapp下に配置します。以下の例は、回転アニメーションを桁ごとに作成しています。



図8. imagesフォルダの配置

## 実装手順

### 1. 配当計算ロジック用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  
slotmachine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

---

- SlotmachineBean.java  
    パッケージ : jp.ict.aso.model  
    名前 : SlotmachineBean  
    ソースコード : 図6を参照して考えましょう！

【ヒント】配点計算ロジックの構成：

```
public void slotRolling(){  
    mes="";  
    numA=new java.util.Random().nextInt(10)%10;  
    numB=new java.util.Random().nextInt(10)%10;  
    numC=new java.util.Random().nextInt(10)%10;
```

```

if((numA==numB)&&(numB==numC)){
    zanten=zanten+kaketen*matchi3;
    mes="VerryGood ";
}else if((numA==numB)|| (numB==numC)|| (numA==numC)){
    zanten=zanten+kaketen*matchi2;
    mes="Good ";
}else{
    zanten=zanten-kaketen;
}
}
}

```

## 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

slotmachine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

---

- SlotmachineServlet.java

パッケージ : jp.ict.aso.controller

クラス名 : SlotmachineServlet

ソースコード : 考えましょう！ ※アノテーションは/Slotm

【ヒント】ソースコードの構成：

doGet

//slotmachineView.jsp にフォワード

doPost

//session変数の宣言

HttpSession session = request.getSession();

//ローカル変数の宣言

□□□□□□□□□□□□□□□□ ← 考えましょう

//リクエストパラメータからactionの値を取得

String action = request.getParameter("action")

//actionパラメータの条件判断

if(action==null) {

//slotmachineView.jsp にフォワード

}else if(action.equals("start")) {

//slotmachineViewから送信されたリクエストパラメータを取得

zanten=request.getParameter("zanten")

kaketen=request.getParameter("kaketen")

//セッションパラメータに保存

session.setAttribute("zanten", zanten)

session.setAttribute("kaketen", kaketen)

//slotmachineStart.jsp にフォワード

}else if(action.equals("result")) {

```

//slotmachineStartから送信されたセッションパラメータを取得
zanten=(String)session.getAttribute("zanten")
kaketen=(String)session.getAttribute("kaketen")
//配当計算ロジックの実体化とローリング設定実行
SlotmachineBean sb=new SlotmachineBean()
sb.setKaketen(Integer.parseInt(kaketen))
sb.setZanten(Integer.parseInt(zanten))
sb.slotRolling()
//配点反映結果をセッションパラメータに保存
session.setAttribute("zanten", sb.getZanten())
session.setAttribute("kaketen", sb.getKaketen())
//インスタンスをリクエストパラメータに保存
request.setAttribute("rolling", sb)
//slotmachineResult.jsp にフォワード
}else if(action.equals("end")) {
    //セッション破棄
    session.invalidate();
    //slotmachineView.jspにフォワード
}
}

```

### 3. 初期状態の画面の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  
slotmachine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

---

・ slotmachineView.jsp

保存場所： slotmachine/src/main/webapp/**WEB-INF/jsp** ← 注意！

ファイル名： slotmachineView.jsp

ソースコード：考えましょう！

【ヒント】

```

<body background="images/back.png" >
<h1>スロットマシン</h1>
<form action="Slotm?action=start" method="post">
<input type="submit" value="スタート "><font color=black size=+2>
    掛け点:<input type="number" name="kaketen" value="1">
    残点:<input type="number" name="zanten" value="10" >
    状態:</font>
</form>
<hr>
<p><table border=0><tr>
<td><img src=images/7.png width=200></td>

```

```
<td><img src=images/7.png width=200></td>
<td><img src=images/7.png width=200></td>
</tr></table>
</p>
</body>
```

## 4. 回転開始状態の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

slotmachine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

---

- slotmachineStart.jsp

保存場所：slotmachine/src/main/webapp/**WEB-INF/jsp** ← 注意！

ファイル名：slotmachineStart.jsp

ソースコード：考えましょう！

【ヒント】

```
<body background="images/back.png" >
<h1>スロットマシン</h1>
<form action="Slotm?action=result" method="post">
<input type="submit" value="ストップ" ><font color=black size=+2>
  掛け点：<%=session.getAttribute("kaketen") %>
  残点：<%=session.getAttribute("zanten") %>
  状態：</font><br>
</form>
<hr>
<p><table border=0><tr>
<td><img src=images/number1.gif width=200></td>
<td><img src=images/number2.gif width=200></td>
<td><img src=images/number3.gif width=200></td>
</tr></table>
</p>
</body>
```

## 5. 持ち点確定状態の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

slotmachine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

- slotmachineResult.jsp

保存場所：slotmachine/src/main/webapp/**WEB-INF/jsp** ← 注意！

ファイル名：slotmachineResult.jsp

ソースコード：考えましょう！**リクエストスコープで転送されたbeanのインスタンスを取得するのを忘れないように！**

【ヒント】

```
<body background="images/back.png" >
<h1>スロットマシン</h1>
<form action="Slotm?action=start" method="post">
<input type="hidden" name="kaketen" value='<%=session.getAttribute("kaketen") %>' >
<input type="hidden" name="zanten" value='<%=session.getAttribute("zanten") %>' >
<input type="submit" value="再スタート"><font color=black size=+2>
  掛け点：<%=sb.getKaketen() %>
  残点：<%=sb.getZanten() %>
  状態：<%=sb.getMes() %></font>
</form>
<hr>
<p><table border=0><tr>
<td><img src=images/<%=sb.getNumA() %>.png width=200></td>
<td><img src=images/<%=sb.getNumB() %>.png width=200></td>
<td><img src=images/<%=sb.getNumC() %>.png width=200></td>
</tr></table>
</p>
<form action="Slotm?action=end" method="post">
<input type="submit" value="終了">
</form>
</body>
```

## 実行確認

サーブレットクラス（SlotmachineServlet.java）を実行します。初期状態の画像、回転状態の画像、持ち点の増減を確認します。



図8. スロットマシン実行画面

## ソースコード例と画像例

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。

また、プログラム中で使用するイメージ画像を次のセクションに置きますので、ダウンロードして使用してください。

----- このラインより上のエリアが無料で表示されます。 -----

### 1. SlotmachineBean.java

```
1 package jp.ict.aso.model;
2 import java.io.Serializable;
3 public class SlotmachineBean implements Serializable{
4     private int numA,numB,numC;
5     private int kaketen,zanten;
6     private String mes;
7
8     private int matchi3;
9     private int matchi2;
10
11     public SlotmachineBean(){
12         matchi3=5; //3つ合致で5倍
13         matchi2=2; //2つ合致で2倍
14     }
15     public void slotRolling(){
16         mes="";
17         numA=new java.util.Random().nextInt(10)%10;
18         numB=new java.util.Random().nextInt(10)%10;
19         numC=new java.util.Random().nextInt(10)%10;
20         if((numA==numB)&&(numB==numC)){
21             zanten=zanten+kaketen*matchi3;
22             mes="VerryGood ";
23         }else if((numA==numB)||(numB==numC)||(numA==numC)){
24             zanten=zanten+kaketen*matchi2;
25             mes="Good ";
26         }else{
27             zanten=zanten-kaketen;
28         }
29     }
30 }
```

図9. SlotmachineBean.java(1)

```

30 public int getNumA() {
31     return numA;
32 }
33 public int getNumB() {
34     return numB;
35 }
36 public int getNumC() {
37     return numC;
38 }
39 public int getKaketen() {
40     return kaketen;
41 }
42 public int getZanten() {
43     return zanten;
44 }
45 public String getMes() {
46     return mes;
47 }
48 public void setKaketen(int kaketen) {
49     this.kaketen = kaketen;
50 }
51 public void setZanten(int zanten) {
52     this.zanten = zanten;
53 }
54 public void setMatch3(int matchi3) {
55     this.matchi3 = matchi3;
56 }
57 public void setMatch2(int matchi2) {
58     this.matchi2 = matchi2;
59 }
60 }

```

図10. SlotmachineBean.java(2)

## 2. SlotmachineServlet.java

```

1 package jp.ict.aso.controller;
2
3 import java.io.IOException;
4
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12
13 import jp.ict.aso.model.SlotmachineBean;
14
15 @WebServlet("/Slotm")
16 public class SlotmachineServlet extends HttpServlet {
17     protected void doGet(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19
20         //フォワード
21         RequestDispatcher rd=request.getRequestDispatcher(
22             ("/WEB-INF/jsp/slotmachineView.jsp");
23         rd.forward(request, response);
24     }
25 }

```

図11. SlotmachineServlet.java(1)

```

26 protected void doPost(HttpServletRequest request, HttpServletResponse response)
27     throws ServletException, IOException {
28
29     HttpSession session = request.getSession();
30     String zanten;
31     String kaketen;
32
33     //リクエストパラメータからactionの値を取得して条件判断
34     String action = request.getParameter("action");
35     if(action==null) {
36         //フォワード
37         RequestDispatcher rd=request.getRequestDispatcher(
38             ("/WEB-INF/jsp/slotmachineView.jsp");
39         rd.forward(request, response);
40     }
41     else if(action.equals("start")) {
42         //slotmachineViewからフォームのリクエストパラメータを取得
43         zanten=request.getParameter("zanten");
44         kaketen=request.getParameter("kaketen");
45         //セッションパラメータに保存
46         session.setAttribute("zanten", zanten);
47         session.setAttribute("kaketen", kaketen);
48         //フォワード
49         RequestDispatcher rd=request.getRequestDispatcher(
50             ("/WEB-INF/jsp/slotmachineStart.jsp");
51         rd.forward(request, response);
52     }
53     else if(action.equals("result")) {
54         //slotmachineStartからフォームのセッションパラメータを取得
55         zanten=(String)session.getAttribute("zanten");
56         kaketen=(String)session.getAttribute("kaketen");
57         //ローリングの設定と実行
58         SlotmachineBean sb=new SlotmachineBean();
59         sb.setKaketen(Integer.parseInt(kaketen));
60         sb.setZanten(Integer.parseInt(zanten));
61         sb.slotRolling();
62     }

```

図12. SlotmachineServlet.java(2)

```

63 //結果をセッションパラメータに保存
64 session.setAttribute("zanten", sb.getZanten());
65 session.setAttribute("kaketen", sb.getKaketen());
66 //インスタンスをリクエストパラメータに保存
67 request.setAttribute("rolling", sb);
68
69 //フォワード
70 RequestDispatcher rd=request.getRequestDispatcher
71 ("/WEB-INF/jsp/slotmachineResult.jsp");
72 rd.forward(request, response);
73
74 }else if(action.equals("end")) {
75 session.invalidate();
76 //フォワード
77 RequestDispatcher rd=request.getRequestDispatcher
78 ("/WEB-INF/jsp/slotmachineView.jsp");
79 rd.forward(request, response);
80
81 }
82 }
83 }

```

図13. SlotmachineServlet.java(3)

### 3. slotmachineView.jsp

```

1 <%@ page language="java" contentType="text/html;
2 charset=UTF-8" pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>スロットマシン</title>
8 </head>
9 <body background="images/back.png" >
10
11 <h1>スロットマシン</h1>
12 <form action="/slot?action=start" method="post">
13 <input type="submit" value="スタート" ><font color=black size=+2>
14 掛け点: <input type="number" name="kaketen" value="1">
15 残点: <input type="number" name="zanten" value="10">
16 状態: </font>
17 </form>
18 <hr>
19 <p><table border=0><tr>
20 <td><img src=images/7.png width=200></td>
21 <td><img src=images/7.png width=200></td>
22 <td><img src=images/7.png width=200></td>
23 </tr></table>
24 </p>
25 </body>
26 </html>

```

図14. slotmachineView.jsp

### 4. slotmachineStart.jsp

```

1 <%@ page language="java" contentType="text/html;
2 charset=UTF-8" pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>スロットマシン</title>
8 </head>
9 <body background="images/back.png" >
10
11 <h1>スロットマシン</h1>
12 <form action="/slot?action=result" method="post">
13 <input type="submit" value="ストップ" ><font color=black size=+2>
14 掛け点: <%=session.getAttribute("kaketen") %>
15 残点: <%=session.getAttribute("zanten") %>
16 状態: </font><br>
17 </form>
18 <hr>
19
20 <p>
21 <img src=images/number1.gif width=200>
22 <img src=images/number2.gif width=200>
23 <img src=images/number3.gif width=200><br>
24 </p>
25
26 </body>
27 </html>

```

図15. slotmachineStart.jsp

### 5. slotmachineResult.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2 pageEncoding="UTF-8" import="jp.ict.aso.model.*"%>
3 <%
4 SlotmachineBean sb=(SlotmachineBean) request.getAttribute("rolling");
5 %>
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="UTF-8">
10 <title>スロットマシン</title>
11 </head>
12 <body background="images/back.png" >
13
14 <h1>スロットマシン</h1>
15
16 <form action="Slotm?action=start" method="post">
17 <input type="hidden" name="kaketen" value="<%=session.getAttribute("kaketen") %>" >
18 <input type="hidden" name="zanten" value="<%=session.getAttribute("zanten") %>" >
19 <input type="submit" value="再スタート"><font color=black size=2>
20 掛け点 : <%=sb.getKaketen() %>
21 残点 : <%=sb.getZanten() %>
22 状態 : <%=sb.getMes() %></font>
23 </form>
24 <br>
25
26 <p><table border=0><tr>
27 <td><img src=images/<%=sb.getNumA() %>.png width=200</td>
28 <td><img src=images/<%=sb.getNumB() %>.png width=200</td>
29 <td><img src=images/<%=sb.getNumC() %>.png width=200</td>
30 </tr></table>
31 </p>
32 <form action="Slotm?action=end" method="post">
33 <input type="submit" value="終了">
34 </form>
35 </body>
36 </html>

```

図16. slotmachineResult.jsp

## 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](mailto:info@slowlife.halfmoon.jp)