

## ※課題は「servletKadai2」という 名前で新しくプロジェクトを作成して行うこと

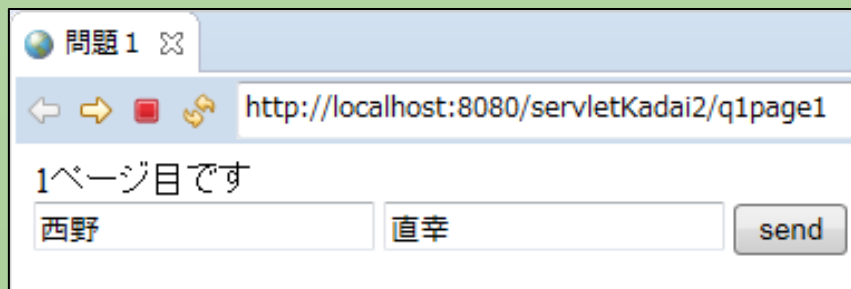
### 【問題 1】

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/servletKadai2/q1page1`

画面：

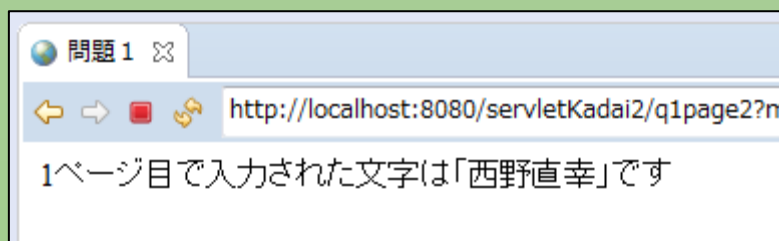


仕様：入力エリア二つについて、初期状態は空っぽ。send ボタンをクリックすると  
2 ページ目へ画面遷移する

○2 ページ目

URL : `http://localhost:8080/servletKadai2/q1page2`

画面：



仕様：1 ページ目の入力された二つの文字列を結合して表示する。

作成は、サーブレットと JSP を使うこと（サーブレット→JSP の転送を使うこと）  
サーブレットのクラス名は `Q1Page1Servlet`、`Q1Page2Servlet`、JSP ファイル名は  
`q1page1.jsp`、`q1page2.jsp` とする

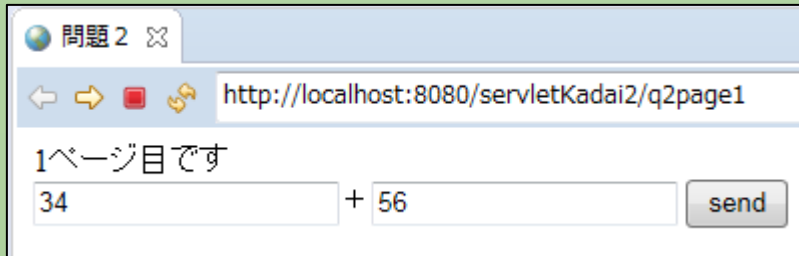
### 【問題 2】

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/servletKadai2/q2page1>

画面：

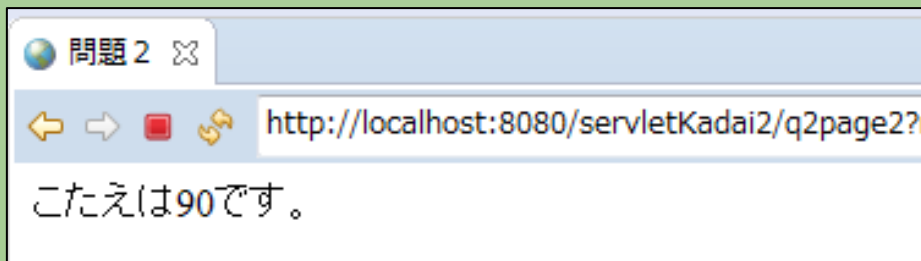


仕様：入力エリア二つについて、初期状態は空っぽ。send ボタンをクリックすると  
2 ページ目へ画面遷移する

○2 ページ目

URL : <http://localhost:8080/servletKadai2/q2page2>

画面：



仕様：1 ページ目の入力された二つの数値を加算して表示する。

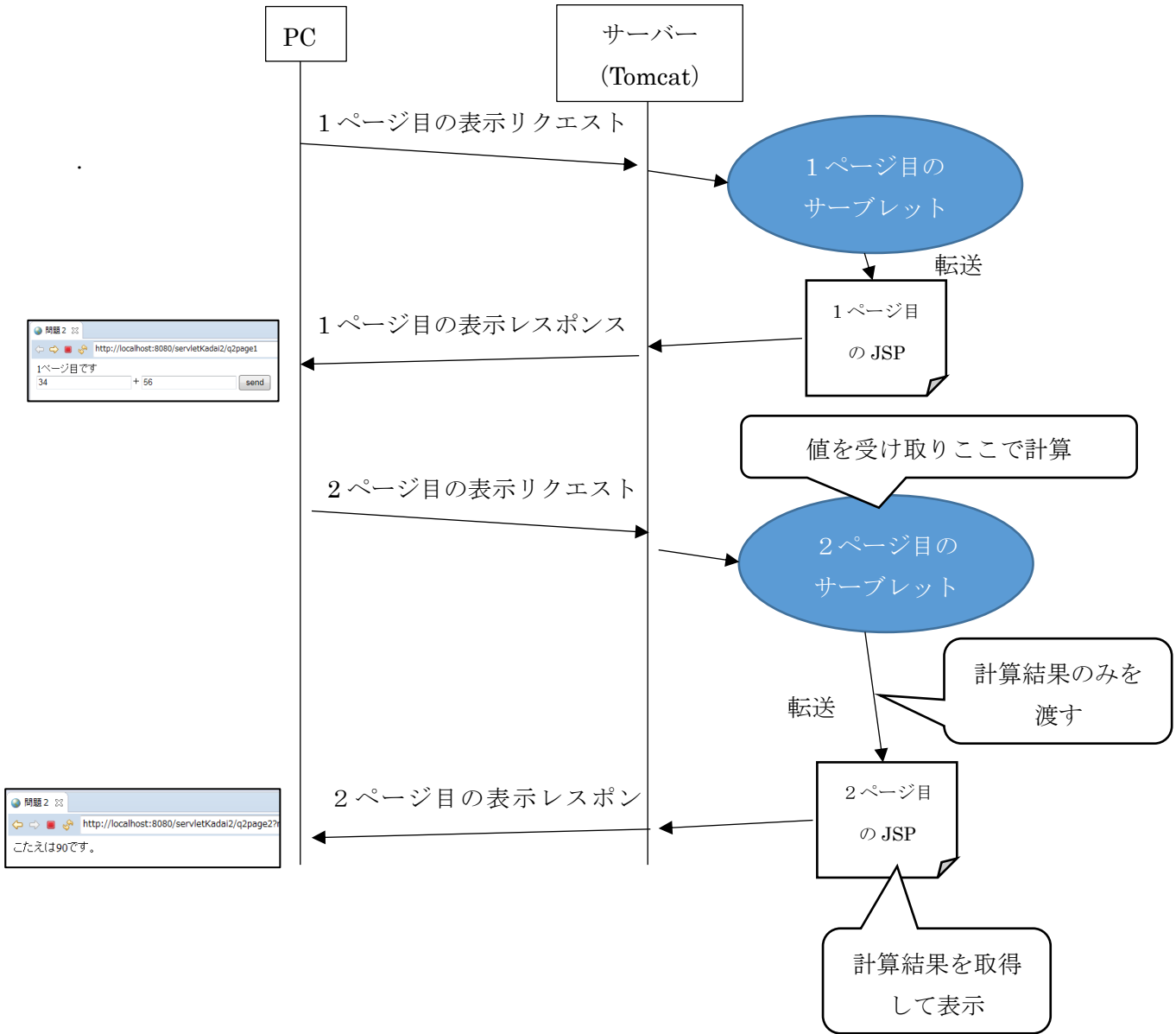
作成は、サーブレットと JSP を使うこと（サーブレット→JSP の転送を使うこと）

ただし、2 ページ目の JSP の body 部は以下の通りとする

```
9 <body>↵
10 <%↵
11 Integer num = (Integer)request.getAttribute("num");↵
12 %>↵
13 こたえは<%=num %>です。↵
14 </body>↵
```

ヒント：サーブレット側で、数値の変換と足し算を行い、JSP には結果だけを送信する

問題 2 のヒント（サーブレットと JSP の連動イメージ）シーケンス図



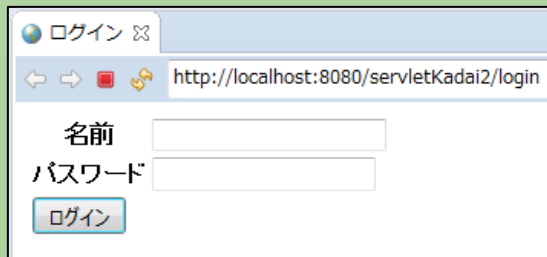
### 【問題3】ログインもどき2を作る

以前 JSP で作ったログインもどきのものをサーブレット&JSP で作ってみよう！

○1 ページ目

URL : <http://localhost:8080/servletKadai2/login>

画面：



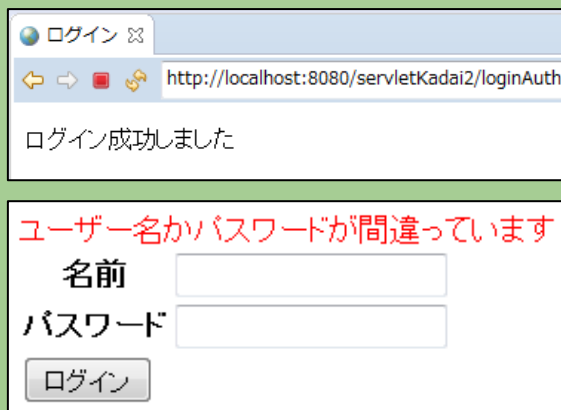
The screenshot shows a web browser window with the title "ログイン". The address bar displays "http://localhost:8080/servletKadai2/login". The page content includes two input fields: "名前" (Name) and "パスワード" (Password). Below these fields is a button labeled "ログイン" (Login).

仕様：名前とパスワードの初期状態は空っぽ。send ボタンをクリックすると  
2 ページ目へ画面遷移する。なお 2 ページ目の通信方法は POST 通信とする。

○2 ページ目

URL : <http://localhost:8080/servletKadai2/loginAuth>

画面：



The top screenshot shows a web browser window with the title "ログイン". The address bar displays "http://localhost:8080/servletKadai2/loginAuth". The page content shows the message "ログイン成功しました" (Login successful).

The bottom screenshot shows the same web browser window, but with a red error message "ユーザー名かパスワードが間違っています" (Username or password is incorrect) displayed above the login form. The form includes input fields for "名前" (Name) and "パスワード" (Password), and a "ログイン" (Login) button.

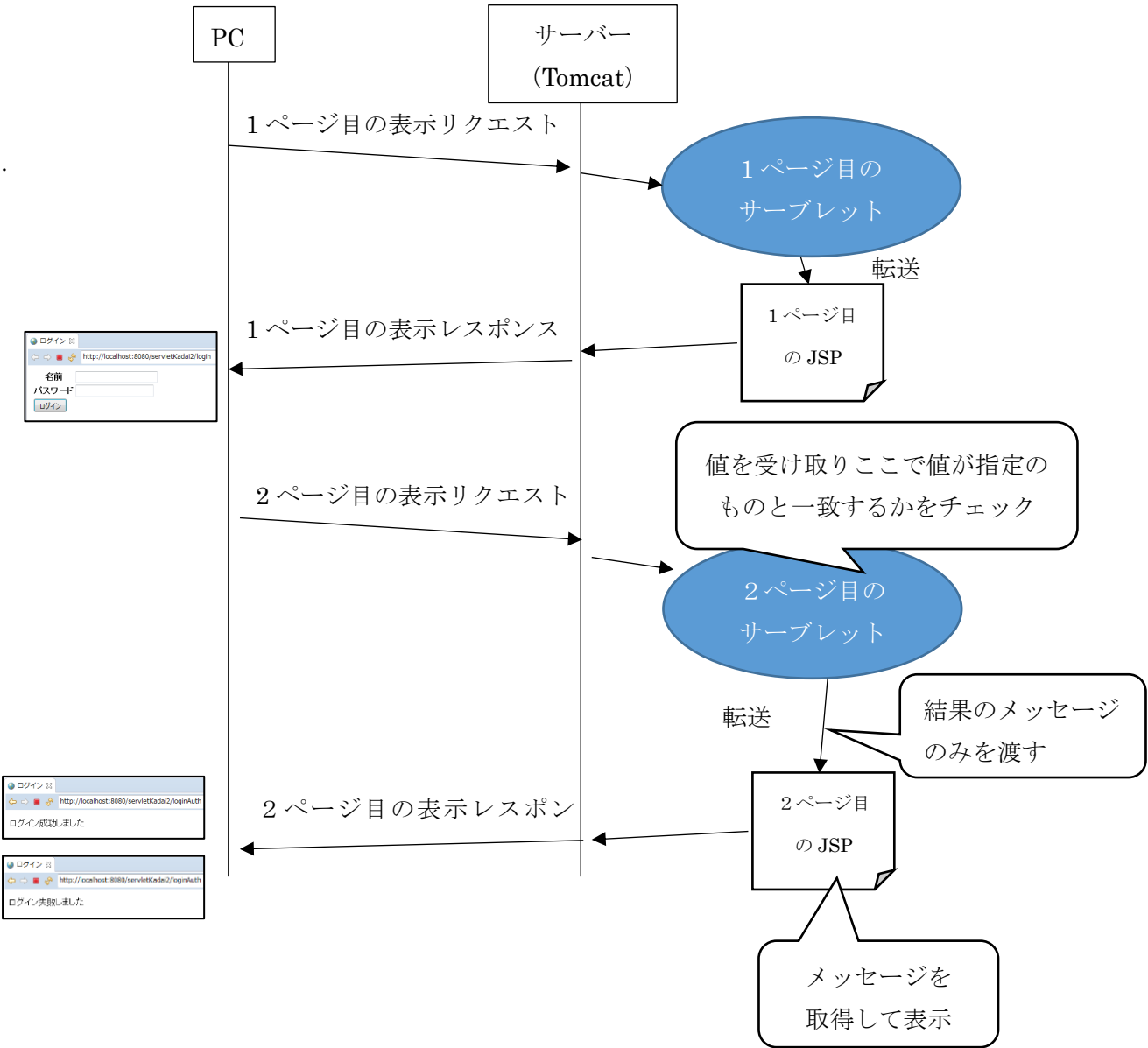
仕様：1 ページ目に入力された値が

名前=nishino、パスワード=1111 の場合は画面に「ログイン成功しました」  
を表示する。それ以外の場合はログイン画面を再表示して、「ユーザー名か  
パスワードが間違っています」を赤文字で表示する。

作成は、サーブレットと JSP を使うこと（サーブレット→JSP の転送を使うこと）

ヒント：POST 通信だから、サーブレット側は doGet ではなくて・・・？

問題 3 のヒント（サーブレットと JSP の連動イメージ）シーケンス図



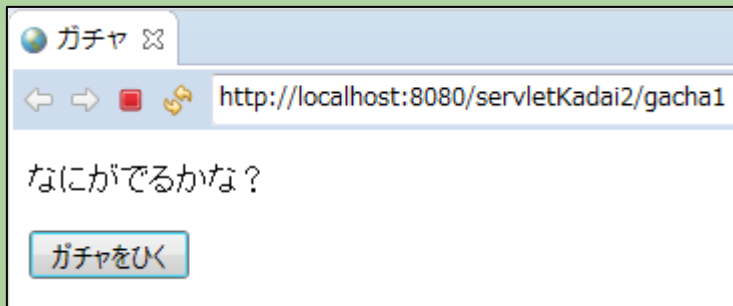
#### 【問題4】ガチャシステムを作ろう

次のように画面遷移する Web アプリケーションを作りたい

○1 ページ目

URL: <http://localhost:8080/servletKadai2/gacha1>

画面：



仕様：「ガチャをひく」ボタンをクリックすると 2 ページ目へ遷移する

○2 ページ目

URL : <http://localhost:8080/servletKadai2/gacha2>

画面：



仕様：乱数を発生させ、以下の確率で、画像とメッセージを表示する

確率	メッセージ	画像
1%	スーパーレア	

10%	レア	
25%	ちょいレア	
64%	普通	

※画像は変更しても良い

上記画像は **GitHub** の課題フォルダにおいてあります。

ヒント：乱数の取得は以下のようにすると 0～99 の値が取得できます。

```
//乱数を取得
```

```
Random rand = new Random();
```

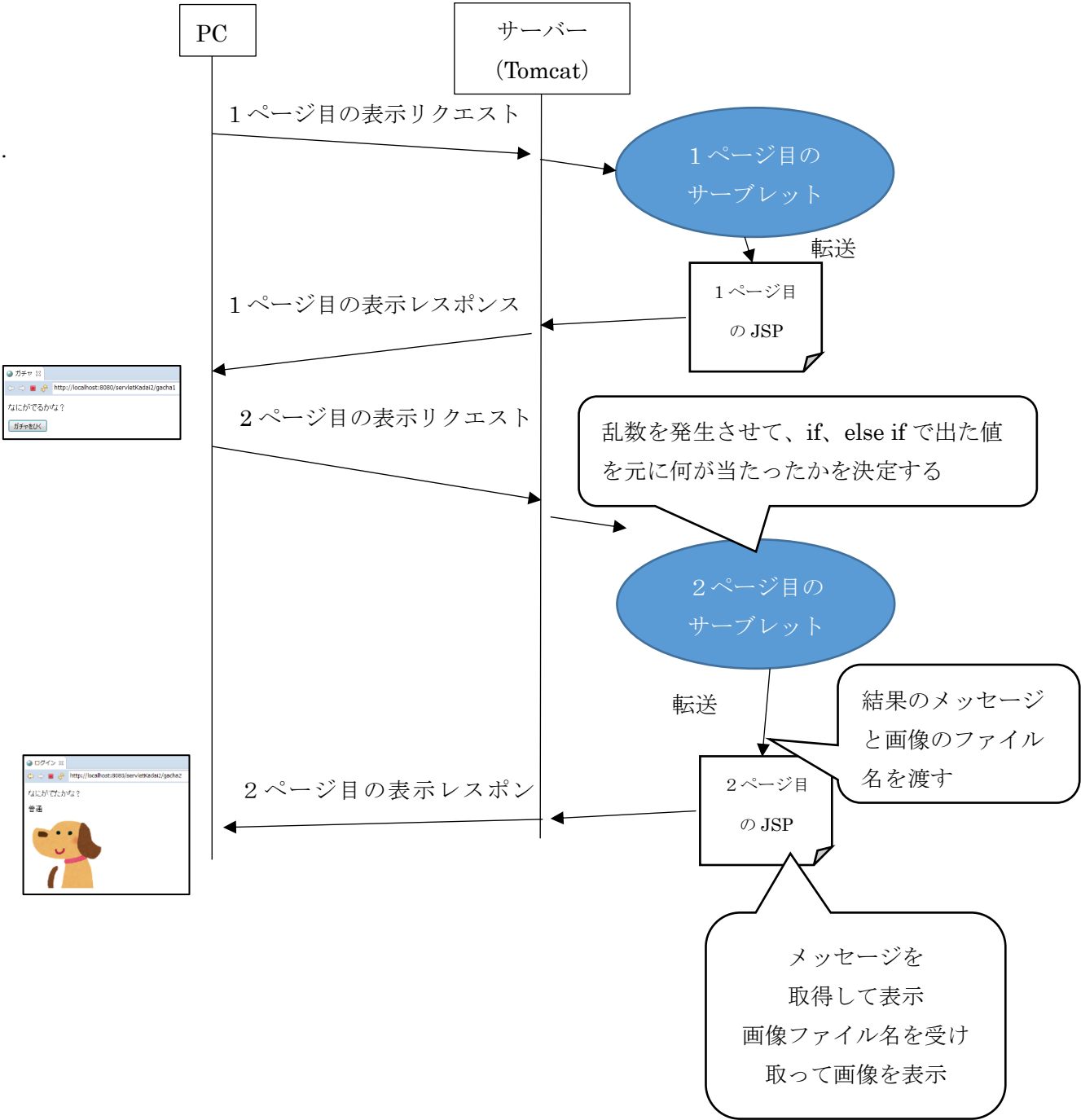
```
int val = rand.nextInt(100);
```

乱数の結果によって、表示する画像とメッセージを決定する。

JSP に転送するのは、「メッセージ」と「画像のファイル名」の2つ。

JSP では、メッセージはそのまま使用し、画像のファイル名は **img** タグで使用する。

問題 4 のヒント（サーブレットと JSP の連動イメージ）シーケンス図





次ページにさらなるヒント！どうしてもわからない人だけ見よう！！！！

本当にみちやって良いのかい？

## 2 ページ目のサーブレット側のソースコード

```
@WebServlet("/gacha2")  
public class Q4Page2Servlet extends HttpServlet {  
    @Override  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        String filename = "";  
        String message = "";  
        //乱数を取得  
        Random rand = new Random();  
        int val = rand.nextInt(100);  
        //値によって表示を変える  
        if (val == 0) {  
            //スーパーレアのメッセージと画像のパス、両方を変数にセット  
              
              
        } else if (val <= 10) {  
            //レアのメッセージと画像のパス、両方を変数にセット  
              
              
        } else if (val <= 35) {  
            //ちよいレアのメッセージと画像のパス、両方を変数にセット  
              
              
        } else {  
            //普通のメッセージと画像のパス、両方変数にをセット  
              
              
        }  
        //メッセージを結果画面へ送る  
         "filename", filename);  
         "message", message);  
        //画面遷移をする  
          
          
    }  
}
```