Javaバイブルシリーズ Webアプリケーション構築の教科書

別冊

練習問題解答
総合演習サンプルプログラム

株式会社SCC

第2章　練習問題の模範解答

練習1　九九の表

プロジェクト名　practice2_1

ファイル　Kuku.java

```
01:  package pac1;
02:
03:  import java.io.IOException;
04:  import java.io.PrintWriter;
05:
06:  import javax.servlet.ServletException;
07:  import javax.servlet.annotation.WebServlet;
08:  import javax.servlet.http.HttpServlet;
09:  import javax.servlet.http.HttpServletRequest;
10:  import javax.servlet.http.HttpServletResponse;
11:
12:  @WebServlet("/Kuku")
13:  public class Kuku extends HttpServlet {
14:
15:    @Override
16:    protected void doGet(HttpServletRequest request,
17:        HttpServletResponse response) throws ServletException,
18:        IOException {
19:
20:      PrintWriter out = response.getWriter();
21:      out.println("<!DOCTYPE html>");
22:      out.println("<html>");
23:      out.println("<head>");
24:      out.println("<meta charset=¥"UTF-8¥"/>");
25:      out.println("<title>九九の表</title>");
26:      out.println("</head>");
27:      out.println("<body>");
28:      out.println("<h3>九九の表</h3>");
29:      out.println("<table border=¥"3¥">");
30:      for (int i = 0; i < 10; i++) {
31:        out.println("<tr>");
32:        for (int j = 0; j < 10; j++) {
33:          if (i == 0) {
34:            if (j == 0) {
35:              out.println("<td> </td>");
36:            } else {
37:              out.println("<th>" + j + "</th>");
38:            }
39:          } else {
40:            if (j == 0) {
41:              out.println("<th>" + i + "</th>");
```

```
42:           } else {
43:             out.println("<td style=¥"text-align: right¥">" + (i * j)
44:                 + "</td>");
45:           }
46:         }
47:       }
48:       out.println("</tr>");
49:     }
50:     out.println("</table>");
51:     out.println("<p>以上");
52:     out.println("</body>");
53:     out.println("</html>");
54:   }
55: }
```

なお、ExampleFilter.javaを準備します。

練習2　お天気予想

プロジェクト名　practice2_2

ファイル　WeatherServlet.java

```
01: package pac1;
02:
03: import java.io.IOException;
04: import java.io.PrintWriter;
05:
06: import javax.servlet.ServletException;
07: import javax.servlet.annotation.WebServlet;
08: import javax.servlet.http.HttpServlet;
09: import javax.servlet.http.HttpServletRequest;
10: import javax.servlet.http.HttpServletResponse;
11:
12: /**
13:  * Servlet implementation class TenkiServlet
14:  */
15: @WebServlet("/WeatherServlet")
16: public class WeatherServlet extends HttpServlet {
17:
18:   @Override
19:   protected void doGet(HttpServletRequest request,
20:       HttpServletResponse response)
21:       throws ServletException, IOException {
22:
23:     PrintWriter out = response.getWriter();
24:     out.println("<html>");
25:     out.println("<head>");
26:     out.println("<title>お天気予想</title>");
27:     out.println("</head>");
```

```
28:         out.println("<body>");
29:         out.println("<h1>お天気を予想します</h1>");
30:         double weather = Math.random();
31:         if (weather <= 0.5d) {
32:           out.println("天気は、晴れです。<p>");
33:         } else if (weather <= 0.8d) {
34:           out.println("天気は、くもりですねぇ。<p>");
35:         } else {
36:           out.println("天気は、雨でしょう。<p>");
37:         }
38:         out.println("</body>");
39:         out.println("</html>");
40:       }
41:     }
```

なお、ExampleFilter.javaを準備します。

練習3　アクセスカウンタ

プロジェクト名　practice2_3

ファイル　AccessCounterServlet.java

```
01:     package pac1;
02:
03:     import java.io.IOException;
04:     import java.io.PrintWriter;
05:
06:     import javax.servlet.ServletException;
07:     import javax.servlet.annotation.WebServlet;
08:     import javax.servlet.http.HttpServlet;
09:     import javax.servlet.http.HttpServletRequest;
10:     import javax.servlet.http.HttpServletResponse;
11:
12:     /**
13:      * Servlet implementation class AccessCounterServlet
14:      */
15:     @WebServlet("/AccessCounterServlet")
16:     public class AccessCounterServlet extends HttpServlet {
17:
18:       int accessCount = 0;
19:
20:       @Override
21:       protected void doGet(HttpServletRequest request,
22:           HttpServletResponse response)
23:           throws ServletException, IOException {
24:
25:         accessCount++;
26:         PrintWriter out = response.getWriter();
27:                         out.println("<!DOCTYPE html>");
```

```
28:                     out.println("<html>");
29:                     out.println("<head>");
30:                     out.println("<meta charset=¥"UTF-8¥"/>");
31:       out.println("<title>アクセスカウンタ</title>");
32:       out.println("</head>");
33:       out.println("<body>");
34:       out.println("<h1>アクセスカウンタ</h1>");
35:       out.println("あなたは、" + accessCount + "人目の訪問者です！");
36:       out.println("</body>");
37:       out.println("</html>");
38:     }
39:   }
```

練習4　数当て

プロジェクト名　practice2_4

ファイル　HitNumberServlet.java

```
01:   package pac1;
02:
03:   import java.io.IOException;
04:   import java.io.PrintWriter;
05:
06:   import javax.servlet.ServletException;
07:   import javax.servlet.annotation.WebServlet;
08:   import javax.servlet.http.HttpServlet;
09:   import javax.servlet.http.HttpServletRequest;
10:   import javax.servlet.http.HttpServletResponse;
11:
12:   /**
13:    * Servlet implementation class NumberateServlet
14:    */
15:   @WebServlet("/HitNumberServlet")
16:   public class HitNumberServlet extends HttpServlet {
17:
18:     int targetNumber = 0;
19:     int times = 0;
20:
21:     @Override
22:     protected void doGet(HttpServletRequest request,
23:         HttpServletResponse response)
24:         throws ServletException, IOException {
25:
26:       if (times == 0) {
27:         targetNumber = (int) (Math.random() * 9d) + 1;
28:       }
29:       times++;
30:       String Number_s = request.getParameter("number");
31:       int enteredNumber = Integer.parseInt(Number_s);
```

```
32:
33:         String result = "はずれ～！";
34:         if (enteredNumber == targetNumber) {
35:           result = "当たり";
36:         }
37:
38:         PrintWriter out = response.getWriter();
39:                         out.println("<!DOCTYPE html>");
40:                         out.println("<html>");
41:                         out.println("<head>");
42:                         out.println("<meta charset=¥"UTF-8¥"/>");
43:         out.println("<title>数当てゲーム結果</title>");
44:         out.println("</head>");
45:         out.println("<body>");
46:         out.println("<h1>数当てゲーム結果</h1>");
47:         out.println(times + "回目　あなたが選んだ数：" + enteredNumber);
48:         out.println("<br>");
49:         out.println("<br>");
50:         out.println("結果は、" + result);
51:         out.println("<br>");
52:         out.println("<br>");
53:         out.println("<a href=¥"hitnumber.html¥">もう一度やる</a>");
54:         out.println("</body>");
55:         out.println("</html>");
56:         if (enteredNumber == targetNumber) {
57:           times = 0;
58:         }
59:       }
60:     }
```

ファイル　hitnumber.html

```
01:     <!DOCTYPE html>
02:
03:     <html>
04:     <head>
05:     <meta charset="UTF-8">
06:     <title>数当てゲーム</title>
07:     </head>
08:     <body>
09:     <h1>数当てゲーム</h1>
10:     コンピュータが考えている数字（1～9）を当てましょう。
11:     <br>
12:     <br>
13:     <a href="HitNumberServlet?number=1">1</a> 
14:     <a href="HitNumberServlet?number=2">2</a> 
15:     <a href="HitNumberServlet?number=3">3</a> 
16:     <a href="HitNumberServlet?number=4">4</a> 
17:     <a href="HitNumberServlet?number=5">5</a> 
18:     <a href="HitNumberServlet?number=6">6</a> 
```

```
19:     <a href="HitNumberServlet?number=7">7</a> 
20:     <a href="HitNumberServlet?number=8">8</a> 
21:     <a href="HitNumberServlet?number=9">9</a> 
22:     </body>
23:     </html>
```

練習5　単純な電卓

プロジェクト名　practice2_5

ファイル　CalcServlet.java

```
01:     package pac1;
02:
03:     import java.io.IOException;
04:     import java.io.PrintWriter;
05:
06:     import javax.servlet.ServletException;
07:     import javax.servlet.annotation.WebServlet;
08:     import javax.servlet.http.HttpServlet;
09:     import javax.servlet.http.HttpServletRequest;
10:     import javax.servlet.http.HttpServletResponse;
11:
12:     /**
13:      * Servlet implementation class KeisanServlet
14:      */
15:     @WebServlet("/CalcServlet")
16:     public class CalcServlet extends HttpServlet {
17:
18:       @Override
19:       protected void doGet(HttpServletRequest request,
20:           HttpServletResponse response)
21:           throws ServletException, IOException {
22:
23:         String strnumber1 = request.getParameter("number1");
24:         int number1 = Integer.parseInt(strnumber1);
25:         String strnumber2 = request.getParameter("number2");
26:         int number2 = Integer.parseInt(strnumber2);
27:         String execute = request.getParameter("execute");
28:
29:         PrintWriter out = response.getWriter();
30:         out.println("<!DOCTYPE html>");
31:         out.println("<html>");
32:         out.println("<head>");
33:         out.println("<meta charset=¥"UTF-8¥"/>");
34:         out.println("<title>四則演算結果</title>");
35:         out.println("</head>");
36:         out.println("<body>");
37:         out.println("<h1>四則演算の結果</h1>");
```

```
38:         out.println("<table>");
39:         out.println("<tr><td>数字１つ目</td><td>  </td><td>数字２つ目</td><td>  </td><td>
40:       結果</td></tr>");
41:         out.println("<tr style=¥"text-align: center¥"><td>" + number1 + "</td><td>");
42:         switch (execute) {
43:           case "addition": out.print("＋");break;
44:           case "subtraction": out.print("－");break;
45:           case "multiplication": out.print("×");break;
46:           case "division": out.print("÷");break;
47:         }
48:         out.println("</td><td>" + number2 + "</td><td>＝</td><td>");
49:         switch (execute) {
50:           case "addition":out.print(number1 + number2);break;
51:           case "subtraction":out.print(number1 - number2);break;
52:           case "multiplication":out.print(number1 * number2);break;
53:           case "division":out.print(number1 / number2);break;
54:         }
55:         out.println("</td></tr>");
56:         out.println("</table>");
57:         out.println("</body>");
58:         out.println("</html>");
59:       }
60:     }
```

ファイル calc.html

```
01:     <!DOCTYPE html>
02:
03:     <html>
04:     <head>
05:     <meta charset="UTF-8">
06:     <head><title>四則演算処理</title></head>
07:     <body>
08:     <h1>四則演算処理</h1>
09:     <form action="CalcServlet" method="get">
10:       <table>
11:         <tr>
12:           <td style="text-align: center">数字１つ目<br>
13:             <input size="5" type="text" name="number1" value="30"></td>
14:           <td>
15:             <input type="radio" name="execute" value="addition" checked>＋<br>
16:             <input type="radio" name="execute" value="subtraction">－<br>
17:             <input type="radio" name="execute" value="multiplication">×<br>
18:             <input type="radio" name="execute" value="division">÷</td>
19:           <td style="text-align: center">数字２つ目<br>
20:             <input size="5" type="text" name="number2" value="20"></td>
21:           <td style="text-align: center">
22:             <input type="submit" value="="></td>
23:         </tr>
24:       </table>
```

```
25:     </form>
26:     </body>
27:     </html>
```

練習6-1 じゃんけん

プロジェクト名 practice2_6

ファイル JankenServlet.java

```
01:     package pac1;
02:
03:     import java.io.IOException;
04:     import java.io.PrintWriter;
05:
06:     import javax.servlet.ServletException;
07:     import javax.servlet.annotation.WebServlet;
08:     import javax.servlet.http.HttpServlet;
09:     import javax.servlet.http.HttpServletRequest;
10:     import javax.servlet.http.HttpServletResponse;
11:
12:     /**
13:      * Servlet implementation class JankenServlet
14:      */
15:     @WebServlet("/JankenServlet")
16:     public class JankenServlet extends HttpServlet {
17:
18:       @Override
19:       protected void doGet(HttpServletRequest request,
20:           HttpServletResponse response) throws ServletException,
21:           IOException {
22:
23:         /**
24:          * 人が選んだ手を受け取る
25:          */
26:         String player = request.getParameter("hand");
27:
28:         /**
29:          * コンピュータの手を決める
30:          */
31:         double jankenPon = Math.random();
32:         String computer = "";
33:         if (jankenPon <= 0.3d) {
34:           computer = "g";
35:         } else if (jankenPon <= 0.6d) {
36:           computer = "c";
37:         } else {
38:           computer = "p";
39:         }
40:
```

```
41:       /**
42:        * 勝負の判定
43:        */
44:       String result = "";
45:       if (computer.equals(player)) {
46:         result = "あいこ";
47:       } else if ((computer.equals("g")) && (player.equals("c"))) {
48:         result = "Computer";
49:       } else if ((computer.equals("c")) && (player.equals("p"))) {
50:         result = "Computer";
51:       } else if ((computer.equals("p")) && (player.equals("g"))) {
52:         result = "Computer";
53:       } else {
54:         result = "Player";
55:       }
56:       if (result.equals("あいこ")) {
57:         result = "引き分け";
58:       } else if (result.equals("Computer")) {
59:         result = "プログラムの勝ち";
60:       } else {
61:         result = "あなたが勝ち";
62:       }
63:
64:       PrintWriter out = response.getWriter();
65:       out.println("<!DOCTYPE html>");
66:       out.println("<html>");
67:       out.println("<head>");
68:       out.println("<meta charset=¥"UTF-8¥"/>");
69:       out.println("<title>じゃんけん結果</title>");
70:       out.println("</head>");
71:       out.println("<body>");
72:       out.println("<h1>じゃんけん結果</h1>");
73:       out.println("あなたが選んだ手：" + Judgment(player) + "<br>");
74:       out.println("プログラムが選んだ手：" + Judgment(computer) + "<br>");
75:       out.println("勝負の結果は、" + result + "でした！");
76:       out.println("<br>");
77:       out.println("<br>");
78:       out.println("<a href=¥"janken.html¥">もう一度やる</a>");
79:       out.println("</body>");
80:       out.println("</html>");
81:     }
82:
83:     String Judgment(String hand) {
84:       switch (hand) {
85:       case "g":
86:         return "グー";
87:       case "c":
88:         return "チョキ";
89:       default:
90:         return "パー";
```

```
91:         }
92:
93:       }
94:     }
```

ファイル janken.html

```
01:     <!DOCTYPE html>
02:     <html>
03:     <head>
04:     <meta charset="UTF-8">
05:     <title>じゃんけんゲーム</title>
06:     </head>
07:     <body>
08:       <H1>じゃんけんゲーム</H1>
09:        あなたの選ぶ手は<br>
10:       <br>
11:       <a href="JankenServlet?hand=g">グー</a>
12:       <br>
13:       <br>
14:       <a href="JankenServlet?hand=c">チョキ</a>
15:       <br>
16:       <br>
17:       <a href="JankenServlet?hand=p">パー</a>
18:     </body>
19:     </html>
```

練習6-2 じゃんけん応用問題

プロジェクト名 practice2_6

ファイル JankenServlet2.java

```
01:     package pac1;
02:
03:     import java.io.IOException;
04:     import java.io.PrintWriter;
05:
06:     import javax.servlet.ServletException;
07:     import javax.servlet.annotation.WebServlet;
08:     import javax.servlet.http.HttpServlet;
09:     import javax.servlet.http.HttpServletRequest;
10:     import javax.servlet.http.HttpServletResponse;
11:
12:     /**
13:      * Servlet implementation class JankenServlet
14:      */
15:     @WebServlet("/JankenServlet2")
16:     public class JankenServlet2 extends HttpServlet {
```

```
17:
18:     int times = 0;
19:     int winComputer = 0;
20:     int winPlayer = 0;
21:
22:     @Override
23:     protected void doGet(HttpServletRequest request,
24:         HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:       /**
28:        * 人が選んだ手を受け取る
29:        */
30:       String player = request.getParameter("hand");
31:
32:       /**
33:        * コンピュータの手を決める
34:        */
35:       double jankenPon = Math.random();
36:       String computer = "";
37:       if (jankenPon <= 0.3d) {
38:             computer = "g";
39:       } else if (jankenPon <= 0.6d) {
40:             computer = "c";
41:       } else {
42:             computer = "p";
43:       }
44:
45:       times++;
46:
47:       /**
48:        * 勝負の判定
49:        */
50:       String result = "";
51:       if (computer.equals(player)) {
52:             result = "あいこ";
53:       } else if ((computer.equals("g")) && (player.equals("c"))) {
54:             result = "Computer";
55:       } else if ((computer.equals("c")) && (player.equals("p"))) {
56:             result = "Computer";
57:       } else if ((computer.equals("p")) && (player.equals("g"))) {
58:             result = "Computer";
59:       } else {
60:             result = "Player";
61:       }
62:       if (result.equals("あいこ")) {
63:             result = "引き分け";
64:       } else if (result.equals("Computer")) {
65:         winComputer++;
66:         result = "プログラムの勝ち";
```

```
67:       } else {
68:         winPlayer++;
69:         result = "あなたが勝ち";
70:       }
71:
72:       PrintWriter out = response.getWriter();
73:                       out.println("<!DOCTYPE html>");
74:                       out.println("<html>");
75:                       out.println("<head>");
76:                       out.println("<meta charset=¥"UTF-8¥"/>");
77:       out.println("<title>じゃんけん結果</title>");
78:       out.println("</head>");
79:       out.println("<body>");
80:       out.println("<h1>じゃんけん結果（" + times + "回目）</h1>");
81:       out.println("あなたが選んだ手：" + Judgment(player) + "("
82:           + winPlayer + "勝)<br>");
83:       out.println("プログラムが選んだ手：" + Judgment(computer) + "("
84:           + winComputer + "勝)<br>");
85:       out.println("勝負の結果は、" + result + "でした！");
86:       out.println("<br>");
87:       out.println("<br>");
88:       out.println("<a href=¥"janken2.html¥">もう一度やる</a>");
89:       out.println("</body>");
90:       out.println("</html>");
91:     }
92:
93:     String Judgment(String hand) {
94:       switch (hand) {
95:       case "g":
96:         return "グー";
97:       case "c":
98:         return "チョキ";
99:       default:
100:        return "パー";
101:      }
102:
103:    }
104:  }
```

ファイル janken2.html

```
01:   <!DOCTYPE html>
02:
03:   <html lang="ja">
04:   <head>
05:   <meta charset="UTF-8">
06:   <title>じゃんけんゲーム(応用編)</title>
07:   </head>
08:   <body>
09:     <h1>じゃんけんゲーム（応用編）</h1>
```

```
10:       あなたの選ぶ手は<br>
11:       <br>
12:       <a href="JankenServlet2?hand=g">グー</a>
13:       <br>
14:       <br>
15:       <a href="JankenServlet2?hand=c">チョキ</a>
16:       <br>
17:       <br>
18:       <a href="JankenServlet2?hand=p">パー</a>
19:     </body>
20:     </html>
```

第3章　練習問題の模範解答

練習1　グラデーション その1

プロジェクト名　practice3_1

ファイル　gradientjsp1.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>グラデーション JSP その 1</title>
</head>
<body>
<% int add = 16; %>

<h1>グラデーション JSP その 1</h1>
グラデーションの幅：<%= add %><p>

<table border="1">
  <tr>
<%
  for (int color = 0; color <= 255; color += add) {
    String colorHex = String.format("%02x", color);
                                          // color を 16 進数(2 ケタ)に変換
    String colorHTML = colorHex + colorHex + colorHex;
                                          // HTML の色指定形式の作成
    out.println("    <td style=¥"height: 30px; width: 30px; "
        + "background-color: #"
        + colorHTML + ";¥"><br></td>");
  }
%>
  </tr>
</table>
</body>
31：</html>
```

練習2 グラデーション その2

プロジェクト名 practice3_2

ファイル gradientjsp2.jsp

```
01:  <%@ page language="java" contentType="text/html; charset=UTF-8"
02:      pageEncoding="UTF-8"%>
03:  <!DOCTYPE html>
04:  <html>
05:  <head>
06:  <meta charset="UTF-8">
07:  <title>グラデーション JSP その 2</title>
08:  </head>
09:  <body>
10:
11:  <% int add = 42; %>
12:
13:  <h1>グラデーション JSP その 2</h1>
14:  グラデーションの幅：<%= add %><p>
15:  <table border="1">
16:  <%
17:    int color;
18:
19:    for(int i=0; i<4; i++){
20:      out.println("  <tr>");
21:      color = i*add;
22:
23:      for (int j=0; j<4; j++) {
24:        String colorHex = String.format("%02x", color);
25:                                        // color を 16 進数(2 ケタ)に変換
26:        String colorHTML = colorHex + colorHex + colorHex;
27:                                        // HTML の色指定形式の作成
28:        out.println("    <td style=¥"height: 50px; width: 50px; background-color: #"
29:            + colorHTML + ";¥"><br></td>");
30:        color += add;
31:      }
32:
33:      out.println("  </tr>");
34:    }
35:  %>
36:  </table>
37:
38:  </body>
39:  </html>
```

第4章　練習問題の模範解答

練習1　クッキーの登録

プロジェクト名　practice4_1

ファイル　ReadCookieServlet.java

```
01:  package pac1;
02:
03:  import java.io.IOException;
04:  import java.io.PrintWriter;
05:
06:  import javax.servlet.ServletException;
07:  import javax.servlet.annotation.WebServlet;
08:  import javax.servlet.http.Cookie;
09:  import javax.servlet.http.HttpServlet;
10:  import javax.servlet.http.HttpServletRequest;
11:  import javax.servlet.http.HttpServletResponse;
12:
13:  /**
14:   * Servlet implementation class ReadCookieServlet
15:   */
16:  @WebServlet("/ReadCookieServlet")
17:  public class ReadCookieServlet extends HttpServlet {
18:
19:    @Override
20:    protected void doGet(HttpServletRequest request,
21:        HttpServletResponse response) throws ServletException,
22:        IOException {
23:
24:      PrintWriter out = response.getWriter();
25:
26:      out.println("<!DOCTYPE html>");
27:      out.println("<html>");
28:      out.println("<head>");
29:      out.println("<meta charset=¥"UTF-8¥">");
30:      out.println("<title>クッキー情報</title>");
31:      out.println("</head>");
32:      out.println("<body>");
33:      out.println("<h1>クッキー情報</h1>");
34:      Cookie[] cookies = request.getCookies();
35:      if (cookies != null) {
36:        String name = "";
37:        int count = 0;
38:        for (Cookie cookie : cookies) {
39:          if (cookie.getName().equals("MyName")) {
40:            name = cookie.getValue();
41:          }
```

```
42:         if (cookie.getName().equals("Count")) {
43:           count = Integer.parseInt(cookie.getValue());
44:           count++;
45:           String strCount = count + "";
46:           Cookie cookCount = new Cookie("Count", strCount);
47:           cookCount.setMaxAge(30);
48:           response.addCookie(cookCount);
49:         }
50:       }
51:       out.println("<p>");
52:       out.println(name + "さん");
53:       out.println("<p>");
54:       out.println("こんにちは" + count + "回目の訪問ですね");
55:     } else {
56:       out.println("<p>");
57:       out.println("初めての訪問ですね。");
58:       out.println("<p>");
59:       out.println("My 情報を登録しましょう");
60:       out.println("<p>");
61:       out.println("<a href=¥"cookieinfo.html¥">登録</a>");
62:     }
63:     out.println("</body>");
64:     out.println("</html>");
65:   }
66: }
```

ファイル WriteCookie.java

```
01: package pac1;
02:
03: import java.io.IOException;
04: import java.io.PrintWriter;
05:
06: import javax.servlet.ServletException;
07: import javax.servlet.annotation.WebServlet;
08: import javax.servlet.http.Cookie;
09: import javax.servlet.http.HttpServlet;
10: import javax.servlet.http.HttpServletRequest;
11: import javax.servlet.http.HttpServletResponse;
12:
13: /**
14:  * Servlet implementation class WriteCookieServlet
15:  */
16: @WebServlet("/WriteCookieServlet")
17: public class WriteCookieServlet extends HttpServlet {
18:
19:   @Override
20:   protected void doPost(HttpServletRequest request,
21:       HttpServletResponse response) throws ServletException,
22:       IOException {
```

```
23:
24:         String myno = request.getParameter("myno");
25:         String myname = request.getParameter("myname");
26:         String mymail = request.getParameter("mymail");
27:
28:         Cookie cookMyNo = new Cookie("MyNo", myno);
29:         cookMyNo.setMaxAge(30);
30:         response.addCookie(cookMyNo);
31:         Cookie cookMyName = new Cookie("MyName", myname);
32:         cookMyName.setMaxAge(30);
33:         response.addCookie(cookMyName);
34:         Cookie cookMyMail = new Cookie("MyMail", mymail);
35:         cookMyMail.setMaxAge(30);
36:         response.addCookie(cookMyMail);
37:         Cookie cookCount = new Cookie("Count", "1");
38:         cookCount.setMaxAge(30);
39:         response.addCookie(cookCount);
40:
41:         PrintWriter out = response.getWriter();
42:         out.println("<!DOCTYPE html>");
43:         out.println("<html>");
44:         out.println("<head>");
45:         out.println("<meta charset=¥"UTF-8¥">");
46:         out.println("<title>クッキー書き込み</title>");
47:         out.println("</head>");
48:         out.println("<body>");
49:         out.println("<h1>クッキー情報</h1>");
50:         out.println("<p>");
51:         out.println("MyNo: " + myno);
52:         out.println("<p>");
53:         out.println("MyName: " + myname);
54:         out.println("<p>");
55:         out.println("MyMail: " + mymail);
56:         out.println("<p>");
57:         out.println("クッキーを書きました");
58:         out.println("<p>");
59:         out.println("<a href=¥"ReadCookieServlet¥">戻る</a>");
60:         out.println("</body>");
61:         out.println("</html>");
62:       }
63:
64:     }
```

ファイル cookieinfo.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>クッキー情報登録</title>
</head>
<body>
          <h1>My 情報登録</h1>
          <form action="WriteCookieServlet" method="post">
                    MyNo <input type="text" name="myno">
                    MyName <input type="text" name="myname">
                    MyMail <input type="text" name="mymail">
                    <input type="submit" value="登録">
          </form>
</body>
</html>
```

第5章　練習問題の模範解答

練習1 「お小遣い帳」アプリ作成の下準備～入金レコードと Beans クラスの作成

プロジェクト名　practice5_1

ファイル PaymentRecordBean.java

```
01:  package pac1;
02:
03:  import java.io.Serializable;
04:
05:  public class PaymentRecordBean implements Serializable {
06:
07:    private int payment;    // 入金額
08:    private String comment; // コメント
09:
10:    //Constructor
11:    public PaymentRecordBean() {
12:    }
13:
14:    public void setPayment(int payment) {
15:      this.payment = payment;
16:    }
17:
18:    public int getPayment() {
19:      return this.payment;
20:    }
21:
22:    public void setComment(String comment) {
23:      this.comment = comment;
24:    }
25:
26:    public String getComment() {
27:      return this.comment;
28:    }
29:
30:  }
```

ファイル PettyCashbookBean.java

```
01:  package pac1;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class PettyCashbookBean implements Serializable {
07:
08:    private ArrayList<PaymentRecordBean> paymentHistoryArray
```

```
09:         = new ArrayList<PaymentRecordBean>(); // 貯金の履歴
10:
11:     //Constructor
12:     public PettyCashbookBean() {
13:     }
14:
15:     public ArrayList<PaymentRecordBean> getPaymentHistoryArray() {
16:       return paymentHistoryArray;
17:     }
18:
19:     public void addPaymentHistoryArray(PaymentRecordBean payment) {
20:       paymentHistoryArray.add(payment);
21:     }
22:
23:     public int getTotalPayment() {
24:       int total = 0;
25:
26:       for (PaymentRecordBean rec : paymentHistoryArray) {
27:         total += rec.getPayment();
28:       }
29:       return total;
30:     }
31:
32:   }
```

練習2 「お小遣い帳」アプリの完成

プロジェクト名　practice5_1

ファイル　pettycashbook_input.html

```
01:   <!DOCTYPE html>
02:   <html>
03:   <head>
04:   <meta charset="UTF-8">
05:   <title>お小遣い帳 Beans - 入力画面 HTML</title>
06:   </head>
07:   <body>
08:
09:   <h1>お小遣い帳 Beans - 入力画面 HTML</h1>
10:   <form action="PettyCashbookServlet" method="get">
11:     <br>
12:       貯金額を入力 <input type="text" name="payment" size="3"><br>
13:       コメント <input type="text" name="comment" size="20">
14:     <br>
15:     <input type="submit" value="送信">
16:   </form>
```

```
17:
18:     </body>
19:     </html>
```

ファイル pettycashbook_output.jsp

```
01:     <%@ page import="pac1.PaymentRecordBean" %>
02:     <%@ page import="java.util.ArrayList" %>
03:     <%@ page language="java" contentType="text/html; charset=UTF-8"
04:      pageEncoding="UTF-8" %>
05:     <html>
06:     <head>
07:     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
08:     <title>お小遣い帳 Beans - 出力画面 JSP</title>
09:     </head>
10:     <body>
11:     <jsp:useBean id="pettyCashbookBean" class="pac1.PettyCashbookBean"
12:      scope="session" />
13:
14:     <h1>お小遣い帳 Beans - 出力画面 JSP</h1>
15:     貯金履歴の一覧表示<p>
16:
17:     <table border="1">
18:     <tr>
19:         <th>履歴</th>
20:         <th>貯金額</th>
21:         <th>コメント</th>
22:     </tr>
23:     <%
24:     ArrayList<PaymentRecordBean> paymentHistoryArray =
25:       pettyCashbookBean.getPaymentHistoryArray();
26:
27:     int i = 1;
28:     for(PaymentRecordBean paymentRec : paymentHistoryArray){
29:       out.println("<tr><td>" + i + "回目</td><td>"
30:           + paymentRec.getPayment() + "円</td><td>"
31:           + paymentRec.getComment() + "</td></tr>");
32:       i++;
33:     }
34:     %>
35:     </table>
36:
37:     <br>
38:     <b>合計金額 ： <%= pettyCashbookBean.getTotalPayment()  %>円</b>
39:     <br>
40:     <br>
41:     <a href="pettycashbook_input.html">さらに入金する</a><br>
42:
43:     </body>
44:     </html>
```

ファイル PettyCashbookServlet.java

```
01:  package pac1;
02:
03:  import java.io.IOException;
04:
05:  import javax.servlet.ServletException;
06:  import javax.servlet.annotation.WebServlet;
07:  import javax.servlet.http.HttpServlet;
08:  import javax.servlet.http.HttpServletRequest;
09:  import javax.servlet.http.HttpServletResponse;
10:  import javax.servlet.http.HttpSession;
11:
12:  /**
13:   * Servlet implementation class PettyCashbookServlet
14:   */
15:  @WebServlet("/PettyCashbookServlet")
16:  public class PettyCashbookServlet extends HttpServlet {
17:
18:    PettyCashbookBean pettyCashbookBean = new PettyCashbookBean();
19:
20:    @Override
21:    protected void doGet(HttpServletRequest request,
22:        HttpServletResponse response) throws ServletException,
23:        IOException {
24:
25:      int payment;    // 入金額
26:      String comment; // コメント
27:
28:      /* Form から入力された値の取り出し */
29:      try {
30:        payment = Integer.parseInt(request.getParameter("payment"));
31:        comment = request.getParameter("comment");
32:
33:      } catch (NumberFormatException e) {
34:        getServletContext().getRequestDispatcher("/errorinput.html")
35:            .forward(request, response);
36:        return;
37:      }
38:
39:      /* 入金レコードの作成 */
40:      PaymentRecordBean paymentRec = new PaymentRecordBean();
41:      paymentRec.setPayment(payment);
42:      paymentRec.setComment(comment);
43:
44:      /* Beans 実行 */
45:      pettyCashbookBean.addPaymentHistoryArray(paymentRec); // 入金レコードを Beans へ追
46:  加
47:
48:      /* セッションに Beans を登録し、JSP で出力 */
49:      HttpSession session = request.getSession();
```

```
50:         session.setAttribute("pettyCashbookBean", pettyCashbookBean);
51:         getServletContext().getRequestDispatcher(
52:             "/pettycashbook_output.jsp").forward(request, response);
53:
54:       }
55:     }
```

第6章　練習問題の模範解答

練習1　科目表

プロジェクト名　practice6_1

ファイル SelectServlet1.java

```
01:  package pac1;
02:
03:  import java.io.IOException;
04:  import java.io.PrintWriter;
05:  import java.sql.Connection;
06:  import java.sql.DriverManager;
07:  import java.sql.ResultSet;
08:  import java.sql.Statement;
09:  import java.util.ArrayList;
10:
11:  import javax.servlet.ServletException;
12:  import javax.servlet.annotation.WebServlet;
13:  import javax.servlet.http.HttpServlet;
14:  import javax.servlet.http.HttpServletRequest;
15:  import javax.servlet.http.HttpServletResponse;
16:
17:  /**
18:   * Servlet implementation class InsertSampleServlet
19:   */
20:  @WebServlet("/SelectServlet1")
21:  public class SelectServlet1 extends HttpServlet {
22:
23:    @Override
24:    protected void doGet(HttpServletRequest request,
25:        HttpServletResponse response) throws ServletException,
26:        IOException {
27:
28:      ArrayList<String> list = new ArrayList<String>();
29:
30:      try {
31:        Class.forName("com.mysql.jdbc.Driver"); // 検索に失敗する場合は必要
32:        try (Connection conn = DriverManager.getConnection(
33:            "jdbc:mysql://localhost:3306/sampledb?serverTimezone=JST",
34:            "user", "password");
35:            Statement stmt = conn.createStatement();
36:            ResultSet rs = stmt.executeQuery(
37:                "SELECT * FROM SUBJECT")) {
38:          while (rs.next()) {
39:            String s = "<td>" + rs.getInt("SUBJECTID") + "</td>"
40:                + "<td>" + rs.getString("SUBJECTNAME") + "</td>";
41:            list.add(s);
```

```
        }
      }
    } catch (Exception e) {
      e.printStackTrace();
    }

    PrintWriter out = response.getWriter();

    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<meta charset=¥"UTF-8¥">");
    out.println("<title>練習問題 6-1</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<table border=¥"1¥">");

    out.println("<tr>");
    out.println("<th>科目 ID</th>");
    out.println("<th>科目名</th>");
    out.println("</tr>");

    for (String str : list) {
      out.println("<tr>" + str + "</tr>");
    }

    out.println("</table>");
    out.println("</body>");
    out.println("</html>");
  }
}
```

なお、ExampleFilter.javaを準備します。

練習2　科目ごとの平均点

プロジェクト名　practice6_2

ファイル　SelectServlet2.java

```
package pac1;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;

import javax.servlet.ServletException;
```

```
12:  import javax.servlet.annotation.WebServlet;
13:  import javax.servlet.http.HttpServlet;
14:  import javax.servlet.http.HttpServletRequest;
15:  import javax.servlet.http.HttpServletResponse;
16:
17:  /**
18:   * Servlet implementation class InsertSampleServlet
19:   */
20:  @WebServlet("/SelectServlet2")
21:  public class SelectServlet2 extends HttpServlet {
22:
23:    @Override
24:    protected void doGet(HttpServletRequest request,
25:        HttpServletResponse response) throws ServletException,
26:        IOException {
27:
28:      ArrayList<String> list = new ArrayList<String>();
29:
30:      try {
31:        Class.forName("com.mysql.jdbc.Driver"); // 検索に失敗する場合は必要
32:        try (Connection conn = DriverManager.getConnection(
33:            "jdbc:mysql://localhost:3306/sampledb?serverTimezone=JST",
34:            "user", "password");
35:            Statement stmt = conn.createStatement();
36:            ResultSet rs = stmt.executeQuery(
37:                "SELECT SUBJECTNAME,AVG(SCORE) FROM GRADE,SUBJECT WHERE GRADE.SUBJECTID
38:  = SUBJECT.SUBJECTID GROUP BY SUBJECTNAME")) {
39:          while (rs.next()) {
40:            String s = "<td>" + rs.getString("SUBJECTNAME") + "</td>"
41:                + "<td>" + rs.getDouble("AVG(SCORE)") + "</td>";
42:            list.add(s);
43:          }
44:        }
45:      } catch (Exception e) {
46:        e.printStackTrace();
47:      }
48:
49:      PrintWriter out = response.getWriter();
50:
51:      out.println("<!DOCTYPE html>");
52:      out.println("<html>");
53:      out.println("<head>");
54:      out.println("<meta charset=¥"UTF-8¥">");
55:      out.println("<title>練習問題 6-2</title>");
56:      out.println("</head>");
57:      out.println("<body>");
58:      out.println("<table border=¥"1¥">");
59:      out.println("<tr>");
60:      out.println("<th>科目名</th>");
61:      out.println("<th>平均点</th>");
```

```
62:         out.println("</tr>");
63:         for (String str : list) {
64:           out.println("<tr>" + str + "</tr>");
65:         }
66:         out.println("</table>");
67:         out.println("</body>");
68:         out.println("</html>");
69:       }
70:     }
```

なお、ExampleFilter.javaを準備します。

練習3　科目を登録

プロジェクト名　practice6_3

ファイル　index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>練習 6-3</title>
</head>
<body>
 <h1>練習 6-3</h1>
 <h2>科目テーブルに新規科目を登録します。</h2>
 <form action="InsertServlet1" method="get">
  <div>
   科目 ID：<input type="text" name="subjectid">
  </div>
  <div>
   科目名：<input type="text" name="subjectname">
  </div>
  <div>
   <input type="submit" value="登録"><input type="reset">
  </div>
 </form>
</body>
</html>
```

ファイル　InsertServlet1.java

```
package pac1;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class InsertSampleServlet
 */
@WebServlet("/InsertServlet1")
public class InsertServlet1 extends HttpServlet {

```

```
21:     @Override
22:     protected void doGet(HttpServletRequest request,
23:         HttpServletResponse response) throws ServletException,
24:         IOException {
25:
26:       int ret = 0;
27:       int id = 0;
28:       String name = null;
29:
30:       try {
31:         id = Integer.parseInt(request.getParameter("subjectid"));
32:       } catch (NumberFormatException e) {
33:         return;
34:       }
35:       name = request.getParameter("subjectname");
36:
37:       try {
38:         Class.forName("com.mysql.jdbc.Driver"); // 検索に失敗する場合は必要
39:         try (Connection conn = DriverManager.getConnection(
40:             "jdbc:mysql://localhost:3306/sampledb?serverTimezone=JST",
41:             "user", "password");
42:             Statement stmt = conn.createStatement()) {
43:           ret = stmt.executeUpdate("INSERT INTO SUBJECT VALUES(" + id
44:               + ",¥'" + name + "¥')");
45:         }
46:       } catch (Exception e) {
47:         e.printStackTrace();
48:       }
49:
50:       PrintWriter out = response.getWriter();
51:
52:       out.println("<!DOCTYPE html>");
53:       out.println("<html>");
54:       out.println("<head>");
55:       out.println("<meta charset=¥"UTF-8¥">");
56:       out.println("<title>練習 6-3</title>");
57:       out.println("</head>");
58:       out.println("<body>");
59:       out.println("<h1>練習 6-3</h1>");
60:       out.println("<h2>" + ret + "件追加しました。</h2>");
61:       out.println("<p>");
62:       out.println("科目 ID：" + id + "<br>");
63:       out.println("科目名：" + name + "<br>");
64:       out.println("</body>");
65:       out.println("</html>");
66:     }
67:   }
```

なお、ExampleFilter.javaを準備します。

第11章　総合演習サンプルプログラム

第1段階

プロジェクト名 STEP1

```
STEP1
├─src
│  ├─chapter11
│  │  │  TweetArrayBean.java
│  │  │  TweetBean.java
│  │  │  TweetInputServlet.java
│  │  │  TweetListServlet.java
│  │  │  TweetPersonServlet.java
│  │  │
│  │  ├─dao
│  │  │      DAOBase.java
│  │  │      TweetDAO.java
│  │  │
│  │  ├─exception
│  │  │      DatabaseException.java
│  │  │      ParameterException.java
│  │  │      SystemException.java
│  │  │
│  │  └─parameter
│  │          DAOParameters.java
│  │          ExceptionParameters.java
│  │          TweetParameters.java
│  │
│  └─filter
│          ExampleFilter.java
│
└─WebContent
    │  error.jsp
    │  tweetinput.html
    │  tweetlist.jsp
    │  tweetpersonlist.jsp
    │
    └─WEB-INF
            web.xml
```

なお、ExampleFilter.javaは前章と同じためここでは、プログラムの掲載を省略します。

ファイル TweetArrayBean.java

```
01:  package chapter11;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class TweetArrayBean implements Serializable {
07:
08:    private ArrayList<TweetBean> TweetArray;
09:
10:    public TweetArrayBean() {
11:      TweetArray = new ArrayList<TweetBean>();
12:    }
13:
14:    public void addTweet(TweetBean obj) {
15:      TweetArray.add(obj);
16:    }
17:
18:    public int getArraySize() {
19:      return TweetArray.size();
20:    }
21:
22:    public ArrayList<TweetBean> getTweetArray() {
23:      return TweetArray;
24:    }
25:
26:    public void setTweetArray(ArrayList<TweetBean> TweetArray) {
27:      this.TweetArray = TweetArray;
28:    }
29:
30:  }
```

ファイル TweetBean.java

```
01:  package chapter11;
02:
03:  import java.io.Serializable;
04:  import java.util.Date;
05:
06:  public class TweetBean implements Serializable {
07:
08:    private String name;
09:    private String Tweet;
10:    private Date datetime;
11:
12:    public TweetBean() {
13:    }
14:
15:    public String getName() {
```

```
16:       return name;
17:     }
18:
19:     public void setName(String name) {
20:       this.name = name;
21:     }
22:
23:     public String getTweet() {
24:       return Tweet;
25:     }
26:
27:     public void setTweet(String Tweet) {
28:       this.Tweet = Tweet;
29:     }
30:
31:     public Date getDatetime() {
32:       return datetime;
33:     }
34:
35:     public void setDatetime(Date datetime) {
36:       this.datetime = datetime;
37:     }
38:
39:   }
```

ファイル TweetInputServlet.java

```
01:   package chapter11;
02:
03:   import java.io.IOException;
04:   import java.util.Date;
05:
06:   import javax.servlet.ServletException;
07:   import javax.servlet.annotation.WebServlet;
08:   import javax.servlet.http.HttpServlet;
09:   import javax.servlet.http.HttpServletRequest;
10:   import javax.servlet.http.HttpServletResponse;
11:   import javax.servlet.http.HttpSession;
12:
13:   import chapter11.dao.TweetDAO;
14:   import chapter11.exception.DatabaseException;
15:   import chapter11.exception.SystemException;
16:
17:   @WebServlet("/TweetInputServlet")
18:   public class TweetInputServlet extends HttpServlet {
19:
20:     @Override
21:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
22:         throws ServletException, IOException {
23:
```

```
24:         try {
25:           String name = request.getParameter("name");
26:           String Tweet = request.getParameter("tweet");
27:           Date date = new Date();
28:
29:           TweetBean tweetBean = new TweetBean();
30:           tweetBean.setName(name);
31:           tweetBean.setTweet(Tweet);
32:           tweetBean.setDatetime(date);
33:
34:           TweetDAO dao = new TweetDAO();
35:           dao.setTweet(tweetBean);
36:
37:           response.sendRedirect("TweetListServlet");
38:         } catch (SystemException e) {
39:           e.printStackTrace();
40:           HttpSession session = request.getSession();
41:           session.setAttribute("Except", e);
42:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:               response);
44:         } catch (DatabaseException e) {
45:           e.printStackTrace();
46:           HttpSession session = request.getSession();
47:           session.setAttribute("Except", e);
48:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
50:               response);
51:         }
52:       }
53:
54:     }
```

ファイル TweetListServlet.java

```
01:     package chapter11;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.TweetDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:
16:     @WebServlet("/TweetListServlet")
17:     public class TweetListServlet extends HttpServlet {
```

```
18:
19:       @Override
20:       protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:           throws ServletException, IOException {
22:
23:         TweetArrayBean TweetArray;
24:
25:         try {
26:           TweetDAO dao = new TweetDAO();
27:           TweetArray = dao.getTweetArray();
28:           HttpSession session = request.getSession();
29:           session.setAttribute("TweetArrayBean", TweetArray);
30:           getServletContext().getRequestDispatcher("/tweetlist.jsp").forward(
31:               request, response);
32:         } catch (SystemException e) {
33:           e.printStackTrace();
34:           HttpSession session = request.getSession();
35:           session.setAttribute("Except", e);
36:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
37:               response);
38:         } catch (DatabaseException e) {
39:           e.printStackTrace();
40:           HttpSession session = request.getSession();
41:           session.setAttribute("Except", e);
42:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:               response);
44:         }
45:       }
46:     }
```

ファイル TweetPersonServlet.java

```
01:     package chapter11;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.TweetDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:
16:     @WebServlet("/TweetPersonServlet")
17:     public class TweetPersonServlet extends HttpServlet {
18:
```

```
19:       @Override
20:       protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:           throws ServletException, IOException {
22:
23:         TweetArrayBean tweetArray;
24:
25:         try {
26:           String name = request.getParameter("name");
27:
28:           TweetDAO dao = new TweetDAO();
29:           tweetArray = dao.getPersonTweetArray(name);
30:           HttpSession session = request.getSession();
31:           session.setAttribute("TweetArrayBean", tweetArray);
32:           getServletContext().getRequestDispatcher("/tweetpersonlist.jsp")
33:               .forward(request, response);
34:         } catch (SystemException e) {
35:           e.printStackTrace();
36:           HttpSession session = request.getSession();
37:           session.setAttribute("Except", e);
38:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
39:               response);
40:         } catch (DatabaseException e) {
41:           e.printStackTrace();
42:           HttpSession session = request.getSession();
43:           session.setAttribute("Except", e);
44:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
45:               response);
46:         }
47:
48:       }
50:     }
```

ファイル DAOBase.java

```
01:  package chapter11.dao;
02:
03:  import java.sql.Connection;
04:  import java.sql.DriverManager;
05:  import java.sql.SQLException;
06:  import java.sql.Statement;
07:
08:  import chapter11.exception.DatabaseException;
09:  import chapter11.exception.SystemException;
10:  import chapter11.parameter.DAOParameters;
11:  import chapter11.parameter.ExceptionParameters;
12:
13:  public class DAOBase {
14:
15:    Connection con;
16:
17:    protected void open() throws DatabaseException, SystemException {
18:      try {
19:        Class.forName(DAOParameters.DRIVER_NAME);
20:        con = DriverManager.getConnection(DAOParameters.CONNECT_STRING,
21:            DAOParameters.USERID, DAOParameters.PASSWORD);
22:      } catch (ClassNotFoundException e) {
23:        throw new SystemException(ExceptionParameters.SYSTEM_EXCEPTION_MESSAGE, e);
24:      } catch (SQLException e) {
25:        throw new DatabaseException(
26:            ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
27:      }
28:    }
29:
30:    protected void close(Statement stmt) throws DatabaseException {
31:      try {
32:        if (stmt != null) {
33:          stmt.close();
34:        }
35:        if (con != null) {
36:          con.close();
37:        }
38:      } catch (SQLException e) {
39:        throw new DatabaseException(
40:            ExceptionParameters.DATABASE_CLOSE_EXCEPTION_MESSAGE, e);
41:      }
42:    }
43:
44:  }
```

ファイル TweetDAO.java

```
01:  package chapter11.dao;
02:
03:  import java.sql.PreparedStatement;
04:  import java.sql.ResultSet;
05:  import java.sql.SQLException;
06:  import java.sql.Statement;
07:  import java.text.SimpleDateFormat;
08:
09:  import chapter11.TweetArrayBean;
10:  import chapter11.TweetBean;
11:  import chapter11.exception.DatabaseException;
12:  import chapter11.exception.SystemException;
13:  import chapter11.parameter.ExceptionParameters;
14:  import chapter11.parameter.TweetParameters;
15:
16:  public class TweetDAO extends DAOBase {
17:
18:    public TweetArrayBean getTweetArray() throws DatabaseException,
19:        SystemException {
20:      Statement stmt = null;
21:      TweetArrayBean TweetArray = new TweetArrayBean();
22:      this.open();
23:      try {
24:        stmt = con.createStatement();
25:        ResultSet rs = stmt.executeQuery(TweetParameters.SQL_SELECT_ALL);
26:        while (rs.next()) {
27:          TweetBean record = new TweetBean();
28:          record.setName(rs.getString(TweetParameters.NAME));
29:          record.setTweet(rs.getString(TweetParameters.TWEET));
30:          record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
31:          TweetArray.addTweet(record);
32:        }
33:      } catch (SQLException e) {
34:        throw new DatabaseException(
35:            ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
36:      } finally {
37:        this.close(stmt);
38:      }
39:      return TweetArray;
40:    }
41:
42:    public TweetArrayBean getPersonTweetArray(String name)
43:        throws DatabaseException, SystemException {
44:      PreparedStatement stmt = null;
45:      TweetArrayBean TweetArray = new TweetArrayBean();
46:      this.open();
47:      try {
48:        stmt = con.prepareStatement(TweetParameters.SQL_SELECT_PERSON);
49:50:     stmt.setString(1, name);
```

```
51:         ResultSet rs = stmt.executeQuery();
52:         while (rs.next()) {
53:           TweetBean record = new TweetBean();
54:           record.setName(rs.getString(TweetParameters.NAME));
55:           record.setTweet(rs.getString(TweetParameters.TWEET));
56:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
57:           TweetArray.addTweet(record);
58:         }
59:       } catch (SQLException e) {
60:         throw new DatabaseException(
61:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
62:       } finally {
63:         this.close(stmt);
64:       }
65:       return TweetArray;
66:     }
67:
68:     public void setTweet(TweetBean Tweet) throws DatabaseException,
69:         SystemException {
70:       PreparedStatement stmt = null;
71:       this.open();
72:       try {
73:         stmt = con.prepareStatement(TweetParameters.SQL_INSERT);
74:         stmt.setString(1, Tweet.getName());
75:         stmt.setString(2, Tweet.getTweet());
76:         stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
77:             .format(Tweet.getDatetime()));
78:         stmt.executeUpdate();
79:       } catch (SQLException e) {
80:         throw new DatabaseException(
81:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
82:       } finally {
83:         this.close(stmt);
84:       }
85:     }
   }
```

ファイル DatabaseException.java

```
01: package chapter11.exception;
02:
03: public class DatabaseException extends Exception {
04:
05:   public DatabaseException() {
06:     super();
07:   }
08:
09:   public DatabaseException(String message, Throwable cause,
10:       boolean enableSuppression, boolean writableStackTrace) {
11:     super(message, cause, enableSuppression, writableStackTrace);
```

```
12:     }
13:
14:     public DatabaseException(String message, Throwable cause) {
15:       super(message, cause);
16:     }
17:
18:     public DatabaseException(String message) {
19:       super(message);
20:     }
21:
22:     public DatabaseException(Throwable cause) {
23:       super(cause);
24:     }
25:
26:   }
```

ファイル ParameterException.java

```
01:   package chapter11.exception;
02:
03:   public class ParameterException extends Exception {
04:
05:     public ParameterException() {
06:     }
07:
08:     public ParameterException(String message) {
09:       super(message);
10:     }
11:
12:     public ParameterException(Throwable cause) {
13:       super(cause);
14:     }
15:
16:     public ParameterException(String message, Throwable cause) {
17:       super(message, cause);
18:     }
19:
20:     public ParameterException(String message, Throwable cause,
21:         boolean enableSuppression, boolean writableStackTrace) {
22:       super(message, cause, enableSuppression, writableStackTrace);
23:     }
24:
25:   }
```

ファイル SystemException.java

```
01:  package chapter11.exception;
02:
03:  public class SystemException extends Exception {
04:
05:    public SystemException(String message, Throwable cause,
06:        boolean enableSuppression, boolean writableStackTrace) {
07:      super(message, cause, enableSuppression, writableStackTrace);
08:    }
09:
10:    public SystemException(String message, Throwable cause) {
11:      super(message, cause);
12:    }
13:
14:    public SystemException(String message) {
15:      super(message);
16:    }
17:
18:    public SystemException(Throwable cause) {
19:      super(cause);
20:    }
21:
22:  }
```

ファイル DAOParameters.java

```
01:  package chapter11.parameter;
02:
03:  public class DAOParameters {
04:    public static final String DRIVER_NAME = "com.mysql.jdbc.Driver";
05:    public static final String CONNECT_STRING =
06:  "jdbc:mysql://localhost:3306/tweet?serverTimezone=JST";
07:    public static final String USERID = "user";
08:    public static final String PASSWORD = "password";
09:  }
```

ファイル ExceptionParameters.java

```
01:  package chapter11.parameter;
02:
03:  public class ExceptionParameters {
04:    public static final String SYSTEM_EXCEPTION_MESSAGE = "システムエラーが発生しました";
05:
06:    public static final String DATABASE_CONNECTION_EXCEPTION_MESSAGE = "データベースへの
07:  接続時にエラーが発生しました";
08:    public static final String DATABASE_CLOSE_EXCEPTION_MESSAGE = "データベースからの切断
09:  時にエラーが発生しました";
10:
11:    public static final String Parameter_FORMAT_EXCEPTION_MESSAGE = "入力したデータの形式
```

```
12:     が正しくありません";
13:     }
```

ファイル TweetParameters.java

```
01:     package chapter11.parameter;
02:
03:     public class TweetParameters {
04:       public static final String SQL_SELECT_ALL = "SELECT * FROM TWEET ORDER BY DATE DESC";
05:       public static final String SQL_SELECT_PERSON = "SELECT * FROM TWEET WHERE NAME = ? ORDER
06:     BY DATE DESC";
07:
08:       public static final String SQL_INSERT = "INSERT INTO TWEET VALUES(?,?,?)";
09:
10:       public static final String NAME = "NAME";
11:       public static final String TWEET = "TWEET";
12:       public static final String DATE = "DATE";
13:     }
14:
```

ファイル error.jsp

```
01:     <%@ page language="java" contentType="text/html; charset=UTF-8"
02:      pageEncoding="UTF-8"%>
03:     <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
04:     "http://www.w3.org/TR/html4/loose.dtd">
05:     <html>
06:     <head>
07:     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
08:     <title>エラー</title>
09:     </head>
10:     <body>
11:      <h1>エラー画面</h1>
12:      <%
13:        Exception e = (Exception) session.getAttribute("Except");
14:      %>
15:      <p><%=e.getMessage()%>
16:      <p>
17:       <a href="javascript:history.back();">戻る</a>
18:     </body>
19:     </html>
```

ファイル tweetinput.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <h1>つぶやきアプリ</h1>
 <form action="TweetInputServlet" method="post">
  <p>
   名前:<br> <input type="text" name="name" maxlength="50">
  </p>
  <p>
   つぶやき:<br>
   <textarea rows="5" cols="40" name="tweet"></textarea>
  </p>
  <p>
   <input type="submit" value="送信">
          <input type="reset">
   <input type="button" onclick="location.href='TweetListServlet'"
    value="戻る">
 </form>
</html>
```

ファイル tweetlist.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="java.text.SimpleDateFormat"%>
<%@ page import="chapter11.*"%>
<%@ page import="java.net.URLEncoder"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="TweetArrayBean"
  class="chapter11.TweetArrayBean" scope="session" />
 <h1>つぶやきアプリ</h1>
 <form action="tweetinput.html" method="get">
  <input type="submit" value="書き込み">
 </form>
 <br>
 <table border="1">
  <tr>
```

```
23:        <th>名前</th>
24:        <th>つぶやき</th>
25:        <th>日時</th>
26:       </tr>
27:       <%
28:         ArrayList<TweetBean> TweetArray = TweetArrayBean
29:           .getTweetArray();
30:           for (TweetBean record : TweetArray) {
31:       %>
32:       <tr>
33:        <td>
34:          <a
35:     href="TweetPersonServlet?name=<%=URLEncoder.encode(record.getName(),"UTF-8")%>">
36:          <%=record.getName()%></a>
37:        </td>
38:        <td><%=record.getTweet()%></td>
39:        <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
40:              .format(record.getDatetime())%></td>
41:       </tr>
42:       <%
43:         }
44:       %>
45:      </table>
46:     </body>
47:     </html>
```

ファイル tweetpersonlist.jsp

```
01:     <%@ page language="java" contentType="text/html; charset=UTF-8"
02:      pageEncoding="UTF-8"%>
03:     <%@ page import="java.util.ArrayList"%>
04:     <%@ page import="java.text.SimpleDateFormat"%>
05:     <%@ page import="chapter11.*"%>
06:     <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
07:     "http://www.w3.org/TR/html4/loose.dtd">
08:     <html>
09:     <head>
10:     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11:     <title>つぶやきアプリ</title>
12:     </head>
13:     <body>
14:      <jsp:useBean id="TweetArrayBean"
15:       class="chapter11.TweetArrayBean" scope="session" />
16:      <h1>つぶやきアプリ</h1>
17:      <h2><%=request.getParameter("name")%>さんのつぶやき
18:      </h2>
19:      <form action="TweetListServlet" method="get">
20:      <input type="submit" value="戻る">
21:      </form>
22:      <br>
```

```
 <table border="1">
  <tr>
   <th>名前</th>
   <th>つぶやき</th>
   <th>日時</th>
  </tr>
  <%
    ArrayList<TweetBean> TweetArray = TweetArrayBean
      .getTweetArray();
      for (TweetBean record : TweetArray) {
  %>
  <tr>
   <td><%=record.getName()%></td>
   <td><%=record.getTweet()%></td>
   <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
         .format(record.getDatetime())%></td>
  </tr>
  <%
    }
  %>
 </table>
</body>
</html>
```

ファイル web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>STEP1</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

第2段階

プロジェクト名 STEP2

```
STEP2
├─src
│  ├─chapter11
│  │  │  LogoutServlet.java
│  │  │  TweetArrayBean.java
│  │  │  TweetBean.java
│  │  │  TweetDeleteServlet.java
│  │  │  TweetInputServlet.java
│  │  │  TweetListServlet.java
│  │  │  TweetPersonServlet.java
│  │  │
│  │  ├─dao
│  │  │      DAOBase.java
│  │  │      TweetDAO.java
│  │  │
│  │  ├─exception
│  │  │      DatabaseException.java
│  │  │      ParameterException.java
│  │  │      SystemException.java
│  │  │
│  │  └─parameter
│  │         DAOParameters.java
│  │         ExceptionParameters.java
│  │         TweetParameters.java
│  │         UserRoleParameters.java
│  │
│  └─filter
│        ExampleFilter.java
│
└─WebContent
    │  error.jsp
    │  login.jsp
    │  loginerror.jsp
    │  tweetinput.html
    │  tweetlist.jsp
    │  tweetpersonlist.jsp
    │
    └─WEB-INF
          web.xml
```

なお、ExampleFilter.javaは前章と同じためここでは、プログラムの掲載を省略します。

ファイル LogoutServlet.java

```
01:  package chapter11;
02:
03:  import java.io.IOException;
04:
05:  import javax.servlet.ServletException;
06:  import javax.servlet.annotation.WebServlet;
07:  import javax.servlet.http.HttpServlet;
08:  import javax.servlet.http.HttpServletRequest;
09:  import javax.servlet.http.HttpServletResponse;
10:  import javax.servlet.http.HttpSession;
11:
12:  @WebServlet("/LogoutServlet")
13:  public class LogoutServlet extends HttpServlet {
14:
15:    @Override
16:    protected void doGet(HttpServletRequest request, HttpServletResponse response)
17:        throws ServletException, IOException {
18:      HttpSession session = request.getSession(false);
19:      if (session != null) {
20:        session.invalidate();
21:        response.sendRedirect("TweetListServlet");
22:      }
23:    }
24:  }
```

ファイル TweetArrayBean.java

```
01:  package chapter11;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class TweetArrayBean implements Serializable {
07:
08:    private ArrayList<TweetBean> tweetArray;
09:
10:    public TweetArrayBean() {
11:      tweetArray = new ArrayList<TweetBean>();
12:    }
13:
14:    public void addTsubuyaki(TweetBean obj) {
15:      tweetArray.add(obj);
16:    }
17:
18:    public int getArraySize() {
19:      return tweetArray.size();
20:    }
21:
```

```
22:     public ArrayList<TweetBean> getTweetArray() {
23:       return tweetArray;
24:     }
25:
26:     public void setTweetArray(ArrayList<TweetBean> tweetArray) {
27:       this.tweetArray = tweetArray;
28:     }
29:
30:   }
```

ファイル TweetBean.java

```
01:   package chapter11;
02:
03:   import java.io.Serializable;
04:   import java.util.Date;
05:
06:   public class TweetBean implements Serializable {
07:
08:     private String name;
09:     private String tweet;
10:     private Date datetime;
11:
12:     public TweetBean() {
13:     }
14:
15:     public String getName() {
16:       return name;
17:     }
18:
19:     public void setName(String name) {
20:       this.name = name;
21:     }
22:
23:     public String getTweet() {
24:       return tweet;
25:     }
26:
27:     public void setTweet(String tweet) {
28:       this.tweet = tweet;
29:     }
30:
31:     public Date getDatetime() {
32:       return datetime;
33:     }
34:
35:     public void setDatetime(Date datetime) {
36:       this.datetime = datetime;
37:     }
38:
```

```
39:     }
```

ファイル TweetDeleteServlet.java

```
01:     package chapter11;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.TweetDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:
16:     @WebServlet("/TweetDeleteServlet")
17:     public class TweetDeleteServlet extends HttpServlet {
18:
19:       @Override
20:       protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:           throws ServletException, IOException {
22:         try {
23:           String name = request.getParameter("name");
24:
25:           TweetDAO dao = new TweetDAO();
26:           dao.deleteTweet(name);
27:
28:           response.sendRedirect("TweetListServlet");
29:         } catch (SystemException e) {
30:           e.printStackTrace();
31:           HttpSession session = request.getSession();
32:           session.setAttribute("Except", e);
33:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
34:               response);
35:         } catch (DatabaseException e) {
36:           e.printStackTrace();
37:           HttpSession session = request.getSession();
38:           session.setAttribute("Except", e);
39:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
40:               response);
41:         }
42:       }
43:
44:     }
```

ファイル TweetInputServlet.java

```
01:  package chapter11;
02:
03:  import java.io.IOException;
04:  import java.util.Date;
05:
06:  import javax.servlet.ServletException;
07:  import javax.servlet.annotation.WebServlet;
08:  import javax.servlet.http.HttpServlet;
09:  import javax.servlet.http.HttpServletRequest;
10:  import javax.servlet.http.HttpServletResponse;
11:  import javax.servlet.http.HttpSession;
12:
13:  import chapter11.dao.TweetDAO;
14:  import chapter11.exception.DatabaseException;
15:  import chapter11.exception.SystemException;
16:
17:  @WebServlet("/TweetInputServlet")
18:  public class TweetInputServlet extends HttpServlet {
19:
20:    @Override
21:    protected void doPost(HttpServletRequest request, HttpServletResponse response)
22:        throws ServletException, IOException {
23:      try {
24:        String name = request.getParameter("name");
25:        String tsubuyaki = request.getParameter("tweet");
26:        Date date = new Date();
27:
28:        TweetBean tweetBean = new TweetBean();
29:        tweetBean.setName(name);
30:        tweetBean.setTweet(tsubuyaki);
31:        tweetBean.setDatetime(date);
32:
33:        TweetDAO dao = new TweetDAO();
34:        dao.setTweet(tweetBean);
35:
36:        response.sendRedirect("TweetListServlet");
37:      } catch (SystemException e) {
38:        e.printStackTrace();
39:        HttpSession session = request.getSession();
40:        session.setAttribute("Except", e);
41:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
42:            response);
43:      } catch (DatabaseException e) {
44:        e.printStackTrace();
45:        HttpSession session = request.getSession();
46:        session.setAttribute("Except", e);
47:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
48:            response);
49:      }
```

```
  }

}
```

ファイル TweetListServlet.java

```
package chapter11;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import chapter11.dao.TweetDAO;
import chapter11.exception.DatabaseException;
import chapter11.exception.SystemException;

@WebServlet("/TweetListServlet")
public class TweetListServlet extends HttpServlet {

  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {

    TweetArrayBean tweetArray;

    try {
      TweetDAO dao = new TweetDAO();
      tweetArray = dao.getTsubuyakiArray();
      HttpSession session = request.getSession();
      session.setAttribute("TweetArrayBean", tweetArray);
      getServletContext().getRequestDispatcher("/tweetlist.jsp").forward(
          request, response);
    } catch (SystemException e) {
      e.printStackTrace();
      HttpSession session = request.getSession();
      session.setAttribute("Except", e);
      getServletContext().getRequestDispatcher("/error.jsp").forward(request,
          response);
    } catch (DatabaseException e) {
      e.printStackTrace();
      HttpSession session = request.getSession();
      session.setAttribute("Except", e);
      getServletContext().getRequestDispatcher("/error.jsp").forward(request,
          response);
    }
```

```
45:       }
46:     }
```

ファイル TweetPersonServlet.java

```
01:     package chapter11;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.TweetDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:
16:     @WebServlet("/TweetPersonServlet")
17:     public class TweetPersonServlet extends HttpServlet {
18:
19:       @Override
20:       protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:           throws ServletException, IOException {
22:
23:         TweetArrayBean tweetArray;
24:
25:         try {
26:           String name = request.getParameter("name");
27:
28:           TweetDAO dao = new TweetDAO();
29:           tweetArray = dao.getPersonTweetArray(name);
30:           HttpSession session = request.getSession();
31:           session.setAttribute("TweetArrayBean", tweetArray);
32:           getServletContext().getRequestDispatcher("/tweetpersonlist.jsp")
33:               .forward(request, response);
34:         } catch (SystemException e) {
35:           e.printStackTrace();
36:           HttpSession session = request.getSession();
37:           session.setAttribute("Except", e);
38:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
39:               response);
40:         } catch (DatabaseException e) {
41:           e.printStackTrace();
42:           HttpSession session = request.getSession();
43:           session.setAttribute("Except", e);
44:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
45:               response);
```

```
46:       }
47:
48:     }
49:   }
```

ファイル DAOBase.java

```
01:   package chapter11.dao;
02:
03:   import java.sql.Connection;
04:   import java.sql.DriverManager;
05:   import java.sql.SQLException;
06:   import java.sql.Statement;
07:
08:   import chapter11.exception.DatabaseException;
09:   import chapter11.exception.SystemException;
10:   import chapter11.parameter.DAOParameters;
11:   import chapter11.parameter.ExceptionParameters;
12:
13:   public class DAOBase {
14:
15:     Connection con;
16:
17:     protected void open() throws DatabaseException, SystemException {
18:       try {
19:         Class.forName(DAOParameters.DRIVER_NAME);
20:         con = DriverManager.getConnection(DAOParameters.CONNECT_STRING,
21:             DAOParameters.USERID, DAOParameters.PASSWORD);
22:       } catch (ClassNotFoundException e) {
23:         throw new SystemException(ExceptionParameters.SYSTEM_EXCEPTION_MESSAGE, e);
24:       } catch (SQLException e) {
25:         throw new DatabaseException(
26:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
27:       }
28:     }
29:
30:     protected void close(Statement stmt) throws DatabaseException {
31:       try {
32:         if (stmt != null) {
33:           stmt.close();
34:         }
35:         if (con != null) {
36:           con.close();
37:         }
38:       } catch (SQLException e) {
39:         throw new DatabaseException(
40:             ExceptionParameters.DATABASE_CLOSE_EXCEPTION_MESSAGE, e);
41:       }
42:     }
43:
```

```
44:     }
```

ファイル TweetDAO.java

```
01:   package chapter11.dao;
02:
03:   import java.sql.PreparedStatement;
04:   import java.sql.ResultSet;
05:   import java.sql.SQLException;
06:   import java.sql.Statement;
07:   import java.text.SimpleDateFormat;
08:
09:   import chapter11.TweetArrayBean;
10:   import chapter11.TweetBean;
11:   import chapter11.exception.DatabaseException;
12:   import chapter11.exception.SystemException;
13:   import chapter11.parameter.ExceptionParameters;
14:   import chapter11.parameter.TweetParameters;
15:
16:   public class TweetDAO extends DAOBase {
17:
18:     public TweetArrayBean getTsubuyakiArray() throws DatabaseException,
19:         SystemException {
20:       Statement stmt = null;
21:       TweetArrayBean tweetArray = new TweetArrayBean();
22:       this.open();
23:       try {
24:         stmt = con.createStatement();
25:         ResultSet rs = stmt.executeQuery(TweetParameters.SQL_SELECT_ALL);
26:         while (rs.next()) {
27:           TweetBean record = new TweetBean();
28:           record.setName(rs.getString(TweetParameters.NAME));
29:           record.setTweet(rs.getString(TweetParameters.TWEET));
30:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
31:           tweetArray.addTsubuyaki(record);
32:         }
33:       } catch (SQLException e) {
34:         throw new DatabaseException(
35:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
36:       } finally {
37:         this.close(stmt);
38:       }
39:       return tweetArray;
40:     }
41:
42:     public TweetArrayBean getPersonTweetArray(String name)
43:         throws DatabaseException, SystemException {
44:       PreparedStatement stmt = null;
45:       TweetArrayBean tweetArray = new TweetArrayBean();
46:       this.open();
```

```
47:       try {
48:         stmt = con.prepareStatement(TweetParameters.SQL_SELECT_PERSON);
49:         stmt.setString(1, name);
50:         ResultSet rs = stmt.executeQuery();
51:         while (rs.next()) {
52:           TweetBean record = new TweetBean();
53:           record.setName(rs.getString(TweetParameters.NAME));
54:           record.setTweet(rs.getString(TweetParameters.TWEET));
55:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
56:           tweetArray.addTsubuyaki(record);
57:         }
58:       } catch (SQLException e) {
59:         throw new DatabaseException(
60:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
61:       } finally {
62:         this.close(stmt);
63:       }
64:       return tweetArray;
65:     }
66:
67:     public void setTweet(TweetBean tweet) throws DatabaseException,
68:         SystemException {
69:       PreparedStatement stmt = null;
70:       this.open();
71:       try {
72:         stmt = con.prepareStatement(TweetParameters.SQL_INSERT);
73:         stmt.setString(1, tweet.getName());
74:         stmt.setString(2, tweet.getTweet());
75:         stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
76:             .format(tweet.getDatetime()));
77:         stmt.executeUpdate();
78:       } catch (SQLException e) {
79:         throw new DatabaseException(
80:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
81:       } finally {
82:         this.close(stmt);
83:       }
84:     }
85:
86:     public void deleteTweet(String name) throws DatabaseException,
87:         SystemException {
88:       PreparedStatement stmt = null;
89:       this.open();
90:       try {
91:         stmt = con.prepareStatement(TweetParameters.SQL_DELETE);
92:         stmt.setString(1, name);
93:         stmt.executeUpdate();
94:       } catch (SQLException e) {
95:         throw new DatabaseException(
96:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
```

```
97:       } finally {
98:         this.close(stmt);
99:       }
100:    }
101:  }
```

ファイル DatabaseException.java

```
01:   package chapter11.exception;
02:
03:   public class DatabaseException extends Exception {
04:
05:     public DatabaseException(String message, Throwable cause,
06:         boolean enableSuppression, boolean writableStackTrace) {
07:       super(message, cause, enableSuppression, writableStackTrace);
08:     }
09:
10:     public DatabaseException(String message, Throwable cause) {
11:       super(message, cause);
12:     }
13:
14:     public DatabaseException(String message) {
15:       super(message);
16:     }
17:
18:     public DatabaseException(Throwable cause) {
19:       super(cause);
20:     }
21:
22:   }
```

ファイル ParameterException.java

```
01:   package chapter11.exception;
02:
03:   public class ParameterException extends Exception {
04:
05:     public ParameterException(String message) {
06:       super(message);
07:     }
08:
09:     public ParameterException(Throwable cause) {
10:       super(cause);
11:     }
12:
13:     public ParameterException(String message, Throwable cause) {
14:       super(message, cause);
15:     }
16:
17:     public ParameterException(String message, Throwable cause,
```

```
18:         boolean enableSuppression, boolean writableStackTrace) {
19:       super(message, cause, enableSuppression, writableStackTrace);
20:     }
21:
22:   }
```

ファイル SystemException.java

```
01:   package chapter11.exception;
02:
03:   public class SystemException extends Exception {
04:
05:     public SystemException(String message, Throwable cause,
06:         boolean enableSuppression, boolean writableStackTrace) {
07:       super(message, cause, enableSuppression, writableStackTrace);
08:     }
09:
10:     public SystemException(String message, Throwable cause) {
11:       super(message, cause);
12:     }
13:
14:     public SystemException(String message) {
15:       super(message);
16:     }
17:
18:     public SystemException(Throwable cause) {
19:       super(cause);
20:     }
21:
22:   }
```

ファイル DAOParameters.java

```
01:   package chapter11.parameter;
02:
03:   public class DAOParameters {
04:     public static final String DRIVER_NAME = "com.mysql.jdbc.Driver";
05:     public static final String CONNECT_STRING =
06:   "jdbc:mysql://localhost:3306/tweet?serverTimezone=JST";
07:     public static final String USERID = "user";
08:     public static final String PASSWORD = "password";
09:   }
```

ファイル ExceptionParameters.java

```
01:   package chapter11.parameter;
02:
03:   public class ExceptionParameters {
04:     public static final String SYSTEM_EXCEPTION_MESSAGE = "システムエラーが発生しました";
05:
```

```
06:       public static final String DATABASE_CONNECTION_EXCEPTION_MESSAGE = "データベースへの接続時
07:     にエラーが発生しました";
08:       public static final String DATABASE_CLOSE_EXCEPTION_MESSAGE = "データベースからの切断時にエ
09:     ラーが発生しました";
10:
11:       public static final String Parameter_FORMAT_EXCEPTION_MESSAGE = "入力したデータの形式が正
12:     しくありません";
13:     }
```

ファイル TweetParameters.java

```
01:  package chapter11.parameter;
02:
03:  public class TweetParameters {
04:    public static final String SQL_SELECT_ALL = "SELECT * FROM TWEET ORDER BY DATE DESC";
05:    public static final String SQL_SELECT_PERSON = "SELECT * FROM TWEET WHERE NAME = ?
06:  ORDER BY DATE DESC";
07:
08:    public static final String SQL_INSERT = "INSERT INTO TWEET VALUES(?,?,?)";
09:
10:    public static final String SQL_DELETE = "DELETE FROM TWEET WHERE NAME=?";
11:
12:    public static final String NAME = "NAME";
13:    public static final String TWEET = "TWEET";
14:    public static final String DATE = "DATE";
15:  }
16:
```

ファイル UserRoleParameters.java

```
01:  package chapter11.parameter;
02:
03:  public class UserRoleParameters {
04:    public static final String ROLE_ADMINS = "admins";
05:    public static final String ROLE_USERS = "users";
06:  }
```

ファイル error.jsp

```
01:  <%@ page language="java" contentType="text/html; charset=UTF-8"
02:   pageEncoding="UTF-8"%>
03:  <!DOCTYPE html>
04:  <html>
05:  <head>
06:  <meta charset="UTF-8">
07:  <title>エラー</title>
08:  </head>
09:  <body>
10:   <h1>エラー画面</h1>
11:   <%
12:     Exception e = (Exception) session.getAttribute("Except");
13:   %>
14:   <p><%=e.getMessage()%>
15:   <p>
16:    <a href="javascript:history.back();">戻る</a>
17:  </body>
18:  </html>
```

ファイル login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやき ログイン</title>
</head>
<body>
 <h1>つぶやき</h1>
 <h2>ログイン</h2>
 <form action="j_security_check" method="post">
  ログイン ID：<input type="text" name="j_username"><br>
  パスワード：<input type="password" name="j_password"><br>
 <input type="submit" value="LOGIN"> <input type="reset">
 </form>
</body>
</html>
```

ファイル loginerror.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやき ログインエラー</title>
</head>
<body>
 <h1>つぶやき</h1>
 <h2>ログインエラー</h2>
 ログイン ID またはパスワードに誤りがあります。
 <br> 再度入力してください。
 <br>
 <a href=" javascript:history.back();">戻る</a>
</body>
</html>
```

ファイル tweetinput.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
```

```
<body>
 <h1>つぶやきアプリ</h1>
 <form action="TweetInputServlet" method="post">
  <p>
   名前:<br> <input type="text" name="name" maxlength="50">
  </p>
  <p>
   つぶやき:<br>
   <textarea rows="5" cols="40" name="tweet"></textarea>
  </p>
  <p>
   <input type="submit" value="送信"> <input type="reset">
   <input type="button" onclick="location.href='TweetListServlet'"
    value="戻る">
 </form>
</html>
```

ファイル tweetlist.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="java.text.SimpleDateFormat"%>
<%@ page import="chapter11.*"%>
<%@ page import="java.net.URLEncoder"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="TweetArrayBean"
  class="chapter11.TweetArrayBean" scope="session" />
 <h1>つぶやきアプリ</h1>
 <%=request.getRemoteUser()%>：
 <a href="LogoutServlet">ログアウト</a>
 <br>
  <br>
 <form action="tweetinput.html" method="get">
  <input type="submit" value="書き込み">
 </form>
 <br>
 <table border="1">
  <tr>
   <th>名前</th>
   <th>つぶやき</th>
   <th>日時</th>
  </tr>
  <%
```

```
32:     ArrayList<TweetBean> tweetArray = TweetArrayBean
33:       .getTweetArray();
34:       for (TweetBean record : tweetArray) {
35:   %>
36:   <tr>
37:    <td>
38:        <a
39:  href="TweetPersonServlet?name=<%=URLEncoder.encode(record.getName(),"UTF-8")%>"><%=
40:  record.getName()%></a>
41:    </td>
42:    <td><%=record.getTweet()%></td>
43:    <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
44:             .getDatetime())%></td>
45:   </tr>
46:   <%
47:     }
48:   %>
49:  </table>
50: </body>
</html>
```

ファイル tweetpersonlist.jsp

```
01: <%@ page language="java" contentType="text/html; charset=UTF-8"
02:  pageEncoding="UTF-8"%>
03: <%@ page import="java.util.ArrayList"%>
04: <%@ page import="java.text.SimpleDateFormat"%>
05: <%@ page import="chapter11.*"%>
06: <%@ page import="chapter11.parameter.*"%>
07: <!DOCTYPE html>
08: <html>
09: <head>
10: <meta charset="UTF-8">
11: <title>つぶやきアプリ</title>
12: </head>
13: <body>
14:  <jsp:useBean id="TweetArrayBean"
15:   class="chapter11.TweetArrayBean" scope="session" />
16:  <h1>つぶやきアプリ</h1>
17:  <h2><%=request.getParameter("name")%>さんのつぶやき
18:  </h2>
19:  <table>
20:   <tr>
21:    <td><form action="TweetListServlet" method="get">
22:      <input type="submit" value="戻る">
23:     </form></td>
24:    <%
25:      if (request.isUserInRole(UserRoleParameters.ROLE_ADMINS)) {
26:    %>
27:    <td><form action="TweetDeleteServlet" method="get">
```

```
28:         <input type="hidden" name="name"
29:          value="<%=request.getParameter("name")%>"> <input
30:          type="submit" value="削除">
31:        </form></td>
32:       <%
33:         }
34:       %>
35:      </tr>
36:     </table>
37:     <br>
38:     <table border="1">
39:      <tr>
40:       <th>名前</th>
41:       <th>つぶやき</th>
42:       <th>日時</th>
43:      </tr>
44:      <%
45:        ArrayList<TweetBean> tweetArray = TweetArrayBean
46:          .getTweetArray();
47:          for (TweetBean record : tweetArray) {
48:      %>
49:      <tr>
50:       <td><%=record.getName()%></td>
51:       <td><%=record.getTweet()%></td>
52:       <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
53:               .getDatetime())%></td>
54:      </tr>
55:      <%
56:        }
57:      %>
58:     </table>
59:    </body>
60:    </html>
```

ファイル web.xml

```
01:    <?xml version="1.0" encoding="UTF-8"?>
02:    <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
03:    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
04:    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
05:    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
06:      <display-name>STEP2</display-name>
07:      <welcome-file-list>
08:        <welcome-file>index.html</welcome-file>
09:        <welcome-file>index.htm</welcome-file>
10:        <welcome-file>index.jsp</welcome-file>
11:        <welcome-file>default.html</welcome-file>
12:        <welcome-file>default.htm</welcome-file>
13:        <welcome-file>default.jsp</welcome-file>
14:      </welcome-file-list>
```

```
15:
16:     <security-constraint>
17:     <web-resource-collection>
18:       <web-resource-name>FORM 認証</web-resource-name>
19:       <url-pattern>/*</url-pattern>
20:     </web-resource-collection>
21:     <auth-constraint>
22:       <role-name>admins</role-name>
23:     </auth-constraint>
24:   </security-constraint>
25:   <security-constraint>
26:     <web-resource-collection>
27:       <web-resource-name>FORM 認証</web-resource-name>
28:       <url-pattern>/TweetListServlet</url-pattern>
29:       <url-pattern>/TweetPersonServlet</url-pattern>
30:       <url-pattern>/TweetInputServlet</url-pattern>
31:       <url-pattern>/LogoutServlet</url-pattern>
32:       <url-pattern>/tweetinput.html</url-pattern>
33:       <url-pattern>/tweetlist.jsp</url-pattern>
34:       <url-pattern>/tweetpersonlist.jsp</url-pattern>
35:       <url-pattern>/error.jsp</url-pattern>
36:     </web-resource-collection>
37:     <auth-constraint>
38:       <role-name>admins</role-name>
39:       <role-name>users</role-name>
40:     </auth-constraint>
41:   </security-constraint>
42:   <login-config>
43:     <auth-method>FORM</auth-method>
44:     <form-login-config>
45:       <form-login-page>/login.jsp</form-login-page>
46:       <form-error-page>/loginerror.jsp</form-error-page>
47:     </form-login-config>
48:   </login-config>
49:   <security-role>
50:     <role-name>users</role-name>
51:   </security-role>
52:   <security-role>
53:     <role-name>admins</role-name>
54:   </security-role>
55:
56: </web-app>
```

第3段階

プロジェクト名 STEP3

```
STEP3
├─src
│  ├─chapter11
│  │  │  LogoutServlet.java
│  │  │
│  │  ├─dao
│  │  │      DAOBase.java
│  │  │      TweetDAO.java
│  │  │      UserDAO.java
│  │  │
│  │  ├─exception
│  │  │      DatabaseException.java
│  │  │      ParameterException.java
│  │  │      SystemException.java
│  │  │
│  │  ├─parameter
│  │  │      DAOParameters.java
│  │  │      ExceptionParameters.java
│  │  │      TweetParameters.java
│  │  │      UserRoleParameters.java
│  │  │
│  │  ├─Tweet
│  │  │      TweetArrayBean.java
│  │  │      TweetBean.java
│  │  │      TweetDeleteServlet.java
│  │  │      TweetInputServlet.java
│  │  │      TweetListServlet.java
│  │  │      TweetPersonServlet.java
│  │  │
│  │  └─user
│  │          UserArrayBean.java
│  │          UserBean.java
│  │          UserDeleteServlet.java
│  │          UserInputServlet.java
│  │          UserListServlet.java
│  │          UserPersonServlet.java
│  │          UserUpdateServlet.java
│  │
│  └─filter
│          ExampleFilter.java
│
└─WebContent
    │  error.jsp
    │  login.jsp
    │  loginerror.jsp
    │  tweetinput.html
```

```
    |  tweetlist.jsp
    |  tweetpersonlist.jsp
    |  userinput.html
    |  userlist.jsp
    |  userperson.jsp
    |
    └──📂WEB-INF
           web.xml
```

なお、ExampleFilter.javaは前章と同じためここでは、プログラムの掲載を省略します。

ファイル LogoutServlet.java

```
01:  package chapter11;
02:
03:  import java.io.IOException;
04:
05:  import javax.servlet.ServletException;
06:  import javax.servlet.annotation.WebServlet;
07:  import javax.servlet.http.HttpServlet;
08:  import javax.servlet.http.HttpServletRequest;
09:  import javax.servlet.http.HttpServletResponse;
10:  import javax.servlet.http.HttpSession;
11:
12:  @WebServlet("/LogoutServlet")
13:  public class LogoutServlet extends HttpServlet {
14:
15:    @Override
16:    protected void doGet(HttpServletRequest request, HttpServletResponse response)
17:        throws ServletException, IOException {
18:      HttpSession session = request.getSession(false);
19:      if (session != null) {
20:        session.invalidate();
21:        response.sendRedirect("TweetListServlet");
22:      }
23:    }
24:  }
```

ファイル DAOBase.java

```
01:  package chapter11.dao;
02:
03:  import java.sql.Connection;
04:  import java.sql.DriverManager;
05:  import java.sql.SQLException;
06:  import java.sql.Statement;
07:
08:  import chapter11.exception.DatabaseException;
09:  import chapter11.exception.SystemException;
10:  import chapter11.parameter.DAOParameters;
11:  import chapter11.parameter.ExceptionParameters;
12:
13:  public class DAOBase {
14:
15:    Connection con;
16:
17:    protected void open() throws DatabaseException, SystemException {
18:      try {
19:        Class.forName(DAOParameters.DRIVER_NAME);
20:        con = DriverManager.getConnection(DAOParameters.CONNECT_STRING,
21:            DAOParameters.USERID, DAOParameters.PASSWORD);
```

```
    } catch (ClassNotFoundException e) {
      throw new SystemException(ExceptionParameters.SYSTEM_EXCEPTION_MESSAGE, e);
    } catch (SQLException e) {
      throw new DatabaseException(
          ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
    }
  }

  protected void close(Statement stmt) throws DatabaseException {
    try {
      if (stmt != null) {
        stmt.close();
      }
      if (con != null) {
        con.close();
      }
    } catch (SQLException e) {
      throw new DatabaseException(
          ExceptionParameters.DATABASE_CLOSE_EXCEPTION_MESSAGE, e);
    }
  }

}
```

ファイル TweetDAO.java

```
package chapter11.dao;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;

import chapter11.exception.DatabaseException;
import chapter11.exception.SystemException;
import chapter11.parameter.ExceptionParameters;
import chapter11.parameter.TweetParameters;
import chapter11.tweet.TweetArrayBean;
import chapter11.tweet.TweetBean;

public class TweetDAO extends DAOBase {

  public TweetArrayBean getTweetArray() throws DatabaseException,
      SystemException {
    Statement stmt = null;
    TweetArrayBean tweetArray = new TweetArrayBean();
    this.open();
    try {
      stmt = con.createStatement();
```

```
25:         ResultSet rs = stmt.executeQuery(TweetParameters.SQL_SELECT_ALL);
26:         while (rs.next()) {
27:           TweetBean record = new TweetBean();
28:           record.setName(rs.getString(TweetParameters.NAME));
29:           record.setTweet(rs.getString(TweetParameters.TWEET));
30:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
31:           tweetArray.addTweet(record);
32:         }
33:       } catch (SQLException e) {
34:         throw new DatabaseException(
35:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
36:       } finally {
37:         this.close(stmt);
38:       }
39:       return tweetArray;
40:     }
41:
42:     public TweetArrayBean getPersonTweetArray(String name)
43:         throws DatabaseException, SystemException {
44:       PreparedStatement stmt = null;
45:       TweetArrayBean tweetArray = new TweetArrayBean();
46:       this.open();
47:       try {
48:         stmt = con.prepareStatement(TweetParameters.SQL_SELECT_PERSON);
49:         stmt.setString(1, name);
50:         ResultSet rs = stmt.executeQuery();
51:         while (rs.next()) {
52:           TweetBean record = new TweetBean();
53:           record.setName(rs.getString(TweetParameters.NAME));
54:           record.setTweet(rs.getString(TweetParameters.TWEET));
55:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
56:           tweetArray.addTweet(record);
57:         }
58:       } catch (SQLException e) {
59:         throw new DatabaseException(
60:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
61:       } finally {
62:         this.close(stmt);
63:       }
64:       return tweetArray;
65:     }
66:
67:     public void setTweet(TweetBean tweet) throws DatabaseException,
68:         SystemException {
69:       PreparedStatement stmt = null;
70:       this.open();
71:       try {
72:         stmt = con.prepareStatement(TweetParameters.SQL_INSERT);
73:         stmt.setString(1, tweet.getName());
74:         stmt.setString(2, tweet.getTweet());
```

```
75:         stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
76:             .format(tweet.getDatetime()));
77:         stmt.executeUpdate();
78:       } catch (SQLException e) {
79:         throw new DatabaseException(
80:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
81:       } finally {
82:         this.close(stmt);
83:       }
84:     }
85:
86:     public void deleteTweet(String name) throws DatabaseException,
87:         SystemException {
88:       PreparedStatement stmt = null;
89:       this.open();
90:       try {
91:         stmt = con.prepareStatement(TweetParameters.SQL_DELETE);
92:         stmt.setString(1, name);
93:         stmt.executeUpdate();
94:       } catch (SQLException e) {
95:         throw new DatabaseException(
96:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
97:       } finally {
98:         this.close(stmt);
99:       }
100:    }
101:  }
```

ファイル UserDAO.java

```
01:  package chapter11.dao;
02:
03:  import java.sql.PreparedStatement;
04:  import java.sql.ResultSet;
05:  import java.sql.SQLException;
06:  import java.sql.Statement;
07:
08:  import chapter11.exception.DatabaseException;
09:  import chapter11.exception.SystemException;
10:  import chapter11.parameter.ExceptionParameters;
11:  import chapter11.parameter.UserRoleParameters;
12:  import chapter11.user.UserArrayBean;
13:  import chapter11.user.UserBean;
14:
15:  public class UserDAO extends DAOBase {
16:
17:    public UserArrayBean getUserArray() throws DatabaseException,
18:        SystemException {
19:      Statement stmt = null;
20:      UserArrayBean userArray = new UserArrayBean();
21:      this.open();
22:      try {
23:        stmt = con.createStatement();
24:        ResultSet rs = stmt
25:            .executeQuery(UserRoleParameters.SQL_SELECT_ALL);
26:        while (rs.next()) {
27:          UserBean record = new UserBean();
28:          record.setUserid(rs.getString(UserRoleParameters.USERID));
29:          record.setRole(rs.getString(UserRoleParameters.ROLE));
30:          userArray.addUser(record);
31:        }
32:      } catch (SQLException e) {
33:        throw new DatabaseException(
34:            ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
35:      } finally {
36:        this.close(stmt);
37:      }
38:      return userArray;
39:    }
40:
41:    public UserBean getUser(String userid) throws DatabaseException,
42:        SystemException {
43:      UserBean record = null;
44:      PreparedStatement stmt = null;
45:      this.open();
46:      try {
47:        stmt = con.prepareStatement(UserRoleParameters.SQL_SELECT_PERSON);
48:        stmt.setString(1, userid);
49:        ResultSet rs = stmt.executeQuery();
```

```
50:         rs.next();
51:         record = new UserBean();
52:         record.setUserid(rs.getString(UserRoleParameters.USERID));
53:         record.setPasswd(rs.getString(UserRoleParameters.PASSWORD));
54:         record.setRole(rs.getString(UserRoleParameters.ROLE));
55:       } catch (SQLException e) {
56:         throw new DatabaseException(
57:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
58:       } finally {
59:         this.close(stmt);
60:       }
61:       return record;
62:     }
63:
64:     public void addUser(UserBean user) throws DatabaseException,
65:         SystemException {
66:
67:       PreparedStatement stmt = null;
68:       this.open();
69:       try {
70:         stmt = con
71:             .prepareStatement(UserRoleParameters.SQL_INSERT_USERTABLE);
72:         stmt.setString(1, user.getUserid());
73:         stmt.setString(2, user.getPasswd());
74:         stmt.executeUpdate();
75:       } catch (SQLException e) {
76:         throw new DatabaseException(
77:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
78:       } finally {
79:         this.close(stmt);
80:       }
81:       this.open();
82:       try {
83:         stmt = con
84:             .prepareStatement(UserRoleParameters.SQL_INSERT_USERROLETABLE);
85:         stmt.setString(1, user.getUserid());
86:         stmt.setString(2, user.getRole());
87:         stmt.executeUpdate();
88:       } catch (SQLException e) {
89:         throw new DatabaseException(
90:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
91:       } finally {
92:         this.close(stmt);
93:       }
94:
95:     }
96:
97:     public void updateUser(UserBean user) throws DatabaseException,
98:         SystemException {
99:       PreparedStatement stmt = null;
```

```
100:       this.open();
101:       try {
102:         stmt = con.prepareStatement(UserRoleParameters.SQL_UPDATE_USER);
103:         stmt.setString(1, user.getPasswd());
104:         stmt.setString(2, user.getUserid());
105:         stmt.executeUpdate();
106:       } catch (SQLException e) {
107:         throw new DatabaseException(
108:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
109:       } finally {
110:         this.close(stmt);
111:       }
112:       this.open();
113:       try {
114:         stmt = con.prepareStatement(UserRoleParameters.SQL_UPDATE_ROLE);
115:         stmt.setString(1, user.getRole());
116:         stmt.setString(2, user.getUserid());
117:         stmt.executeUpdate();
118:       } catch (SQLException e) {
119:         throw new DatabaseException(
120:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
121:       } finally {
122:         this.close(stmt);
123:       }
124:
125:     }
126:
127:     public void deleteUser(String userid) throws DatabaseException,
128:         SystemException {
129:       PreparedStatement stmt = null;
130:       this.open();
131:       try {
132:         stmt = con.prepareStatement(UserRoleParameters.SQL_DELETE_ROLE);
133:         stmt.setString(1, userid);
134:         stmt.executeUpdate();
135:       } catch (SQLException e) {
136:         throw new DatabaseException(
137:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
138:       } finally {
139:         this.close(stmt);
140:       }
141:       this.open();
142:       try {
143:         stmt = con.prepareStatement(UserRoleParameters.SQL_DELETE_USER);
144:         stmt.setString(1, userid);
145:         stmt.executeUpdate();
146:       } catch (SQLException e) {
147:         throw new DatabaseException(
148:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
149:       } finally {
```

```
150:         this.close(stmt);
151:       }
152:
153:     }
154:
155:   }
```

ファイル DatabaseException.java

```
01:   package chapter11.exception;
02:
03:   public class DatabaseException extends Exception {
04:
05:     public DatabaseException(String message, Throwable cause,
06:         boolean enableSuppression, boolean writableStackTrace) {
07:       super(message, cause, enableSuppression, writableStackTrace);
08:     }
09:
10:     public DatabaseException(String message, Throwable cause) {
11:       super(message, cause);
12:     }
13:
14:     public DatabaseException(String message) {
15:       super(message);
16:     }
17:
18:     public DatabaseException(Throwable cause) {
19:       super(cause);
20:     }
21:
22:   }
```

ファイル ParameterException.java

```
01:   package chapter11.exception;
02:
03:   public class ParameterException extends Exception {
04:
05:     public ParameterException(String message) {
06:       super(message);
07:     }
08:
09:     public ParameterException(Throwable cause) {
10:       super(cause);
11:     }
12:
13:     public ParameterException(String message, Throwable cause) {
14:       super(message, cause);
15:     }
16:
```

```
17:       public ParameterException(String message, Throwable cause,
18:           boolean enableSuppression, boolean writableStackTrace) {
19:         super(message, cause, enableSuppression, writableStackTrace);
20:       }
21:
22:     }
```

ファイル SystemException.java

```
01:     package chapter11.exception;
02:
03:     public class SystemException extends Exception {
04:
05:       public SystemException(String message, Throwable cause,
06:           boolean enableSuppression, boolean writableStackTrace) {
07:         super(message, cause, enableSuppression, writableStackTrace);
08:       }
09:
10:       public SystemException(String message, Throwable cause) {
11:         super(message, cause);
12:       }
13:
14:       public SystemException(String message) {
15:         super(message);
16:       }
17:
18:       public SystemException(Throwable cause) {
19:         super(cause);
20:       }
21:
22:     }
```

ファイル DAOParameters.java

```
01:     package chapter11.parameter;
02:
03:     public class DAOParameters {
04:       public static final String DRIVER_NAME = "com.mysql.jdbc.Driver";
05:       public static final String CONNECT_STRING =
06:     "jdbc:mysql://localhost:3306/tweet?serverTimezone=JST";
07:       public static final String USERID = "user";
08:       public static final String PASSWORD = "password";
09:     }
```

ファイル ExceptionParameters.java

```
01:     package chapter11.parameter;
02:
03:     public class ExceptionParameters {
04:       public static final String SYSTEM_EXCEPTION_MESSAGE = "システムエラーが発生しました";
```

```
05:
06:       public static final String DATABASE_CONNECTION_EXCEPTION_MESSAGE = "データベースへの接
07:     続時にエラーが発生しました";
08:       public static final String DATABASE_CLOSE_EXCEPTION_MESSAGE = "データベースからの切断時
09:     にエラーが発生しました";
10:
11:       public static final String Parameter_FORMAT_EXCEPTION_MESSAGE = "入力したデータの形式が
12:     正しくありません";
13:     }
```

ファイル TweetParameters.java

```
01:  package chapter11.parameter;
02:
03:  public class TweetParameters {
04:    public static final String SQL_SELECT_ALL = "SELECT * FROM TWEET ORDER BY DATE DESC";
05:    public static final String SQL_SELECT_PERSON = "SELECT * FROM TWEET WHERE NAME = ?
06:  ORDER BY DATE DESC";
07:
08:    public static final String SQL_INSERT = "INSERT INTO TWEET VALUES(?,?,?)";
09:
10:    public static final String SQL_DELETE = "DELETE FROM TWEET WHERE NAME=?";
11:
12:    public static final String NAME = "NAME";
13:    public static final String TWEET = "TWEET";
14:    public static final String DATE = "DATE";
15:  }
16:
```

ファイル UserRoleParameters.java

```
01:  package chapter11.parameter;
02:
03:  public class UserRoleParameters {
04:    public static final String ROLE_ADMINS = "admins";
05:    public static final String ROLE_USERS = "users";
06:
07:    public static final String SQL_SELECT_ALL = "SELECT * FROM USER_ROLES";
08:    public static final String SQL_SELECT_PERSON = "SELECT * FROM USERS,USER_ROLES WHERE
09:  USERS.USERID = USER_ROLES.USERID AND USERS.USERID=?";
10:    public static final String SQL_INSERT_USERTABLE = "INSERT INTO USERS VALUES(?,?)";
11:    public static final String SQL_INSERT_USERROLETABLE = "INSERT INTO USER_ROLES
12:  VALUES(?,?)";
13:    public static final String SQL_UPDATE_USER = "UPDATE USERS SET PASSWORD=? WHERE
14:  USERID=?";
15:    public static final String SQL_UPDATE_ROLE = "UPDATE USER_ROLES SET ROLE=? WHERE
16:  USERID=?";
17:    public static final String SQL_DELETE_USER = "DELETE FROM USERS WHERE USERID=?";
18:    public static final String SQL_DELETE_ROLE = "DELETE FROM USER_ROLES WHERE USERID=?";
19:
20:    public static final String USERID = "userid";
21:    public static final String PASSWORD = "password";
22:    public static final String ROLE = "role";
23:
24:  }
25:
26:
```

ファイル TweetArrayBean.java

```
01:  package chapter11.tweet;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class TweetArrayBean implements Serializable {
07:
08:    private ArrayList<TweetBean> tweetArray;
09:
10:    public TweetArrayBean() {
11:      tweetArray = new ArrayList<TweetBean>();
12:    }
13:
14:    public void addTweet(TweetBean obj) {
15:      tweetArray.add(obj);
16:    }
17:
18:    public int getArraySize() {
19:      return tweetArray.size();
20:    }
21:
22:    public ArrayList<TweetBean> getTweetArray() {
23:      return tweetArray;
24:    }
25:
26:    public void setTweetArray(ArrayList<TweetBean> tweetArray) {
27:      this.tweetArray = tweetArray;
28:    }
29:
30:  }
```

ファイル TweetBean.java

```
01:  package chapter11.tweet;
02:
03:  import java.io.Serializable;
04:  import java.util.Date;
05:
06:  public class TweetBean implements Serializable {
07:
08:    private String name;
09:    private String tweet;
10:    private Date datetime;
11:
12:    public TweetBean() {
13:    }
14:
15:    public String getName() {
```

```
16:       return name;
17:     }
18:
19:     public void setName(String name) {
20:       this.name = name;
21:     }
22:
23:     public String getTweet() {
24:       return tweet;
25:     }
26:
27:     public void setTweet(String tweet) {
28:       this.tweet = tweet;
29:     }
30:
31:     public Date getDatetime() {
32:       return datetime;
33:     }
34:
35:     public void setDatetime(Date datetime) {
36:       this.datetime = datetime;
37:     }
38:
39:   }
```

ファイル TweetDeleteServlet.java

```
01:   package chapter11.tweet;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   import chapter11.dao.TweetDAO;
13:   import chapter11.exception.DatabaseException;
14:   import chapter11.exception.SystemException;
15:
16:   @WebServlet("/TweetDeleteServlet")
17:   public class TweetDeleteServlet extends HttpServlet {
18:
19:     @Override
20:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:         throws ServletException, IOException {
22:       try {
23:         String name = request.getParameter("name");
```

```
24:
25:         TweetDAO dao = new TweetDAO();
26:         dao.deleteTweet(name);
27:
28:         response.sendRedirect("TweetListServlet");
29:       } catch (SystemException e) {
30:         e.printStackTrace();
31:         HttpSession session = request.getSession();
32:         session.setAttribute("Except", e);
33:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
34:             response);
35:       } catch (DatabaseException e) {
36:         e.printStackTrace();
37:         HttpSession session = request.getSession();
38:         session.setAttribute("Except", e);
39:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
40:             response);
41:       }
42:     }
43:
44:   }
```

ファイル TweetInputServlet.java

```
01:   package chapter11.tweet;
02:
03:   import java.io.IOException;
04:   import java.util.Date;
05:
06:   import javax.servlet.ServletException;
07:   import javax.servlet.annotation.WebServlet;
08:   import javax.servlet.http.HttpServlet;
09:   import javax.servlet.http.HttpServletRequest;
10:   import javax.servlet.http.HttpServletResponse;
11:   import javax.servlet.http.HttpSession;
12:
13:   import chapter11.dao.TweetDAO;
14:   import chapter11.exception.DatabaseException;
15:   import chapter11.exception.SystemException;
16:
17:   @WebServlet("/TweetInputServlet")
18:   public class TweetInputServlet extends HttpServlet {
19:
20:     @Override
21:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
22:         throws ServletException, IOException {
23:       try {
24:         String name = request.getParameter("name");
25:         String tweet = request.getParameter("tweet");
26:         Date date = new Date();
```

```
27:
28:         TweetBean tweetBean = new TweetBean();
29:         tweetBean.setName(name);
30:         tweetBean.setTweet(tweet);
31:         tweetBean.setDatetime(date);
32:
33:         TweetDAO dao = new TweetDAO();
34:         dao.setTweet(tweetBean);
35:
36:         response.sendRedirect("TweetListServlet");
37:       } catch (SystemException e) {
38:         e.printStackTrace();
39:         HttpSession session = request.getSession();
40:         session.setAttribute("Except", e);
41:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
42:             response);
43:       } catch (DatabaseException e) {
44:         e.printStackTrace();
45:         HttpSession session = request.getSession();
46:         session.setAttribute("Except", e);
47:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
48:             response);
49:       }
50:     }
51:
52:   }
```

ファイル TweetListServlet.java

```
01:   package chapter11.tweet;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   import chapter11.dao.TweetDAO;
13:   import chapter11.exception.DatabaseException;
14:   import chapter11.exception.SystemException;
15:
16:   @WebServlet("/TweetListServlet")
17:   public class TweetListServlet extends HttpServlet {
18:
19:     @Override
20:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:         throws ServletException, IOException {
```

```
22:
23:     TweetArrayBean tweetArray;
24:
25:     try {
26:       TweetDAO dao = new TweetDAO();
27:       tweetArray = dao.getTweetArray();
28:       HttpSession session = request.getSession();
29:       session.setAttribute("TweetArrayBean", tweetArray);
30:       getServletContext().getRequestDispatcher("/tweetlist.jsp").forward(
31:           request, response);
32:     } catch (SystemException e) {
33:       e.printStackTrace();
34:       HttpSession session = request.getSession();
35:       session.setAttribute("Except", e);
36:       getServletContext().getRequestDispatcher("/error.jsp").forward(request,
37:           response);
38:     } catch (DatabaseException e) {
39:       e.printStackTrace();
40:       HttpSession session = request.getSession();
41:       session.setAttribute("Except", e);
42:       getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:           response);
44:     }
45:   }
46: }
```

ファイル TweetPersonServlet.java

```
01: package chapter11.tweet;
02:
03: import java.io.IOException;
04:
05: import javax.servlet.ServletException;
06: import javax.servlet.annotation.WebServlet;
07: import javax.servlet.http.HttpServlet;
08: import javax.servlet.http.HttpServletRequest;
09: import javax.servlet.http.HttpServletResponse;
10: import javax.servlet.http.HttpSession;
11:
12: import chapter11.dao.TweetDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TweetPersonServlet")
17: public class TweetPersonServlet extends HttpServlet {
18:   private static final long serialVersionUID = 1L;
19:
20:   @Override
21:   protected void doGet(HttpServletRequest request, HttpServletResponse response)
22:       throws ServletException, IOException {
```

```
23:
24:       TweetArrayBean tweetArray;
25:
26:       try {
27:         String name = request.getParameter("name");
28:
29:         TweetDAO dao = new TweetDAO();
30:         tweetArray = dao.getPersonTweetArray(name);
31:         HttpSession session = request.getSession();
32:         session.setAttribute("TweetArrayBean", tweetArray);
33:         getServletContext().getRequestDispatcher("/tweetpersonlist.jsp")
34:             .forward(request, response);
35:       } catch (SystemException e) {
36:         e.printStackTrace();
37:         HttpSession session = request.getSession();
38:         session.setAttribute("Except", e);
39:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
40:             response);
41:       } catch (DatabaseException e) {
42:         e.printStackTrace();
43:         HttpSession session = request.getSession();
44:         session.setAttribute("Except", e);
45:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
46:             response);
47:       }
48:
49:     }
50:   }
```

ファイル UserArrayBean.java

```
01:  package chapter11.user;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class UserArrayBean implements Serializable {
07:
08:    private ArrayList<UserBean> userArray;
09:
10:    public UserArrayBean() {
11:      userArray = new ArrayList<UserBean>();
12:    }
13:
14:    public void addUser(UserBean obj) {
15:      userArray.add(obj);
16:    }
17:
18:    public int getArraySize() {
19:      return userArray.size();
20:    }
21:
22:    public ArrayList<UserBean> getUserArray() {
23:      return userArray;
24:    }
25:
26:    public void setUserArray(ArrayList<UserBean> userArray) {
27:      this.userArray = userArray;
28:    }
29:
30:  }
```

ファイル UserBean.java

```
01:  package chapter11.user;
02:
03:  import java.io.Serializable;
04:
05:  public class UserBean implements Serializable {
06:
07:    private String userid;
08:    private String passwd;
09:    private String role;
10:
11:    public UserBean() {
12:    }
13:
14:    public String getUserid() {
15:      return userid;
```

```
  }

  public void setUserid(String userid) {
    this.userid = userid;
  }

  public String getPasswd() {
    return passwd;
  }

  public void setPasswd(String passwd) {
    this.passwd = passwd;
  }

  public String getRole() {
    return role;
  }

  public void setRole(String role) {
    this.role = role;
  }

}
```

ファイル UserDeleteServlet.java

```
package chapter11.user;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import chapter11.dao.UserDAO;
import chapter11.exception.DatabaseException;
import chapter11.exception.SystemException;

@WebServlet("/UserDeleteServlet")
public class UserDeleteServlet extends HttpServlet {

  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    try {
      String userid = request.getParameter("userid");

```

```
25:         UserDAO dao = new UserDAO();
26:         dao.deleteUser(userid);
27:
28:         response.sendRedirect("UserListServlet");
29:       } catch (SystemException e) {
30:         e.printStackTrace();
31:         HttpSession session = request.getSession();
32:         session.setAttribute("Except", e);
33:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
34:             response);
35:       } catch (DatabaseException e) {
36:         e.printStackTrace();
37:         HttpSession session = request.getSession();
38:         session.setAttribute("Except", e);
39:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
40:             response);
41:       }
42:     }
43:
44:   }
```

ファイル UserInputServlet.java

```
01:   package chapter11.user;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   import chapter11.dao.UserDAO;
13:   import chapter11.exception.DatabaseException;
14:   import chapter11.exception.SystemException;
15:
16:   @WebServlet("/UserInputServlet")
17:   public class UserInputServlet extends HttpServlet {
18:
19:     @Override
20:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
21:         throws ServletException, IOException {
22:       try {
23:         String userid = request.getParameter("userid");
24:         String passwd = request.getParameter("passwd");
25:         String role = request.getParameter("role");
26:
27:         UserBean userBean = new UserBean();
```

```
28:         userBean.setUserid(userid);
29:         userBean.setPasswd(passwd);
30:         userBean.setRole(role);
31:
32:         UserDAO dao = new UserDAO();
33:         dao.addUser(userBean);
34:
35:         response.sendRedirect("UserListServlet");
36:       } catch (SystemException e) {
37:         e.printStackTrace();
38:         HttpSession session = request.getSession();
39:         session.setAttribute("Except", e);
40:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
41:             response);
42:       } catch (DatabaseException e) {
43:         e.printStackTrace();
44:         HttpSession session = request.getSession();
45:         session.setAttribute("Except", e);
46:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:             response);
48:       }
49:     }
50:
51:   }
```

ファイル UserListServlet.java

```
01:   package chapter11.user;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   import chapter11.dao.UserDAO;
13:   import chapter11.exception.DatabaseException;
14:   import chapter11.exception.SystemException;
15:
16:   @WebServlet("/UserListServlet")
17:   public class UserListServlet extends HttpServlet {
18:
19:     @Override
20:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:         throws ServletException, IOException {
22:
23:       UserArrayBean userArray;
```

```
24:
25:         try {
26:           UserDAO dao = new UserDAO();
27:           userArray = dao.getUserArray();
28:           HttpSession session = request.getSession();
29:           session.setAttribute("UserArrayBean", userArray);
30:           getServletContext().getRequestDispatcher("/userlist.jsp").forward(
31:               request, response);
32:         } catch (SystemException e) {
33:           e.printStackTrace();
34:           HttpSession session = request.getSession();
35:           session.setAttribute("Except", e);
36:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
37:               response);
38:         } catch (DatabaseException e) {
39:           e.printStackTrace();
40:           HttpSession session = request.getSession();
41:           session.setAttribute("Except", e);
42:           getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:               response);
44:         }
45:       }
46:     }
```

ファイル UserPersonServlet.java

```
01:     package chapter11.user;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.UserDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:
16:     @WebServlet("/UserPersonServlet")
17:     public class UserPersonServlet extends HttpServlet {
18:
19:       @Override
20:       protected void doGet(HttpServletRequest request, HttpServletResponse response)
21:           throws ServletException, IOException {
22:
23:         UserBean user = null;
24:
```

```
25:       try {
26:         String userid = request.getParameter("userid");
27:
28:         UserDAO dao = new UserDAO();
29:         user = dao.getUser(userid);
30:         HttpSession session = request.getSession();
31:         session.setAttribute("UserBean", user);
32:         getServletContext().getRequestDispatcher("/userperson.jsp").forward(
33:             request, response);
34:       } catch (SystemException e) {
35:         e.printStackTrace();
36:         HttpSession session = request.getSession();
37:         session.setAttribute("Except", e);
38:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
39:             response);
40:       } catch (DatabaseException e) {
41:         e.printStackTrace();
42:         HttpSession session = request.getSession();
43:         session.setAttribute("Except", e);
44:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
45:             response);
46:       }
47:
48:     }
49:   }
```

ファイル UserUpdateServlet.java

```
01:  package chapter11.user;
02:
03:  import java.io.IOException;
04:
05:  import javax.servlet.ServletException;
06:  import javax.servlet.annotation.WebServlet;
07:  import javax.servlet.http.HttpServlet;
08:  import javax.servlet.http.HttpServletRequest;
09:  import javax.servlet.http.HttpServletResponse;
10:  import javax.servlet.http.HttpSession;
11:
12:  import chapter11.dao.UserDAO;
13:  import chapter11.exception.DatabaseException;
14:  import chapter11.exception.SystemException;
15:
16:  @WebServlet("/UserUpdateServlet")
17:  public class UserUpdateServlet extends HttpServlet {
18:
19:    @Override
20:    protected void doPost(HttpServletRequest request, HttpServletResponse response)
21:        throws ServletException, IOException {
22:
23:      UserBean user = null;
24:
25:      try {
26:        String userid = request.getParameter("userid");
27:        String passwd = request.getParameter("passwd");
28:        String role = request.getParameter("role");
29:
30:        user = new UserBean();
31:        user.setUserid(userid);
32:        user.setPasswd(passwd);
33:        user.setRole(role);
34:
35:        UserDAO dao = new UserDAO();
36:        dao.updateUser(user);
37:
38:        response.sendRedirect("UserListServlet");
39:      } catch (SystemException e) {
40:        e.printStackTrace();
41:        HttpSession session = request.getSession();
42:        session.setAttribute("Except", e);
43:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
44:            response);
45:      } catch (DatabaseException e) {
46:        e.printStackTrace();
47:        HttpSession session = request.getSession();
48:        session.setAttribute("Except", e);
49:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
```

```
50:             response);
51:        }
52:
53:      }
54:    }
```

ファイル error.jsp

```
01:    <%@ page language="java" contentType="text/html; charset=UTF-8"
02:     pageEncoding="UTF-8"%>
03:    <!DOCTYPE html>
04:    <html>
05:    <head>
06:    <meta charset="UTF-8">
07:    <title>エラー</title>
08:    </head>
09:    <body>
10:     <h1>エラー画面</h1>
11:     <%
12:       Exception e = (Exception) session.getAttribute("Except");
13:     %>
14:     <p><%=e.getMessage()%>
15:     <p>
16:      <a href="javascript:history.back();">戻る</a>
17:    </body>
18:    </html>
```

ファイル login.jsp

```
01:    <%@ page language="java" contentType="text/html; charset=UTF-8"
02:     pageEncoding="UTF-8"%>
03:    <!DOCTYPE html>
04:    <html>
05:    <head>
06:    <meta charset="UTF-8">
07:    <title>つぶやきアプリ ログイン</title>
08:    </head>
09:    <body>
10:     <h1>つぶやきアプリ</h1>
11:     <h2>ログイン</h2>
12:     <form action="j_security_check" method="post">
13:      ログイン ID : <input type="text" name="j_username"><br> パスワード : <input
14:       type="password" name="j_password"><br> <input
15:       type="submit" value="LOGIN"> <input type="reset">
16:     </form>
17:    </body>
18:    </html>
```

ファイル loginerror.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ ログインエラー</title>
</head>
<body>
 <h1>つぶやきアプリ</h1>
 <h2>ログインエラー</h2>
 ログイン ID またはパスワードに誤りがあります。
 <br> 再度入力してください。
 <br>
 <a href="javascript:history.back();">戻る</a>
</body>
</html>
```

ファイル tweetinput.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <h1>つぶやきアプリ</h1>
 <form action="TweetInputServlet" method="post">
  <p>
   名前:<br> <input type="text" name="name" maxlength="50">
  </p>
  <p>
   つぶやき:<br>
   <textarea rows="5" cols="40" name="tweet"></textarea>
  </p>
  <p>
   <input type="submit" value="送信"> <input type="reset">
   <input type="button" onclick="location.href='TweetListServlet'"
    value="戻る">
 </form>
</html>
```

ファイル tweetlist.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<%@ page import="java.util.ArrayList"%>
```

```
<%@ page import="java.text.SimpleDateFormat"%>
<%@ page import="chapter11.tweet.*"%>
<%@ page import="chapter11.parameter.*"%>
<%@ page import="java.net.URLEncoder"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="TweetArrayBean"
  class="chapter11.tweet.TweetArrayBean" scope="session" />
 <h1>つぶやきアプリ</h1>
 <%=request.getRemoteUser()%>：
 <a href="LogoutServlet">ログアウト</a>
 <%
   if (request.isUserInRole(UserRoleParameters.ROLE_ADMINS)) {
 %>
 <a href="UserListServlet">ユーザ管理</a>
 <%
   }
 %>
 <br>
 <br>
 <form action="tweetinput.html" method="get">
  <input type="submit" value="書き込み">
 </form>
 <br>
 <table border="1">
  <tr>
   <th>名前</th>
   <th>つぶやき</th>
   <th>日時</th>
  </tr>
  <%
    ArrayList<TweetBean> tweetArray = TweetArrayBean
      .getTweetArray();
     for (TweetBean record : tweetArray) {
  %>
  <tr>
   <td>
       <a
href="TweetPersonServlet?name=<%=URLEncoder.encode(record.getName(),"UTF-8")%>"><%
=record.getName()%></a>
   </td>
   <td><%=record.getTweet()%></td>
   <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
            .getDatetime())%></td>
  </tr>
```

```
54:       <%
55:         }
56:       %>
57:      </table>
58:     </body>
        </html>
```

ファイル tweetpersonlist.jsp

```
01:     <%@ page language="java" contentType="text/html; charset=UTF-8"
02:      pageEncoding="UTF-8"%>
03:     <%@ page import="java.util.ArrayList"%>
04:     <%@ page import="java.text.SimpleDateFormat"%>
05:     <%@ page import="chapter11.tweet.*"%>
06:     <%@ page import="chapter11.parameter.*"%>
07:     <!DOCTYPE html>
08:     <html>
09:     <head>
10:     <meta charset="UTF-8">
11:     <title>つぶやきアプリ</title>
12:     </head>
13:     <body>
14:      <jsp:useBean id="TweetArrayBean"
15:       class="chapter11.tweet.TweetArrayBean" scope="session" />
16:      <h1>つぶやきアプリ</h1>
17:      <h2><%=request.getParameter("name")%>さんのつぶやき
18:      </h2>
19:      <table>
20:       <tr>
21:        <td><form action="TweetListServlet" method="get">
22:          <input type="submit" value="戻る">
23:         </form></td>
24:        <%
25:          if (request.isUserInRole(UserRoleParameters.ROLE_ADMINS)) {
26:        %>
27:        <td><form action="TweetDeleteServlet" method="get">
28:          <input type="hidden" name="name"
29:           value="<%=request.getParameter("name")%>"> <input
30:           type="submit" value="削除">
31:         </form></td>
32:        <%
33:          }
34:        %>
35:       </tr>
36:      </table>
37:      <table border="1">
38:       <tr>
39:        <th>名前</th>
40:        <th>つぶやき</th>
41:        <th>日時</th>
```

```
42:       </tr>
43:       <%
44:         ArrayList<TweetBean> tweetArray = TweetArrayBean
45:           .getTweetArray();
46:           for (TweetBean record : tweetArray) {
47:       %>
48:       <tr>
49:        <td><%=record.getName()%></td>
50:        <td><%=record.getTweet()%></td>
51:        <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
52:                .getDatetime())%></td>
53:       </tr>
54:       <%
55:         }
56:       %>
57:      </table>
58:     </body>
59:     </html>
```

ファイル userinput.html

```
01:     <!DOCTYPE html>
02:     <html>
03:     <head>
04:     <meta charset="UTF-8">
05:     <title>つぶやきアプリ</title>
06:     </head>
07:     <body>
08:      <h1>つぶやきアプリ</h1>
09:      <h2>ユーザ作成</h2>
10:      <form action="UserInputServlet" method="post">
11:       <p>
12:        ユーザＩＤ:<input type="text" name="userid" maxlength="50"><br>
13:        パスワード:<input type="password" name="passwd" maxlength="50"><br>
14:        ロール： <input type="radio" name="role" value="users" checked>users
15:        <input type="radio" name="role" value="admins">admins
16:       </p>
17:       <p>
18:        <input type="submit" value="作成"> <input type="reset">
19:        <input type="button" onclick="location.href='./UserListServlet'"
20:         value="戻る">
21:      </form>
22:     </body>
23:     </html>
```

ファイル userlist.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="chapter11.user.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="UserArrayBean" class="chapter11.user.UserArrayBean"
  scope="session" />
 <h1>つぶやきアプリ</h1>
 <h2>ユーザ管理</h2>
 <%=request.getRemoteUser()%> :
 <a href="LogoutServlet">ログアウト</a>
 <a href="TweetListServlet">つぶやきへ戻る</a>
 <br>
 <br>
 <table border="1">
  <tr>
   <th>ユーザ ID</th>
   <th>ロール</th>
   <td colspan=2><input type="button"
    onclick="location.href='userinput.html'" value="ユーザ作成"></td>
  </tr>
  <%
    ArrayList<UserBean> userArray = UserArrayBean.getUserArray();
    for (UserBean record : userArray) {
  %>
  <tr>
   <td><%=record.getUserid()%></td>
   <td><%=record.getRole()%></td>
   <td><a href="UserPersonServlet?userid=<%=record.getUserid()%>">編集</a></td>
   <td><a href="UserDeleteServlet?userid=<%=record.getUserid()%>">削除</a></td>
  </tr>
  <%
    }
  %>
 </table>
</body>
</html>
```

ファイル userperson.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
```

```
<%@ page import="chapter11.parameter.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="UserBean" class="chapter11.user.UserBean"
  scope="session" />
 <h1>つぶやきアプリ</h1>
 <h2>ユーザ情報変更</h2>

 <form action="UserUpdateServlet" method="post">
  ユーザ ID:<input type="hidden" name="userid"
   value="<%=UserBean.getUserid()%>"><%=UserBean.getUserid()%><br>
  パスワード : <input type="password" name="passwd"
   value="<%=UserBean.getPasswd()%>"><br>
  <%
    String users = "checked";
    String admins = "";
    if (UserRoleParameters.ROLE_ADMINS.equalsIgnoreCase(UserBean.getRole())) {
      users = "";
      admins = "checked";
    }
  %>
  ロール :  <input type="radio" name="role" value="users" <%=users%>>
<%=UserRoleParameters.ROLE_USERS%>
  <input type="radio" name="role" value="admins" <%=admins%>>
<%=UserRoleParameters.ROLE_ADMINS%>
  <br> <input type="submit" value="変更"> <input type="reset">
  <input type="button" onclick="location.href='./UserListServlet'"
   value="戻る">
 </form>
</body>
</html>
```

ファイル web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>STEP3</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
```

```
12:       <welcome-file>default.htm</welcome-file>
13:       <welcome-file>default.jsp</welcome-file>
14:     </welcome-file-list>
15:
16:       <security-constraint>
17:       <web-resource-collection>
18:         <web-resource-name>FORM 認証</web-resource-name>
19:         <url-pattern>/*</url-pattern>
20:       </web-resource-collection>
21:       <auth-constraint>
22:         <role-name>admins</role-name>
23:       </auth-constraint>
24:     </security-constraint>
25:     <security-constraint>
26:       <web-resource-collection>
27:         <web-resource-name>FORM 認証</web-resource-name>
28:         <url-pattern>/TweetListServlet</url-pattern>
29:         <url-pattern>/TweetPersonServlet</url-pattern>
30:         <url-pattern>/TweetInputServlet</url-pattern>
31:         <url-pattern>/LogoutServlet</url-pattern>
32:         <url-pattern>/tweetinput.html</url-pattern>
33:         <url-pattern>/tweetlist.jsp</url-pattern>
34:         <url-pattern>/tweetpersonlist.jsp</url-pattern>
35:         <url-pattern>/error.jsp</url-pattern>
36:       </web-resource-collection>
37:       <auth-constraint>
38:         <role-name>admins</role-name>
39:         <role-name>users</role-name>
40:       </auth-constraint>
41:     </security-constraint>
42:     <login-config>
43:       <auth-method>FORM</auth-method>
44:       <form-login-config>
45:         <form-login-page>/login.jsp</form-login-page>
46:         <form-error-page>/loginerror.jsp</form-error-page>
47:       </form-login-config>
48:     </login-config>
49:     <security-role>
50:       <role-name>users</role-name>
51:     </security-role>
52:     <security-role>
53:       <role-name>admins</role-name>
54:     </security-role>
55:
56:   </web-app>
```

第4段階

プロジェクト名 STEP4

```
STEP4
├─src
│  ├─chapter11
│  │  │  LoginServlet.java
│  │  │  LogoutServlet.java
│  │  │
│  │  ├─dao
│  │  │      DAOBase.java
│  │  │      LoginDAO.java
│  │  │      TweetDAO.java
│  │  │      UserDAO.java
│  │  │
│  │  ├─exception
│  │  │      DatabaseException.java
│  │  │      ParameterException.java
│  │  │      SystemException.java
│  │  │
│  │  ├─parameter
│  │  │      DAOParameters.java
│  │  │      ExceptionParameters.java
│  │  │      TweetParameters.java
│  │  │      UserRoleParameters.java
│  │  │
│  │  ├─Tweet
│  │  │      TweetArrayBean.java
│  │  │      TweetBean.java
│  │  │      TweetDeleteServlet.java
│  │  │      TweetInputServlet.java
│  │  │      TweetListServlet.java
│  │  │      TweetPersonServlet.java
│  │  │
│  │  └─user
│  │          UserArrayBean.java
│  │          UserBean.java
│  │          UserDeleteServlet.java
│  │          UserInputServlet.java
│  │          UserListServlet.java
│  │          UserPersonServlet.java
│  │          UserUpdateServlet.java
│  │
│  └─filter
│          ExampleFilter.java
│
└─WebContent
       error.jsp
       login.html
```

```
        loginerror.jsp
        tweetinput.html
        tweetlist.jsp
        tweetpersonlist.jsp
        userinput.html
        userlist.jsp
        userperson.jsp
```

ファイル LoginServlet.java

```
01:  package chapter11;
02:
03:  import java.io.IOException;
04:  import javax.servlet.ServletException;
05:  import javax.servlet.annotation.WebServlet;
06:  import javax.servlet.http.HttpServlet;
07:  import javax.servlet.http.HttpServletRequest;
08:  import javax.servlet.http.HttpServletResponse;
09:  import javax.servlet.http.HttpSession;
10:
11:  import chapter11.dao.LoginDAO;
12:  import chapter11.dao.TweetDAO;
13:  import chapter11.exception.DatabaseException;
14:  import chapter11.exception.SystemException;
15:  import chapter11.tweet.TweetArrayBean;
16:  import chapter11.user.UserBean;
17:
18:  /**
19:   * Servlet implementation class TweetloginServlet
20:   */
21:  @WebServlet("/LoginServlet")
22:  public class LoginServlet extends HttpServlet {
23:
24:    @Override
25:    protected void doPost(HttpServletRequest request,
26:        HttpServletResponse response) throws ServletException,
27:        IOException {
28:
29:      String actionName = request.getParameter("action");
30:      if (actionName.equals("login")) {
31:
32:        String userid = request.getParameter("login_id");
33:        String password = request.getParameter("login_pass");
34:
35:        try {
36:          LoginDAO loginDao = new LoginDAO();
37:          UserBean user = loginDao.getUser(userid, password);
38:          String retUrl = "/loginerror.jsp";
39:          HttpSession session = request.getSession();
40:          if(user != null) {
41:            session.setAttribute("LoginUser", user);
42:
43:            TweetArrayBean tweetArray;
44:
45:            TweetDAO dao = new TweetDAO();
46:            tweetArray = dao.getTweetArray();
47:            session.setAttribute("TweetArrayBean", tweetArray);
48:            retUrl = "/tweetlist.jsp";
49:          }
```

```
50:           getServletContext().getRequestDispatcher(retUrl)
51:               .forward(request, response);
52:         } catch (SystemException e) {
53:           e.printStackTrace();
54:           HttpSession session = request.getSession();
55:           session.setAttribute("Except", e);
56:           getServletContext().getRequestDispatcher("/error.jsp")
57:               .forward(request, response);
58:         } catch (DatabaseException e) {
59:           e.printStackTrace();
60:           HttpSession session = request.getSession();
61:           session.setAttribute("Except", e);
62:           getServletContext().getRequestDispatcher("/error.jsp")
63:               .forward(request, response);
64:         }
65:
66:       } else {
67:         getServletContext().getRequestDispatcher("/error.jsp")
68:             .forward(request, response);
69:       }
70:     }
71:
72:     @Override
73:     protected void doGet(HttpServletRequest request,
74:         HttpServletResponse response) throws ServletException,
75:         IOException {
76:
77:       HttpSession session = request.getSession(false);
78:       if (session != null) {
79:         session.invalidate();
80:       }
81:       response.sendRedirect("login.html");
82:     }
83:   }
```

ファイル LogoutServlet.java

```
01:   package chapter11;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   @WebServlet("/LogoutServlet")
13:   public class LogoutServlet extends HttpServlet {
```

```
14:
15:       @Override
16:       protected void doGet(HttpServletRequest request,
17:           HttpServletResponse response) throws ServletException,
18:           IOException {
19:
20:         System.out.println("LogoutServlet");
21:         HttpSession session = request.getSession(false);
22:         if (session != null) {
23:           System.out.println("session.invalidate()");
24:           session.invalidate();
25:         }
26:         response.sendRedirect("login.html");
27:       }
28:     }
```

ファイル DAOBase.java

```
01:     package chapter11.dao;
02:
03:     import java.sql.Connection;
04:     import java.sql.DriverManager;
05:     import java.sql.SQLException;
06:     import java.sql.Statement;
07:
08:     import chapter11.exception.DatabaseException;
09:     import chapter11.exception.SystemException;
10:     import chapter11.parameter.DAOParameters;
11:     import chapter11.parameter.ExceptionParameters;
12:
13:     public class DAOBase {
14:
15:       Connection con;
16:
17:       protected void open() throws DatabaseException, SystemException {
18:         try {
19:           Class.forName(DAOParameters.DRIVER_NAME);
20:           con = DriverManager.getConnection(DAOParameters.CONNECT_STRING,
21:               DAOParameters.USERID, DAOParameters.PASSWORD);
22:         } catch (ClassNotFoundException e) {
23:           throw new SystemException(ExceptionParameters.SYSTEM_EXCEPTION_MESSAGE, e);
24:         } catch (SQLException e) {
25:           throw new DatabaseException(
26:               ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
27:         }
28:       }
29:
30:       protected void close(Statement stmt) throws DatabaseException {
31:         try {
32:           if (stmt != null) {
```

```
33:         stmt.close();
34:       }
35:       if (con != null) {
36:         con.close();
37:       }
38:     } catch (SQLException e) {
39:       throw new DatabaseException(
40:           ExceptionParameters.DATABASE_CLOSE_EXCEPTION_MESSAGE, e);
41:     }
42:   }
43:
44: }
```

ファイル LoginDAO.java

```
01: package chapter11.dao;
02:
03: import java.sql.PreparedStatement;
04: import java.sql.ResultSet;
05: import java.sql.SQLException;
06:
07: import chapter11.exception.DatabaseException;
08: import chapter11.exception.SystemException;
09: import chapter11.parameter.ExceptionParameters;
10: import chapter11.parameter.UserRoleParameters;
11: import chapter11.user.UserBean;
12:
13: public class LoginDAO extends DAOBase {
14:
15:   public UserBean getUser(String userid, String password)
16:       throws DatabaseException, SystemException {
17:     UserBean record = null;
18:     PreparedStatement stmt = null;
19:     this.open();
20:     try {
21:       stmt = con.prepareStatement(UserRoleParameters.SQL_SELECT_USER);
22:       stmt.setString(1, userid);
23:       stmt.setString(2, password);
24:       ResultSet rs = stmt.executeQuery();
25:       if (rs.next()) {
26:         record = new UserBean();
27:         record.setUserid(rs.getString(UserRoleParameters.USERID));
28:         record.setPasswd(rs.getString(UserRoleParameters.PASSWORD));
29:         record.setRole(rs.getString(UserRoleParameters.ROLE));
30:       }
31:     } catch (SQLException e) {
32:       throw new DatabaseException(
33:           ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE,
```

```
34:             e);
35:       } finally {
36:         this.close(stmt);
37:       }
38:       return record;
39:     }
40:   }
```

ファイル TweetDAO.java

```
01:   package chapter11.dao;
02:
03:   import java.sql.PreparedStatement;
04:   import java.sql.ResultSet;
05:   import java.sql.SQLException;
06:   import java.sql.Statement;
07:   import java.text.SimpleDateFormat;
08:
09:   import chapter11.exception.DatabaseException;
10:   import chapter11.exception.SystemException;
11:   import chapter11.parameter.ExceptionParameters;
12:   import chapter11.parameter.TweetParameters;
13:   import chapter11.tweet.TweetArrayBean;
14:   import chapter11.tweet.TweetBean;
15:
16:   public class TweetDAO extends DAOBase {
17:
18:     public TweetArrayBean getTweetArray() throws DatabaseException,
19:         SystemException {
20:       Statement stmt = null;
21:       TweetArrayBean tweetArray = new TweetArrayBean();
22:       this.open();
23:       try {
24:         stmt = con.createStatement();
25:         ResultSet rs = stmt.executeQuery(TweetParameters.SQL_SELECT_ALL);
26:         while (rs.next()) {
27:           TweetBean record = new TweetBean();
28:           record.setName(rs.getString(TweetParameters.NAME));
29:           record.setTweet(rs.getString(TweetParameters.TWEET));
30:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
31:           tweetArray.addTweet(record);
32:         }
33:       } catch (SQLException e) {
34:         throw new DatabaseException(
35:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
36:       } finally {
37:         this.close(stmt);
38:       }
```

```
39:       return tweetArray;
40:     }
41:
42:     public TweetArrayBean getPersonTweetArray(String name)
43:         throws DatabaseException, SystemException {
44:       PreparedStatement stmt = null;
45:       TweetArrayBean tweetArray = new TweetArrayBean();
46:       this.open();
47:       try {
48:         stmt = con.prepareStatement(TweetParameters.SQL_SELECT_PERSON);
49:         stmt.setString(1, name);
50:         ResultSet rs = stmt.executeQuery();
51:         while (rs.next()) {
52:           TweetBean record = new TweetBean();
53:           record.setName(rs.getString(TweetParameters.NAME));
54:           record.setTweet(rs.getString(TweetParameters.TWEET));
55:           record.setDatetime(rs.getTimestamp(TweetParameters.DATE));
56:           tweetArray.addTweet(record);
57:         }
58:       } catch (SQLException e) {
59:         throw new DatabaseException(
60:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
61:       } finally {
62:         this.close(stmt);
63:       }
64:       return tweetArray;
65:     }
66:
67:     public void setTweet(TweetBean tweet) throws DatabaseException,
68:         SystemException {
69:       PreparedStatement stmt = null;
70:       this.open();
71:       try {
72:         stmt = con.prepareStatement(TweetParameters.SQL_INSERT);
73:         stmt.setString(1, tweet.getName());
74:         stmt.setString(2, tweet.getTweet());
75:         stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
76:             .format(tweet.getDatetime()));
77:         stmt.executeUpdate();
78:       } catch (SQLException e) {
79:         throw new DatabaseException(
80:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
81:       } finally {
82:         this.close(stmt);
83:       }
84:     }
85:
86:     public int deleteTweet(String name) throws DatabaseException,
87:         SystemException {
88:       PreparedStatement stmt = null;
```

```
89:         this.open();
90:         try {
91:           stmt = con.prepareStatement(TweetParameters.SQL_DELETE);
92:           stmt.setString(1, name);
93:           return stmt.executeUpdate();
94:         } catch (SQLException e) {
95:           throw new DatabaseException(
96:               ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
97:         } finally {
98:           this.close(stmt);
99:         }
100:      }
101:    }
```

ファイル UserDAO.java

```
01:  package chapter11.dao;
02:
03:  import java.sql.PreparedStatement;
04:  import java.sql.ResultSet;
05:  import java.sql.SQLException;
06:  import java.sql.Statement;
07:
08:  import chapter11.exception.DatabaseException;
09:  import chapter11.exception.SystemException;
10:  import chapter11.parameter.ExceptionParameters;
11:  import chapter11.parameter.UserRoleParameters;
12:  import chapter11.user.UserArrayBean;
13:  import chapter11.user.UserBean;
14:
15:  public class UserDAO extends DAOBase {
16:
17:    public UserArrayBean getUserArray() throws DatabaseException,
18:        SystemException {
19:      Statement stmt = null;
20:      UserArrayBean userArray = new UserArrayBean();
21:      this.open();
22:      try {
23:        stmt = con.createStatement();
24:        ResultSet rs = stmt
25:            .executeQuery(UserRoleParameters.SQL_SELECT_ALL);
26:        while (rs.next()) {
27:          UserBean record = new UserBean();
28:          record.setUserid(rs.getString(UserRoleParameters.USERID));
29:          record.setRole(rs.getString(UserRoleParameters.ROLE));
30:          userArray.addUser(record);
31:        }
32:      } catch (SQLException e) {
33:        throw new DatabaseException(
34:            ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
35:      } finally {
36:        this.close(stmt);
37:      }
38:      return userArray;
39:    }
40:
41:    public UserBean getUser(String userid) throws DatabaseException,
42:        SystemException {
43:      UserBean record = null;
44:      PreparedStatement stmt = null;
45:      this.open();
46:      try {
47:        stmt = con.prepareStatement(UserRoleParameters.SQL_SELECT_PERSON);
48:        stmt.setString(1, userid);
49:        ResultSet rs = stmt.executeQuery();
```

```
50:         rs.next();
51:         record = new UserBean();
52:         record.setUserid(rs.getString(UserRoleParameters.USERID));
53:         record.setPasswd(rs.getString(UserRoleParameters.PASSWORD));
54:         record.setRole(rs.getString(UserRoleParameters.ROLE));
55:       } catch (SQLException e) {
56:         throw new DatabaseException(
57:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
58:       } finally {
59:         this.close(stmt);
60:       }
61:       return record;
62:     }
63:
64:     public void addUser(UserBean user) throws DatabaseException,
65:         SystemException {
66:
67:       PreparedStatement stmt = null;
68:       this.open();
69:       try {
70:         stmt = con
71:             .prepareStatement(UserRoleParameters.SQL_INSERT_USERTABLE);
72:         stmt.setString(1, user.getUserid());
73:         stmt.setString(2, user.getPasswd());
74:         stmt.executeUpdate();
75:       } catch (SQLException e) {
76:         throw new DatabaseException(
77:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
78:       } finally {
79:         this.close(stmt);
80:       }
81:       this.open();
82:       try {
83:         stmt = con
84:             .prepareStatement(UserRoleParameters.SQL_INSERT_USERROLETABLE);
85:         stmt.setString(1, user.getUserid());
86:         stmt.setString(2, user.getRole());
87:         stmt.executeUpdate();
88:       } catch (SQLException e) {
89:         throw new DatabaseException(
90:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
91:       } finally {
92:         this.close(stmt);
93:       }
94:
95:     }
96:
97:     public void updateUser(UserBean user) throws DatabaseException,
98:         SystemException {
99:       PreparedStatement stmt = null;
```

```
100:       this.open();
101:       try {
102:         stmt = con.prepareStatement(UserRoleParameters.SQL_UPDATE_USER);
103:         stmt.setString(1, user.getPasswd());
104:         stmt.setString(2, user.getUserid());
105:         stmt.executeUpdate();
106:       } catch (SQLException e) {
107:         throw new DatabaseException(
108:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
109:       } finally {
110:         this.close(stmt);
111:       }
112:       this.open();
113:       try {
114:         stmt = con.prepareStatement(UserRoleParameters.SQL_UPDATE_ROLE);
115:         stmt.setString(1, user.getRole());
116:         stmt.setString(2, user.getUserid());
117:         stmt.executeUpdate();
118:       } catch (SQLException e) {
119:         throw new DatabaseException(
120:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
121:       } finally {
122:         this.close(stmt);
123:       }
124:
125:     }
126:
127:     public void deleteUser(String userid) throws DatabaseException,
128:         SystemException {
129:       PreparedStatement stmt = null;
130:       this.open();
131:       try {
132:         stmt = con.prepareStatement(UserRoleParameters.SQL_DELETE_ROLE);
133:         stmt.setString(1, userid);
134:         stmt.executeUpdate();
135:       } catch (SQLException e) {
136:         throw new DatabaseException(
137:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
138:       } finally {
139:         this.close(stmt);
140:       }
141:       this.open();
142:       try {
143:         stmt = con.prepareStatement(UserRoleParameters.SQL_DELETE_USER);
144:         stmt.setString(1, userid);
145:         stmt.executeUpdate();
146:       } catch (SQLException e) {
147:         throw new DatabaseException(
148:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
149:       } finally {
```

```
150:         this.close(stmt);
151:       }
152:
153:     }
154:
155:   }
```

ファイル DatabaseException.java

```
01:   package chapter11.exception;
02:
03:   public class DatabaseException extends Exception {
04:
05:     public DatabaseException(String message, Throwable cause,
06:         boolean enableSuppression, boolean writableStackTrace) {
07:       super(message, cause, enableSuppression, writableStackTrace);
08:     }
09:
10:     public DatabaseException(String message, Throwable cause) {
11:       super(message, cause);
12:     }
13:
14:     public DatabaseException(String message) {
15:       super(message);
16:     }
17:
18:     public DatabaseException(Throwable cause) {
19:       super(cause);
20:     }
21:
22:   }
```

ファイル ParameterException.java

```
01:   package chapter11.exception;
02:
03:   public class ParameterException extends Exception {
04:
05:     public ParameterException(String message) {
06:       super(message);
07:     }
08:
09:     public ParameterException(Throwable cause) {
10:       super(cause);
11:     }
12:
13:     public ParameterException(String message, Throwable cause) {
14:       super(message, cause);
15:     }
16:
```

```
17:     public ParameterException(String message, Throwable cause,
18:         boolean enableSuppression, boolean writableStackTrace) {
19:       super(message, cause, enableSuppression, writableStackTrace);
20:     }
21:
22:   }
```

ファイル SystemException.java

```
01:   package chapter11.exception;
02:
03:   public class SystemException extends Exception {
04:
05:     public SystemException(String message, Throwable cause,
06:         boolean enableSuppression, boolean writableStackTrace) {
07:       super(message, cause, enableSuppression, writableStackTrace);
08:     }
09:
10:     public SystemException(String message, Throwable cause) {
11:       super(message, cause);
12:     }
13:
14:     public SystemException(String message) {
15:       super(message);
16:     }
17:
18:     public SystemException(Throwable cause) {
19:       super(cause);
20:     }
21:
22:   }
```

ファイル DAOParameters.java

```
01:   package chapter11.parameter;
02:
03:   public class DAOParameters {
04:     public static final String DRIVER_NAME = "com.mysql.jdbc.Driver";
05:     public static final String CONNECT_STRING =
06:   "jdbc:mysql://localhost:3306/tweet?serverTimezone=JST";
07:     public static final String USERID = "user";
08:     public static final String PASSWORD = "password";
09:   }
```

ファイル ExceptionParameters.java

```
01:   package chapter11.parameter;
02:
03:   public class ExceptionParameters {
04:     public static final String SYSTEM_EXCEPTION_MESSAGE = "システムエラーが発生しました";
```

```
05:
06:       public static final String DATABASE_CONNECTION_EXCEPTION_MESSAGE = "データベースへの接
07:     続時にエラーが発生しました";
08:       public static final String DATABASE_CLOSE_EXCEPTION_MESSAGE = "データベースからの切断時
09:     にエラーが発生しました";
10:
11:       public static final String Parameter_FORMAT_EXCEPTION_MESSAGE = "入力したデータの形式が
12:     正しくありません";
13:     }
```

ファイル TweetParameters.java

```
01:     package chapter11.parameter;
02:
03:     public class TweetParameters {
04:       public static final String SQL_SELECT_ALL = "SELECT * FROM TWEET ORDER BY DATE DESC";
05:       public static final String SQL_SELECT_PERSON = "SELECT * FROM TWEET WHERE NAME = ?
06:     ORDER BY DATE DESC";
07:
08:       public static final String SQL_INSERT = "INSERT INTO TWEET VALUES(?,?,?)";
09:
10:       public static final String SQL_DELETE = "DELETE FROM TWEET WHERE NAME=?";
11:
12:       public static final String NAME = "NAME";
13:       public static final String TWEET = "TWEET";
14:       public static final String DATE = "DATE";
15:     }
16:
```

ファイル UserRoleParameters.java

```
01:     package chapter11.parameter;
02:
03:     public class UserRoleParameters {
04:       public static final String ROLE_ADMINS = "admins";
05:       public static final String ROLE_USERS = "users";
06:
07:       public static final String SQL_SELECT_USER = "SELECT * FROM USERS,USER_ROLES WHERE
08:     USERS.USERID = USER_ROLES.USERID AND USERS.USERID=? AND USERS.PASSWORD=?";
09:
10:       public static final String SQL_SELECT_ALL = "SELECT * FROM USER_ROLES";
11:       public static final String SQL_SELECT_PERSON = "SELECT * FROM USERS,USER_ROLES WHERE
12:     USERS.USERID = USER_ROLES.USERID AND USERS.USERID=?";
13:       public static final String SQL_INSERT_USERTABLE = "INSERT INTO USERS VALUES(?,?)";
14:       public static final String SQL_INSERT_USERROLETABLE = "INSERT INTO USER_ROLES
15:     VALUES(?,?)";
16:       public static final String SQL_UPDATE_USER = "UPDATE USERS SET PASSWORD=? WHERE
17:     USERID=?";
18:       public static final String SQL_UPDATE_ROLE = "UPDATE USER_ROLES SET ROLE=? WHERE
19:     USERID=?";
```

```
20:     public static final String SQL_DELETE_USER = "DELETE FROM USERS WHERE USERID=?";
21:     public static final String SQL_DELETE_ROLE = "DELETE FROM USER_ROLES WHERE USERID=?";
22:
23:     public static final String USERID = "userid";
24:     public static final String PASSWORD = "password";
25:     public static final String ROLE = "role";
26:
27:   }
28:
29:
```

ファイル TweetArrayBean.java

```
01:  package chapter11.tweet;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class TweetArrayBean implements Serializable {
07:
08:    private ArrayList<TweetBean> tweetArray;
09:
10:    public TweetArrayBean() {
11:      tweetArray = new ArrayList<TweetBean>();
12:    }
13:
14:    public void addTweet(TweetBean obj) {
15:      tweetArray.add(obj);
16:    }
17:
18:    public int getArraySize() {
19:      return tweetArray.size();
20:    }
21:
22:    public ArrayList<TweetBean> getTweetArray() {
23:      return tweetArray;
24:    }
25:
26:    public void setTweetArray(ArrayList<TweetBean> tweetArray) {
27:      this.tweetArray = tweetArray;
28:    }
29:
30:  }
```

ファイル TweetBean.java

```
01:  package chapter11.tweet;
02:
03:  import java.io.Serializable;
04:  import java.util.Date;
05:
06:  public class TweetBean implements Serializable {
07:
08:    private String name;
09:    private String tweet;
10:    private Date datetime;
11:
12:    public TweetBean() {
13:    }
14:
15:    public String getName() {
```

```
16:       return name;
17:     }
18:
19:     public void setName(String name) {
20:       this.name = name;
21:     }
22:
23:     public String getTweet() {
24:       return tweet;
25:     }
26:
27:     public void setTweet(String tweet) {
28:       this.tweet = tweet;
29:     }
30:
31:     public Date getDatetime() {
32:       return datetime;
33:     }
34:
35:     public void setDatetime(Date datetime) {
36:       this.datetime = datetime;
37:     }
38:
39:   }
```

ファイル TweetDeleteServlet.java

```
01:   package chapter11.tweet;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   import chapter11.dao.TweetDAO;
13:   import chapter11.exception.DatabaseException;
14:   import chapter11.exception.SystemException;
15:   import chapter11.user.UserBean;
16:
17:   @WebServlet("/TweetDeleteServlet")
18:   public class TweetDeleteServlet extends HttpServlet {
19:
20:     @Override
21:     protected void doGet(HttpServletRequest request,
22:         HttpServletResponse response) throws ServletException,
23:         IOException {
```

```
24:
25:       try {
26:         HttpSession session = request.getSession(false);
27:         if (session != null) {
28:           System.out.println("TweetPersonServlet session != null");
29:           UserBean user = (UserBean) session.getAttribute(
30:               "LoginUser");
31:           if (user != null) {
32:             String name = request.getParameter("name");
33:
34:             TweetDAO dao = new TweetDAO();
35:             if(dao.deleteTweet(name) == 1) {
36:               response.sendRedirect("TweetListServlet");
37:             } else {
38:               response.sendRedirect("LogoutServlet");
39:             }
40:           } else {
41:             response.sendRedirect("LogoutServlet");
42:           }
43:         } else {
44:           getServletContext().getRequestDispatcher("/login.html").forward(
45:               request, response);
46:         }
47:       } catch (SystemException e) {
48:         e.printStackTrace();
49:         HttpSession session = request.getSession();
50:         session.setAttribute("Except", e);
51:         getServletContext().getRequestDispatcher("/error.jsp")
52:             .forward(request, response);
53:       } catch (DatabaseException e) {
54:         e.printStackTrace();
55:         HttpSession session = request.getSession();
56:         session.setAttribute("Except", e);
57:         getServletContext().getRequestDispatcher("/error.jsp")
58:             .forward(request, response);
59:       }
60:     }
61:   }
```

ファイル TweetInputServlet.java

```
01:   package chapter11.tweet;
02:
03:   import java.io.IOException;
04:   import java.util.Date;
05:
06:   import javax.servlet.ServletException;
07:   import javax.servlet.annotation.WebServlet;
08:   import javax.servlet.http.HttpServlet;
09:   import javax.servlet.http.HttpServletRequest;
```

```
10:  import javax.servlet.http.HttpServletResponse;
11:  import javax.servlet.http.HttpSession;
12:
13:  import chapter11.dao.TweetDAO;
14:  import chapter11.exception.DatabaseException;
15:  import chapter11.exception.SystemException;
16:
17:  @WebServlet("/TweetInputServlet")
18:  public class TweetInputServlet extends HttpServlet {
19:
20:    @Override
21:    protected void doPost(HttpServletRequest request,
22:        HttpServletResponse response) throws ServletException,
23:        IOException {
24:
25:      try {
26:        String name = request.getParameter("name");
27:        String tweet = request.getParameter("tweet");
28:        Date date = new Date();
29:
30:        TweetBean tweetBean = new TweetBean();
31:        tweetBean.setName(name);
32:        tweetBean.setTweet(tweet);
33:        tweetBean.setDatetime(date);
34:
35:        TweetDAO dao = new TweetDAO();
36:        dao.setTweet(tweetBean);
37:
38:        response.sendRedirect("TweetListServlet");
39:      } catch (SystemException e) {
40:        e.printStackTrace();
41:        HttpSession session = request.getSession();
42:        session.setAttribute("Except", e);
43:        getServletContext().getRequestDispatcher("/error.jsp")
44:            .forward(request, response);
45:      } catch (DatabaseException e) {
46:        e.printStackTrace();
47:        HttpSession session = request.getSession();
48:        session.setAttribute("Except", e);
49:        getServletContext().getRequestDispatcher("/error.jsp")
50:            .forward(request, response);
51:      }
52:    }
53:
54:    @Override
55:    protected void doGet(HttpServletRequest request,
56:        HttpServletResponse response) throws ServletException,
57:        IOException {
58:      HttpSession session = request.getSession(false);
59:      if (session != null) {
```

```
60:         session.invalidate();
61:       }
62:       response.sendRedirect("login.html");
63:     }
64:   }
```

ファイル TweetListServlet.java

```
01:   package chapter11.tweet;
02:
03:   import java.io.IOException;
04:
05:   import javax.servlet.ServletException;
06:   import javax.servlet.annotation.WebServlet;
07:   import javax.servlet.http.HttpServlet;
08:   import javax.servlet.http.HttpServletRequest;
09:   import javax.servlet.http.HttpServletResponse;
10:   import javax.servlet.http.HttpSession;
11:
12:   import chapter11.dao.TweetDAO;
13:   import chapter11.exception.DatabaseException;
14:   import chapter11.exception.SystemException;
15:   import chapter11.user.UserBean;
16:
17:   @WebServlet("/TweetListServlet")
18:   public class TweetListServlet extends HttpServlet {
19:
20:     @Override
21:     protected void doGet(HttpServletRequest request,
22:         HttpServletResponse response) throws ServletException,
23:         IOException {
24:
25:       String retUrl = "/login.html";
26:       try {
27:         TweetArrayBean tweetArray;
28:         HttpSession session = request.getSession(false);
29:         if(session != null) {
30:           UserBean user = (UserBean) session.getAttribute("LoginUser");
31:           if (user != null) {
32:             TweetDAO dao = new TweetDAO();
33:             tweetArray = dao.getTweetArray();
34:             session.setAttribute("TweetArrayBean", tweetArray);
35:             retUrl = "/tweetlist.jsp";
36:           }
37:         }
38:       } catch (SystemException e) {
39:         e.printStackTrace();
40:         HttpSession session = request.getSession();
41:         session.setAttribute("Except", e);
42:         retUrl = "/error.jsp";
```

```
43:         } catch (DatabaseException e) {
44:           e.printStackTrace();
45:           HttpSession session = request.getSession();
46:           session.setAttribute("Except", e);
47:           retUrl = "/error.jsp";
48:         }
49:         getServletContext().getRequestDispatcher(retUrl).forward(
50:             request, response);
51:       }
52:     }
```

ファイル TweetPersonServlet.java

```
01:     package chapter11.tweet;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.TweetDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:     import chapter11.user.UserBean;
16:
17:     @WebServlet("/TweetPersonServlet")
18:     public class TweetPersonServlet extends HttpServlet {
19:       private static final long serialVersionUID = 1L;
20:
21:       @Override
22:       protected void doGet(HttpServletRequest request,
23:           HttpServletResponse response) throws ServletException,
24:           IOException {
25:
26:         String retUrl = "/login.html";
27:         try {
28:           HttpSession session = request.getSession(false);
29:           if (session != null) {
30:             UserBean user = (UserBean) session.getAttribute(
31:                 "LoginUser");
32:             if (user != null) {
33:               String name = request.getParameter("name");
34:               TweetDAO dao = new TweetDAO();
35:               TweetArrayBean tweetArray = dao.getPersonTweetArray(name);
36:               session.setAttribute("TweetArrayBean", tweetArray);
37:               retUrl = "/tweetpersonlist.jsp";
```

```
38:         }
39:       }
40:     } catch (SystemException e) {
41:       e.printStackTrace();
42:       HttpSession session = request.getSession();
43:       session.setAttribute("Except", e);
44:       retUrl = "/error.jsp";
45:     } catch (DatabaseException e) {
46:       e.printStackTrace();
47:       HttpSession session = request.getSession();
48:       session.setAttribute("Except", e);
49:       retUrl = "/error.jsp";
50:     }
51:     getServletContext().getRequestDispatcher(retUrl).forward(
52:        request, response);
53:   }
54: }
```

ファイル UserArrayBean.java

```
01:  package chapter11.user;
02:
03:  import java.io.Serializable;
04:  import java.util.ArrayList;
05:
06:  public class UserArrayBean implements Serializable {
07:
08:    private ArrayList<UserBean> userArray;
09:
10:    public UserArrayBean() {
11:      userArray = new ArrayList<UserBean>();
12:    }
13:
14:    public void addUser(UserBean obj) {
15:      userArray.add(obj);
16:    }
17:
18:    public int getArraySize() {
19:      return userArray.size();
20:    }
21:
22:    public ArrayList<UserBean> getUserArray() {
23:      return userArray;
24:    }
25:
26:    public void setUserArray(ArrayList<UserBean> userArray) {
27:      this.userArray = userArray;
28:    }
29:
30:  }
```

ファイル UserBean.java

```
01:  package chapter11.user;
02:
03:  import java.io.Serializable;
04:
05:  public class UserBean implements Serializable {
06:
07:    private String userid;
08:    private String passwd;
09:    private String role;
10:
11:    public UserBean() {
12:    }
13:
14:    public String getUserid() {
15:      return userid;
```

```
16:       }
17:
18:       public void setUserid(String userid) {
19:         this.userid = userid;
20:       }
21:
22:       public String getPasswd() {
23:         return passwd;
24:       }
25:
26:       public void setPasswd(String passwd) {
27:         this.passwd = passwd;
28:       }
29:
30:       public String getRole() {
31:         return role;
32:       }
33:
34:       public void setRole(String role) {
35:         this.role = role;
36:       }
37:
38:     }
```

ファイル UserDeleteServlet.java

```
01:     package chapter11.user;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.UserDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:     import chapter11.parameter.UserRoleParameters;
16:
17:     @WebServlet("/UserDeleteServlet")
18:     public class UserDeleteServlet extends HttpServlet {
19:
20:       @Override
21:       protected void doGet(HttpServletRequest request,
22:           HttpServletResponse response) throws ServletException,
23:           IOException {
24:
```

```
25:         try {
26:           HttpSession session = request.getSession(false);
27:           if (session != null) {
28:             UserBean user = (UserBean) session.getAttribute(
29:                 "LoginUser");
30:             if (user != null && user.getRole().equals(
31:                 UserRoleParameters.ROLE_ADMINS)) {
32:               String userid = request.getParameter("userid");
33:               UserDAO dao = new UserDAO();
34:               dao.deleteUser(userid);
35:               response.sendRedirect("UserListServlet");
36:             } else {
37:               response.sendRedirect("LogoutServlet");
38:             }
39:           } else {
40:             response.sendRedirect("login.html");
41:           }
42:         } catch (SystemException e) {
43:           e.printStackTrace();
44:           HttpSession session = request.getSession();
45:           session.setAttribute("Except", e);
46:           getServletContext().getRequestDispatcher("/error.jsp")
47:               .forward(request, response);
48:         } catch (DatabaseException e) {
49:           e.printStackTrace();
50:           HttpSession session = request.getSession();
51:           session.setAttribute("Except", e);
52:           getServletContext().getRequestDispatcher("/error.jsp")
53:               .forward(request, response);
54:         }
55:       }
56:
57:     }
```

ファイル UserInputServlet.java

```
01:     package chapter11.user;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.UserDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
```

```
15:
16:  @WebServlet("/UserInputServlet")
17:  public class UserInputServlet extends HttpServlet {
18:
19:    @Override
20:    protected void doPost(HttpServletRequest request, HttpServletResponse response)
21:        throws ServletException, IOException {
22:      try {
23:        String userid = request.getParameter("userid");
24:        String passwd = request.getParameter("passwd");
25:        String role = request.getParameter("role");
26:
27:        UserBean userBean = new UserBean();
28:        userBean.setUserid(userid);
29:        userBean.setPasswd(passwd);
30:        userBean.setRole(role);
31:
32:        UserDAO dao = new UserDAO();
33:        dao.addUser(userBean);
34:
35:        response.sendRedirect("UserListServlet");
36:      } catch (SystemException e) {
37:        e.printStackTrace();
38:        HttpSession session = request.getSession();
39:        session.setAttribute("Except", e);
40:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
41:            response);
42:      } catch (DatabaseException e) {
43:        e.printStackTrace();
44:        HttpSession session = request.getSession();
45:        session.setAttribute("Except", e);
46:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:            response);
48:      }
49:    }
50:
51:    @Override
52:    protected void doGet(HttpServletRequest request,
53:        HttpServletResponse response) throws ServletException,
54:        IOException {
55:
56:      HttpSession session = request.getSession(false);
57:      if (session != null) {
58:        session.invalidate();
59:      }
60:      response.sendRedirect("login.html");
61:    }
62:
63:  }
```

ファイル UserListServlet.java

```
01:  package chapter11.user;
02:
03:  import java.io.IOException;
04:
05:  import javax.servlet.ServletException;
06:  import javax.servlet.annotation.WebServlet;
07:  import javax.servlet.http.HttpServlet;
08:  import javax.servlet.http.HttpServletRequest;
09:  import javax.servlet.http.HttpServletResponse;
10:  import javax.servlet.http.HttpSession;
11:
12:  import chapter11.dao.UserDAO;
13:  import chapter11.exception.DatabaseException;
14:  import chapter11.exception.SystemException;
15:  import chapter11.parameter.UserRoleParameters;
16:
17:  @WebServlet("/UserListServlet")
18:  public class UserListServlet extends HttpServlet {
19:
20:    @Override
21:    protected void doGet(HttpServletRequest request,
22:        HttpServletResponse response) throws ServletException,
23:        IOException {
24:
25:      try {
26:        HttpSession session = request.getSession(false);
27:        if (session != null) {
28:          UserBean user = (UserBean) session.getAttribute(
29:              "LoginUser");
30:          if (user != null && user.getRole().equals(
31:              UserRoleParameters.ROLE_ADMINS)) {
32:            UserArrayBean userArray;
33:            UserDAO dao = new UserDAO();
34:            userArray = dao.getUserArray();
35:            session.setAttribute("UserArrayBean", userArray);
36:            getServletContext().getRequestDispatcher("/userlist.jsp")
37:                .forward(request, response);
38:          } else {
39:            response.sendRedirect("LogoutServlet");
40:          }
41:        } else {
42:          response.sendRedirect("login.html");
43:        }
44:      } catch (SystemException e) {
45:        e.printStackTrace();
46:        HttpSession session = request.getSession();
47:        session.setAttribute("Except", e);
48:        getServletContext().getRequestDispatcher("/error.jsp")
49:            .forward(request, response);
```

```
50:         } catch (DatabaseException e) {
51:           e.printStackTrace();
52:           HttpSession session = request.getSession();
53:           session.setAttribute("Except", e);
54:           getServletContext().getRequestDispatcher("/error.jsp")
55:               .forward(request, response);
56:         }
57:       }
58:     }
```

ファイル UserPersonServlet.java

```
01:     package chapter11.user;
02:
03:     import java.io.IOException;
04:
05:     import javax.servlet.ServletException;
06:     import javax.servlet.annotation.WebServlet;
07:     import javax.servlet.http.HttpServlet;
08:     import javax.servlet.http.HttpServletRequest;
09:     import javax.servlet.http.HttpServletResponse;
10:     import javax.servlet.http.HttpSession;
11:
12:     import chapter11.dao.UserDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:     import chapter11.parameter.UserRoleParameters;
16:
17:     @WebServlet("/UserPersonServlet")
18:     public class UserPersonServlet extends HttpServlet {
19:
20:       @Override
21:       protected void doGet(HttpServletRequest request,
22:           HttpServletResponse response) throws ServletException,
23:           IOException {
24:
25:         try {
26:           HttpSession session = request.getSession(false);
27:           if (session != null) {
28:             UserBean user = (UserBean) session.getAttribute(
29:                 "LoginUser");
30:             if (user != null && user.getRole().equals(UserRoleParameters.ROLE_ADMINS)) {
31:               String userid = request.getParameter("userid");
32:               UserDAO dao = new UserDAO();
33:               UserBean setUser = dao.getUser(userid);
34:               session.setAttribute("UserBean", setUser);
35:               getServletContext().getRequestDispatcher("/userperson.jsp").forward(
36:                   request, response);
37:             } else {
38:               response.sendRedirect("LogoutServlet");
```

```
39:         }
40:       } else {
41:         response.sendRedirect("login.html");
42:       }
43:     } catch (SystemException e) {
44:       e.printStackTrace();
45:       HttpSession session = request.getSession();
46:       session.setAttribute("Except", e);
47:       getServletContext().getRequestDispatcher("/error.jsp").forward(
48:           request, response);
49:     } catch (DatabaseException e) {
49:       e.printStackTrace();
50:       HttpSession session = request.getSession();
51:       session.setAttribute("Except", e);
52:       getServletContext().getRequestDispatcher("/error.jsp").forward(
53:           request, response);
54:     }
55:   }
56: }
57:
```

ファイル UserUpdateServlet.java

```
01:  package chapter11.user;
02:
03:  import java.io.IOException;
04:
05:  import javax.servlet.ServletException;
06:  import javax.servlet.annotation.WebServlet;
07:  import javax.servlet.http.HttpServlet;
08:  import javax.servlet.http.HttpServletRequest;
09:  import javax.servlet.http.HttpServletResponse;
10:  import javax.servlet.http.HttpSession;
11:
12:  import chapter11.dao.UserDAO;
13:  import chapter11.exception.DatabaseException;
14:  import chapter11.exception.SystemException;
15:
16:  @WebServlet("/UserUpdateServlet")
17:  public class UserUpdateServlet extends HttpServlet {
18:
19:    @Override
20:    protected void doPost(HttpServletRequest request, HttpServletResponse response)
21:        throws ServletException, IOException {
22:
23:       UserBean user = null;
24:
25:      try {
26:        String userid = request.getParameter("userid");
27:        String passwd = request.getParameter("passwd");
28:        String role = request.getParameter("role");
29:
30:        user = new UserBean();
31:        user.setUserid(userid);
32:        user.setPasswd(passwd);
33:        user.setRole(role);
34:
35:        UserDAO dao = new UserDAO();
36:        dao.updateUser(user);
37:
38:        response.sendRedirect("UserListServlet");
39:      } catch (SystemException e) {
40:        e.printStackTrace();
41:        HttpSession session = request.getSession();
42:        session.setAttribute("Except", e);
43:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
44:            response);
45:      } catch (DatabaseException e) {
46:        e.printStackTrace();
47:        HttpSession session = request.getSession();
48:        session.setAttribute("Except", e);
49:        getServletContext().getRequestDispatcher("/error.jsp").forward(request,
```

```
50:           response);
51:       }
52:
53:     }
54:
55:     @Override
56:     protected void doGet(HttpServletRequest request,
57:         HttpServletResponse response) throws ServletException,
58:         IOException {
59:       HttpSession session = request.getSession(false);
60:       if (session != null) {
61:         session.invalidate();
62:       }
63:       response.sendRedirect("login.html");
64:     }
65:   }
```

ファイル error.jsp

```
01:   <%@ page language="java" contentType="text/html; charset=UTF-8"
02:    pageEncoding="UTF-8"%>
03:   <!DOCTYPE html>
04:   <html>
05:   <head>
06:   <meta charset="UTF-8">
07:   <title>エラー</title>
08:   </head>
09:   <body>
10:    <h1>エラー画面</h1>
11:    <%
12:      Exception e = (Exception) session.getAttribute("Except");
13:    %>
14:    <p><%=e.getMessage()%>
15:    <p>
16:     <a href="javascript:history.back();">戻る</a>
17:   </body>
18:   </html>
```

ファイル login.html

```
01:   <!DOCTYPE html>
02:   <html>
03:   <head>
04:   <meta charset="UTF-8">
05:   <title>つぶやきアプリ ログイン</title>
06:   </head>
07:   <body>
08:     <h1>つぶやきアプリ</h1>
09:     <h2>ログイン</h2>
10:     <form action="LoginServlet" method="post">
```

```
11:       <input type="hidden" name="action" value="login">
12:       ログイン ID：<input type="text" name="login_id"><br>
13:        パスワード：<input type="password" name="login_pass"><br>
14:            <input type="submit" value="LOGIN"> <input type="reset">
15:     </form>
16:   </body>
17:   </html>
```

ファイル loginerror.jsp

```
01:   <%@ page language="java" contentType="text/html; charset=UTF-8"
02:    pageEncoding="UTF-8"%>
03:   <!DOCTYPE html>
04:   <html>
05:   <head>
06:   <meta charset="UTF-8">
07:   <title>つぶやきアプリ ログインエラー</title>
08:   </head>
09:   <body>
10:    <h1>つぶやきアプリ</h1>
11:    <h2>ログインエラー</h2>
12:    ログイン ID またはパスワードに誤りがあります。
13:    <br> 再度入力してください。
14:    <br>
15:    <a href="login.html"> ログイン</a>
16:   </body>
17:   </html>
```

ファイル tweetinput.html

```
01:   <!DOCTYPE html>
02:   <html>
03:   <head>
04:   <meta charset="UTF-8">
05:   <title>つぶやきアプリ</title>
06:   </head>
07:   <body>
08:    <h1>つぶやきアプリ</h1>
09:    <form action="TweetInputServlet" method="post">
10:     <p>
11:      名前:<br> <input type="text" name="name" maxlength="50">
12:     </p>
13:     <p>
14:      つぶやき:<br>
15:      <textarea rows="5" cols="40" name="tweet"></textarea>
16:     </p>
17:     <p>
18:      <input type="submit" value="送信"> <input type="reset">
```

```
19:        <input type="button" onclick="location.href='TweetListServlet'"
20:         value="戻る">
21:       </form>
22:      </html>
```

ファイル tweetlist.jsp

```
01:      <%@ page language="java" contentType="text/html; charset=UTF-8"
02:       pageEncoding="UTF-8"%>
03:      <%@ page import="java.util.ArrayList"%>
04:      <%@ page import="java.net.URLEncoder"%>
05:      <%@ page import="java.text.SimpleDateFormat"%>
06:      <%@ page import="chapter11.tweet.*"%>
07:      <%@ page import="chapter11.user.*"%>
08:      <%@ page import="chapter11.parameter.*"%>
09:      <!DOCTYPE html>
10:      <html>
11:      <head>
12:      <meta charset="UTF-8">
13:      <title>つぶやきアプリ</title>
14:      </head>
15:      <body>
16:       <jsp:useBean id="TweetArrayBean" class="chapter11.tweet.TweetArrayBean"
17:      scope="session" />
18:       <jsp:useBean id="LoginUser" class="chapter11.user.UserBean" scope="session" />
19:       <h1>つぶやきアプリ</h1>
20:       <%=LoginUser.getUserid()%>：
21:       <a href="LogoutServlet">ログアウト</a>
22:       <%
23:         if (LoginUser.getRole().equals(UserRoleParameters.ROLE_ADMINS)) {
24:       %>
25:       <a href="UserListServlet">ユーザ管理</a>
26:       <%
27:         }
28:       %>
29:       <br>
30:       <br>
31:       <form action="tweetinput.html" method="get">
32:        <input type="submit" value="書き込み">
33:       </form>
34:       <br>
35:       <table border="1">
36:        <tr>
37:         <th>名前</th>
38:         <th>つぶやき</th>
39:         <th>日時</th>
40:        </tr>
41:        <%
42:          ArrayList<TweetBean> tweetArray = TweetArrayBean.getTweetArray();
43:            for (TweetBean record : tweetArray) {
```

```
44:       %>
45:       <tr>
46:        <td><a
47:      href="TweetPersonServlet?name=<%=URLEncoder.encode(record.getName(),"UTF-8")%>">
48:           <%=record.getName()%>
49:       </a></td>
50:       <td><%=record.getTweet()%></td>
51:       <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
52:                .getDatetime())%></td>
53:      </tr>
54:      <%
55:        }
56:      %>
57:     </table>
58:    </body>
59:    </html>
```

ファイル tweetpersonlist.jsp

```
01:    <%@ page language="java" contentType="text/html; charset=UTF-8"
02:     pageEncoding="UTF-8"%>
03:    <%@ page import="java.util.ArrayList"%>
04:    <%@ page import="java.text.SimpleDateFormat"%>
05:    <%@ page import="chapter11.tweet.*"%>
06:    <%@ page import="chapter11.user.*"%>
07:    <%@ page import="chapter11.parameter.*"%>
08:    <!DOCTYPE html>
09:    <html>
10:    <head>
11:    <meta charset="UTF-8">
12:    <title>つぶやきアプリ</title>
13:    </head>
14:    <body>
15:     <jsp:useBean id="TweetArrayBean" class="chapter11.tweet.TweetArrayBean"
16:    scope="session" />
17:     <jsp:useBean id="LoginUser" class="chapter11.user.UserBean" scope="session" />
18:     <h1>つぶやきアプリ</h1>
19:     <h2><%=request.getParameter("name")%>さんのつぶやき
20:     </h2>
21:     <table>
22:      <tr>
23:       <td><form action="TweetListServlet" method="get">
24:         <input type="submit" value="戻る">
25:        </form></td>
26:       <%
27:       if (LoginUser.getRole().equals(UserRoleParameters.ROLE_ADMINS)) {
28:       %>
29:       <td><form action="TweetDeleteServlet" method="get">
30:         <input type="hidden" name="name"
31:          value="<%=request.getParameter("name")%>"> <input
```

```
32:         type="submit" value="削除">
33:       </form></td>
34:       <%
35:         }
36:       %>
37:      </tr>
38:     </table>
39:     <table border="1">
40:      <tr>
41:       <th>名前</th>
42:       <th>つぶやき</th>
43:       <th>日時</th>
44:      </tr>
45:      <%
46:        ArrayList<TweetBean> tweetArray = TweetArrayBean
47:          .getTweetArray();
48:          for (TweetBean record : tweetArray) {
49:      %>
50:      <tr>
51:       <td><%=record.getName()%></td>
52:       <td><%=record.getTweet()%></td>
53:       <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
54:               .getDatetime())%></td>
55:      </tr>
56:      <%
57:        }
58:      %>
59:     </table>
60:    </body>
61:    </html>
```

ファイル userinput.html

```
01:    <!DOCTYPE html>
02:    <html>
03:    <head>
04:    <meta charset="UTF-8">
05:    <title>つぶやきアプリ</title>
06:    </head>
07:    <body>
08:     <h1>つぶやきアプリ</h1>
09:     <h2>ユーザ作成</h2>
10:     <form action="UserInputServlet" method="post">
11:      <p>
12:       ユーザＩＤ:<input type="text" name="userid" maxlength="50"><br>
13:       パスワード:<input type="password" name="passwd" maxlength="50"><br>
14:       ロール： <input type="radio" name="role" value="users" checked>users
15:       <input type="radio" name="role" value="admins">admins
16:      </p>
17:      <p>
```

```
18:       <input type="submit" value="作成"> <input type="reset">
19:       <input type="button" onclick="location.href='./UserListServlet'"
20:        value="戻る">
21:     </form>
22:    </body>
23:    </html>
```

ファイル userlist.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="chapter11.user.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="UserArrayBean" class="chapter11.user.UserArrayBean" scope="session" />
  <jsp:useBean id="LoginUser" class="chapter11.user.UserBean" scope="session" />
 <h1>つぶやきアプリ</h1>
 <h2>ユーザ管理</h2>
 <%=LoginUser.getUserid()%>：
 <a href="LogoutServlet">ログアウト</a>
 <a href="TweetListServlet">つぶやきへ戻る</a>
 <br>
 <br>
 <table border="1">
  <tr>
   <th>ユーザ ID</th>
   <th>ロール</th>
   <td colspan=2><input type="button"
    onclick="location.href='userinput.html'" value="ユーザ作成"></td>
  </tr>
  <%
    ArrayList<UserBean> userArray = UserArrayBean.getUserArray();
    for (UserBean record : userArray) {
  %>
  <tr>
   <td><%=record.getUserid()%></td>
   <td><%=record.getRole()%></td>
   <td><a href="UserPersonServlet?userid=<%=record.getUserid()%>">編集</a></td>
   <td><a href="UserDeleteServlet?userid=<%=record.getUserid()%>">削除</a></td>
  </tr>
  <%
    }
  %>
 </table>
</body>
</html>
```

ファイル userperson.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
```

```
<%@ page import="chapter11.parameter.*"%>
<%@ page import="chapter11.user.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>つぶやきアプリ</title>
</head>
<body>
 <jsp:useBean id="UserBean" class="chapter11.user.UserBean" scope="session" />
 <jsp:useBean id="LoginUser" class="chapter11.user.UserBean" scope="session" />
 <h1>つぶやきアプリ</h1>
 <h2>ユーザ情報変更</h2>

 <form action="UserUpdateServlet" method="post">
  ユーザ ID:<input type="hidden" name="userid"
   value="<%=UserBean.getUserid()%>"><%=UserBean.getUserid()%><br>
  パスワード : <input type="password" name="passwd"
   value="<%=UserBean.getPasswd()%>"><br>
  <%
    String users = "checked";
    String admins = "";
    if (LoginUser.getRole().equals(UserRoleParameters.ROLE_ADMINS)) {
      users = "";
      admins = "checked";
    }
  %>
  ロール :  <input type="radio" name="role" value="users" <%=users%>>
<%=UserRoleParameters.ROLE_USERS%>
  <input type="radio" name="role" value="admins" <%=admins%>>
<%=UserRoleParameters.ROLE_ADMINS%>
  <br> <input type="submit" value="変更"> <input type="reset">
  <input type="button" onclick="location.href='./UserListServlet'"
   value="戻る">
 </form>
</body>
</html>
```