
Javaバイブルシリーズ Webアプリケーション構築 改訂版

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 rensyu2_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:     private static final long serialVersionUID = 1L;
15:
16:     public Kuku() {
17:         super();
18:     }
19:
20:     @Override
21:     protected void doGet(HttpServletRequest request,
22:         HttpServletResponse response)
23:         throws ServletException, IOException {
24:         PrintWriter out = response.getWriter();
25:         out.println("<html>");
26:         out.println("<title>九九の表</title>");
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></html>");
53: }
54: }

```

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 rensyu2_2

ファイル TenkiServlet.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("/TenkiServlet")
16: public class TenkiServlet extends HttpServlet {
17:     private static final long serialVersionUID = 1L;
18:
19:     /**
20:      * @see HttpServlet#doGet(HttpServletRequest request,
21:      *      HttpServletResponse response)
22:      */
23:     @Override
24:     protected void doGet(HttpServletRequest request,
25:         HttpServletResponse response)
26:         throws ServletException, IOException {
27:

```

```

28:     PrintWriter out = response.getWriter();
29:     out.println("<html>");
30:     out.println("<head>");
31:     out.println("<title>お天気予想</title>");
32:     out.println("</head>");
33:     out.println("<body>");
34:     out.println("<h1>お天気を予想します</h1>");
35:     double tenki = Math.random();
36:     if (tenki <= 0.5d) {
37:         out.println("天気は、晴れです。<p>");
38:     } else if (tenki <= 0.8d) {
39:         out.println("天気は、くもりですねえ。<p>");
40:     } else {
41:         out.println("天気は、雨でしょう。<p>");
42:     }
43:     out.println("</body></html>");
44: }
45: }

```

なお、ExampleFilter.javaを準備します。

練習3 アクセスカウンター

プロジェクト名 rensyu2_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:         private static final long serialVersionUID = 1L;
18:         int accessCount = 0;
19:
20:         /**
21:          * @see HttpServlet#HttpServlet()
22:          */

```

```

23:     public AccessCounterServlet() {
24:         super();
25:
26:     }
27:
28:     /**
29:      * @see HttpServlet#doGet(HttpServletRequest request,
30:      *      HttpServletResponse response)
31:      */
32:     @Override
33:     protected void doGet(HttpServletRequest request,
34:         HttpServletResponse response)
35:         throws ServletException, IOException {
36:
37:         accessCount++;
38:         PrintWriter out = response.getWriter();
39:         out.println("<html>");
40:         out.println("<head>");
41:         out.println("<title>アクセスカウンタ</title>");
42:         out.println("</head>");
43:         out.println("<body>");
44:         out.println("<h1>アクセスカウンター</h1>");
45:         out.println("あなたは、" + accessCount + "人目の訪問者です！<p>");
46:         out.println("</body>");
47:         out.println("</html>");
48:     }
49: }

```

練習4 数当て

プロジェクト名 rensyu2_4

ファイル KazuateServlet.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 KazuateServlet
14:      */
15:     @WebServlet("/KazuateServlet")

```

```

16: public class KazuateServlet extends HttpServlet {
17:     private static final long serialVersionUID = 1L;
18:     int targetKazu = 0;
19:     int kaisu = 0;
20:
21:     /**
22:      * @see HttpServlet#doGet(HttpServletRequest request,
23:      *      HttpServletResponse response)
24:      */
25:     @Override
26:     protected void doGet(HttpServletRequest request,
27:         HttpServletResponse response)
28:         throws ServletException, IOException {
29:
30:         if (kaisu == 0) {
31:             targetKazu = (int) (Math.random() * 9d) + 1;
32:         }
33:         kaisu++;
34:         String kazu_s = request.getParameter("kazu");
35:         int enteredKazu = Integer.parseInt(kazu_s);
36:
37:         String kekka = "はずれ～！";
38:         if (enteredKazu == targetKazu) {
39:             kekka = "当たり";
40:         }
41:
42:         PrintWriter out = response.getWriter();
43:         out.println("<html>");
44:         out.println("<head>");
45:         out.println("<title>数当てゲーム結果</title>");
46:         out.println("</head>");
47:         out.println("<body>");
48:         out.println("<h1>数当てゲーム結果</h1>");
49:         out.println(kaisu + "回目 あなたが選んだ数：" + enteredKazu + "<br>");
50:         out.println("<br>");
51:         out.println("結果は、" + kekka);
52:         out.println("<p><a href='\"kazuate.html\"'>もう一度やる</a>");
53:         out.println("</body></html>");
54:
55:         if (enteredKazu == targetKazu) {
56:             kaisu = 0;
57:         }
58:
59:     }
60: }

```

ファイル kazuate.html

```

01:     <!DOCTYPE html>
02:

```

```

03: <html lang="ja">
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: <p>
11: コンピュータが考えている数字（1～9）を当てましょう。
12: <p>
13: <a href="KazuatoServlet?kazu=1">1</a>&nbsp;
14: <a href="KazuatoServlet?kazu=2">2</a>&nbsp;
15: <a href="KazuatoServlet?kazu=3">3</a>&nbsp;
16: <a href="KazuatoServlet?kazu=4">4</a>&nbsp;
17: <a href="KazuatoServlet?kazu=5">5</a>&nbsp;
18: <a href="KazuatoServlet?kazu=6">6</a>&nbsp;
19: <a href="KazuatoServlet?kazu=7">7</a>&nbsp;
20: <a href="KazuatoServlet?kazu=8">8</a>&nbsp;
21: <a href="KazuatoServlet?kazu=9">9</a>&nbsp;
22: </body>
23: </html>

```

練習5 単純な電卓

プロジェクト名 rensyu2_5

ファイル KeisanServlet.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("/KeisanServlet")
16: public class KeisanServlet extends HttpServlet {
17:     private static final long serialVersionUID = 1L;
18:
19:     /**
20:      * @see HttpServlet#HttpServlet()
21:      */

```

```

22:     public KeisanServlet() {
23:         super();
24:
25:     }
26:
27:     /**
28:      * @see HttpServlet#doGet(HttpServletRequest request,
29:      *      HttpServletResponse response)
30:      */
31:     @Override
32:     protected void doGet(HttpServletRequest request,
33:         HttpServletResponse response)
34:         throws ServletException, IOException {
35:
36:         String strKazu1 = request.getParameter("kazu1");
37:         int kazu1 = Integer.parseInt(strKazu1);
38:         String strKazu2 = request.getParameter("kazu2");
39:         int kazu2 = Integer.parseInt(strKazu2);
40:         String syori = request.getParameter("syurui");
41:         PrintWriter out = response.getWriter();
42:         out.println("<html>");
43:         out.println("<head><title>四則演算結果</title></head>");
44:         out.println("<body>");
45:         out.println("<h1>四則演算の結果</h1>");
46:         out.println("<blockquote><table>");
47:         out.println("<tr><td>数字 1 つ目</td><td>  </td><td>数字 2 つ目</td><td>  </td><td>
48:         結果</td></tr>");
49:         out.println("<tr style=¥\"text-align: center¥\"><td> + kazu1 + \"</td><td>");
50:         switch (syori) {
51:             case "tasu": out.print("+");break;
52:             case "hiku": out.print("-");break;
53:             case "kake": out.print("×");break;
54:             case "wari": out.print("÷");break;
55:         }
56:         out.println("</td><td> + kazu2 + \"</td><td>=</td><td>");
57:         switch (syori) {
58:             case "tasu":out.print(kazu1 + kazu2);break;
59:             case "hiku":out.print(kazu1 - kazu2);break;
60:             case "kake":out.print(kazu1 * kazu2);break;
61:             case "wari":out.print(kazu1 / kazu2);break;
62:         }
63:         out.println("</td></tr></table></blockquote>");
64:         out.println("</body></html>");
65:     }
66:
67: }

```

ファイル keisan.html

```

01:     <!DOCTYPE html>

```



```

02:
03: <html lang="ja">
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: <blockquote>
10: <form action="KeisanServlet" method="get">
11:   <table>
12:     <tr><td style="text-align: center">数字 1 つ目<br>
13:       <input size="5" type="text" name="kazu1" value="30"></td>
14:       <td> <input type="radio" name="syurui" value="tasu" checked>+<br>
15:       <input type="radio" name="syurui" value="hiku">-<br>
16:       <input type="radio" name="syurui" value="kake">x<br>
17:       <input type="radio" name="syurui" value="wari">÷</td>
18:     <td style="text-align: center">数字 2 つ目<br>
19:       <input size="5" type="text" name="kazu2" value="20"></td>
20:     <td style="text-align: center">
21:       <input type="submit" value="="></td>
22:   </tr>
23: </table>
24: </form>
25: </blockquote>
26: </body>
27: </html>

```

練習6ー1 じゃんけん

プロジェクト名 rensyu2_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: private static final long serialVersionUID = 1L;
18: /**
19:  * @see HttpServlet#doGet(HttpServletRequest request,
20:  *      HttpServletResponse response)
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 teHito = request.getParameter("te");
31:
32:     /**
33:      * コンピュータの手を決める
34:      */
35:     double jankenPon = Math.random();
36:     String tePGM = "";
37:     if (jankenPon <= 0.3d) {
38:         tePGM = "g";
39:     } else if (jankenPon <= 0.6d) {
40:         tePGM = "c";
41:     } else {
42:         tePGM = "p";
43:     }
44:
45:     /**
46:      * 勝負の判定
47:      */
48:     String kekka = "";
49:     if (tePGM.equals(teHito)) {
50:         kekka = "あいこ";
51:     } else if ((tePGM.equals("g")) && (teHito.equals("c"))) {
52:         kekka = "PGM";
53:     } else if ((tePGM.equals("c")) && (teHito.equals("p"))) {
54:         kekka = "PGM";
55:     } else if ((tePGM.equals("p")) && (teHito.equals("g"))) {
56:         kekka = "PGM";
57:     } else {
58:         kekka = "人";
59:     }
60:     if (kekka.equals("あいこ")){
61:         kekka = "引き分け";
62:     }else if (kekka.equals("PGM")){
63:         kekka = "プログラムの勝ち";
64:     }else{
65:         kekka = "あなたが勝ち";
66:     }

```

```

67:
68:     PrintWriter out = response.getWriter();
69:     out.println("<html>");
70:     out.println("<head>");
71:     out.println("<title>じゃんけん結果</title>");
72:     out.println("</head>");
73:     out.println("<body>");
74:     out.println("<h1>じゃんけん結果</h1>");
75:     out.println("あなたが選んだ手 : " + teHenkan(teHito) + "<br>");
76:     out.println("プログラムが選んだ手 : " + teHenkan(tePGM) + "<br>");
77:     out.println("勝負の結果は、" + kekka + "でした！");
78:     out.println("<p><a href=¥\"janken.html¥\">もう一度やる</a>");
79:     out.println("</body></html>");
80: }
81:
82: String teHenkan(String te) {
83:     switch (te) {
84:         case "g":
85:             return "グー";
86:         case "c":
87:             return "チョキ";
88:         default:
89:             return "パー";
90:     }
91:
92: }
}

```

ファイル janken.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</title>
07:     </head>
08:     <body>
09:     <H1>じゃんけんゲーム</H1>
10:     <p>あなたの選ぶ手は
11:     <p><a href="JankenServlet?te=g">グー</a>
12:     <p><a href="JankenServlet?te=c">チョキ</a>
13:     <p><a href="JankenServlet?te=p">パー</a>
14:     </body>
15:     </html>

```

ファイル JankenServlet2.java

```
001: package pac1;
002:
003: import java.io.IOException;
004: import java.io.PrintWriter;
005:
006: import javax.servlet.ServletException;
007: import javax.servlet.annotation.WebServlet;
008: import javax.servlet.http.HttpServlet;
009: import javax.servlet.http.HttpServletRequest;
010: import javax.servlet.http.HttpServletResponse;
011:
012: /**
013:  * Servlet implementation class JankenServlet
014:  */
015: @WebServlet("/JankenServlet2")
016: public class JankenServlet2 extends HttpServlet {
017:     private static final long serialVersionUID = 1L;
018:     int kaisu = 0;
019:     int winPGM = 0;
020:     int winHito = 0;
021:
022:     /**
023:      * @see HttpServlet#doGet(HttpServletRequest request,
024:      *      HttpServletResponse response)
025:      */
026:     @Override
027:     protected void doGet(HttpServletRequest request,
028:         HttpServletResponse response)
029:         throws ServletException, IOException {
030:
031:         /**
032:          * 人が選んだ手を受け取る
033:          */
034:         String teHito = request.getParameter("te");
035:
036:         /**
037:          * コンピュータの手を決める
038:          */
039:         double jankenPon = Math.random();
040:         String tePGM = "";
041:         if (jankenPon <= 0.3d) {
042:             tePGM = "g";
043:         } else if (jankenPon <= 0.6d) {
044:             tePGM = "c";
045:         } else {
046:             tePGM = "p";
047:         }
```

```

048:
049:     kaisu++;
050:     /**
051:      * 勝負の判定
052:      */
053:     String kekka = "";
054:     if (tePGM.equals(teHito)) {
055:         kekka = "あいこ";
056:     } else if ((tePGM.equals("g")) && (teHito.equals("c"))) {
057:         kekka = "PGM";
058:     } else if ((tePGM.equals("c")) && (teHito.equals("p"))) {
059:         kekka = "PGM";
060:     } else if ((tePGM.equals("p")) && (teHito.equals("g"))) {
061:         kekka = "PGM";
062:     } else {
063:         kekka = "人";
064:     }
065:     if (kekka.equals("あいこ")) {
066:         kekka = "引き分け";
067:     } else if (kekka.equals("PGM")) {
068:         winPGM++;
069:         kekka = "プログラムの勝ち";
070:     } else {
071:         winHito++;
072:         kekka = "あなたが勝ち";
073:     }
074:
075:     PrintWriter out = response.getWriter();
076:     out.println("<html>");
077:     out.println("<head>");
078:     out.println("<title>じゃんけん結果</title>");
079:     out.println("</head>");
080:     out.println("<body>");
081:     out.println("<h1>じゃんけん結果 (" + kaisu + "回目) </h1>");
082:     out.println("あなたが選んだ手 : " + teHenkan(teHito) + "("
083:         + winHito + "勝)<br>");
084:     out.println("プログラムが選んだ手 : " + teHenkan(tePGM) + "("
085:         + winPGM + "勝)<br>");
086:     out.println("勝負の結果は、" + kekka + "でした!");
087:     out.println("<p><a href=¥\"janken2.html¥\">もう一度やる</a>");
088:     out.println("</body></html>");
089: }
090:
091: String teHenkan(String te) {
092:     switch (te) {
093:         case "g":
094:             return "グー";
095:         case "c":
096:             return "チョキ";
097:         default:

```

```
098:         return "/パー";
099:     }
100:
101: }
}
```

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: <p>あなたの選ぶ手は
11: <p><a href="JankenServlet2?te=g">グー</a>
12: <p><a href="JankenServlet2?te=c">チヨキ</a>
13: <p><a href="JankenServlet2?te=p">パー</a>
14: </body>
15: </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 rensyu3_1

ファイル gradientjsp1.jsp

```
01: <%@ page language="java" contentType="text/html; charset=UTF-8"
02:     pageEncoding="UTF-8" %>
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
06: <title>グラデーション JSP その1</title>
07: </head>
08: <body>
09: <% int add = 16; %>
10:
11: <h1>グラデーション JSP その1</h1>
12: グラデーションの幅 : <%= add %><p>
13:
14: <table border="1">
15:   <tr>
16:     <%
17:       for (int color = 0; color <= 255; color += add) {
18:         String colorHex = String.format("%02x", color);
19:                                     // color を 16 進数(2 ケタ)に変換
20:         String colorHTML = colorHex + colorHex + colorHex;
21:                                     // HTML の色指定形式の作成
22:         out.println("   <td style=\"%height: 30px; width: 30px; "
23:           + "background-color: #"
24:           + colorHTML + "\"><br></td>");
25:       }
26:     %>
27:   </tr>
28: </table>
29: </body>
30: </html>
```

プロジェクト名 rensyu3_2

ファイル gradientjsp2.jsp

```
01:    <%@ page language="java" contentType="text/html; charset=UTF-8"
02:        pageEncoding="UTF-8"%>
03:    <html>
04:    <head>
05:    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
06:    <title>グラデーション JSP その2</title>
07:    </head>
08:    <body>
09:
10:    <% int add = 42; %>
11:
12:    <h1>グラデーション JSP その2</h1>
13:    グラデーションの幅 : <%= add %><p>
14:    <table border="1">
15:    <%
16:        int color;
17:
18:        for(int i=0; i<4; i++){
19:            out.println(" <tr>");
20:            color = i*add;
21:
22:            for (int j=0; j<4; j++) {
23:                String colorHex = String.format("%02x", color);
24:                // color を 16 進数(2 ケタ)に変換
25:                String colorHTML = colorHex + colorHex + colorHex;
26:                // HTML の色指定形式の作成
27:                out.println(" <td style=¥"height: 50px; width: 50px; background-color: #"
28:                    + colorHTML + "¥"><br></td>");
29:                color += add;
30:            }
31:
32:            out.println(" </tr>");
33:        }
34:    %>
35:    </table>
36:
37:    </body>
38:    </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 rensyu4_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:     private static final long serialVersionUID = 1L;
19:
20:     /**
21:      * @see HttpServlet#doGet(HttpServletRequest request,
22:      *      HttpServletResponse response)
23:      */
24:     @Override
25:     protected void doGet(HttpServletRequest request,
26:         HttpServletResponse response) throws ServletException,
27:         IOException {
28:         PrintWriter out = response.getWriter();
29:
30:         out.println("<html>");
31:         out.println("<head><title>クッキー情報</title></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>" + name + "さん");
52:     out.println("<p>こんにちは" + count + "回目の訪問ですね");
53: } else {
54:     out.println("<p>初めての訪問ですね。");
55:     out.println("<p>My 情報を登録しましょう");
56:     out.println("<p><a href=¥\"cookieinfo.html¥\">登録</a>");
57: }
58: out.println("</body></html>");
59: }
60:
61: }

```

ファイル WriteCookieServlet

```

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:     private static final long serialVersionUID = 1L;
19:
20:     /**
21:      * @see HttpServlet#doPost(HttpServletRequest request,
22:      *      HttpServletResponse response)
23:      */
24:     @Override
25:     protected void doPost(HttpServletRequest request,
26:         HttpServletResponse response) throws ServletException,
27:         IOException {

```

```

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

```

ファイル cookieinfo.html

```

01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="UTF-8">
05: <title>クッキー情報登録</title>
06: </head>
07: <body>
08: <h1>My 情報登録</h1>
09: <form action="WriteCookieServlet" method="post">
10:     MyNo <input type="text" name="myno">
11:     MyName <input type="text" name="myname">
12:     MyMail <input type="text" name="mymail">
13:     <input type="submit" value="登録">
14: </form>
15: </body>
16: </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 rensyu5_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 static final long serialVersionUID = 1L;
08:     private int payment;    // 入金額
09:     private String comment; // コメント
10:
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 static final long serialVersionUID = 1L;
09:     private ArrayList<PaymentRecordBean> paymentHistoryArray
10:         = new ArrayList<PaymentRecordBean>(); // 貯金の履歴
11:
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 rensyu5_1

ファイル pettycashbook_input.html

```

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

```

```

18:
19:     </body>
20: </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><th>履歴</th><th>貯金額</th><th>コメント</th></tr>
19:     <%
20:     ArrayList<PaymentRecordBean> paymentHistoryArray =
21:         pettyCashbookBean.getPaymentHistoryArray();
22:
23:     int i = 1;
24:     for(PaymentRecordBean paymentRec : paymentHistoryArray){
25:         out.println("<tr><td>" + i + "回目</td><td>"
26:             + paymentRec.getPayment() + "円</td><td>"
27:             + paymentRec.getComment() + "</td></tr>");
28:         i++;
29:     }
30:     %>
31:     </table>
32:
33:     <br>
34:     <b>合計金額 : <%= pettyCashbookBean.getTotalPayment() %>円</b>
35:     <p><br>
36:     <a href="pettycashbook_input.html">さらに入金する</a><br>
37:
38:     </body>
39: </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: @WebServlet("/PettyCashbookServlet")
13: public class PettyCashbookServlet extends HttpServlet {
14:
15:     private static final long serialVersionUID = 1L;
16:     PettyCashbookBean pettyCashbookBean = new PettyCashbookBean();
17:
18:     public PettyCashbookServlet() {
19:         super();
20:     }
21:
22:     @Override
23:     protected void doGet(HttpServletRequest request,
24:         HttpServletResponse response) throws ServletException,
25:         IOException {
26:
27:         int payment;    // 入金額
28:         String comment; // コメント
29:
30:         /* Formから入力された値の取り出し */
31:         try {
32:             payment = Integer.parseInt(request.getParameter("payment"));
33:             comment = request.getParameter("comment");
34:
35:         } catch (NumberFormatException e) {
36:             getServletContext().getRequestDispatcher("/errorinput.html")
37:                 .forward(request, response);
38:             return;
39:         }
40:
41:         /* 入金レコードの作成 */
42:         PaymentRecordBean paymentRec = new PaymentRecordBean();
43:         paymentRec.setPayment(payment);
44:         paymentRec.setComment(comment);
45:
46:         /* Beans 実行 */
47:         pettyCashbookBean.addPaymentHistoryArray(paymentRec); // 入金レコードを Beans へ追加
48:
49:         /* セッションに Beans を登録し、JSP で出力 */
50:         HttpSession session = request.getSession();
51:         session.setAttribute("pettyCashbookBean", pettyCashbookBean);
52:         getServletContext().getRequestDispatcher(

```

```

53:         "/pettycashbook_output.jsp").forward(request, response);
54:
55:     }
56: }

```

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 rensyu6_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.SQLException;
09: import java.sql.Statement;
10: import java.util.ArrayList;
11:
12: import javax.servlet.ServletException;
13: import javax.servlet.annotation.WebServlet;
14: import javax.servlet.http.HttpServlet;
15: import javax.servlet.http.HttpServletRequest;
16: import javax.servlet.http.HttpServletResponse;
17:
18: @WebServlet("/SelectServlet1")
19: public class SelectServlet1 extends HttpServlet {
20:     private static final long serialVersionUID = 1L;
21:
22:     public SelectServlet1() {
23:         super();
24:     }
25:
26:     @Override
27:     protected void doGet(HttpServletRequest request,
28:         HttpServletResponse response)
29:         throws ServletException, IOException {
30:         Connection conn = null;
31:         Statement stmt = null;
32:         ResultSet rs = null;
33:         ArrayList<String> list = new ArrayList<String>();
34:
35:         try {
36:             Class.forName("com.mysql.jdbc.Driver");

```



```

37:         conn = DriverManager.getConnection(
38:             "jdbc:mysql://localhost/sampledб", "user", "password");
39:         stmt = conn.createStatement();
40:         rs = stmt.executeQuery("SELECT * FROM SUBJECT");
41:
42:         while (rs.next()) {
43:             String s = "<td>" + rs.getInt("SUBJECTID") + "</td>" + "<td>"
44:                 + rs.getString("SUBJECTNAME") + "</td>";
45:             list.add(s);
46:         }
47:
48:     } catch (Exception e) {
49:         e.printStackTrace();
50:     } finally {
51:         if (rs != null) {
52:             try {
53:                 rs.close();
54:             } catch (SQLException sqlEx) {
55:             }
56:         }
57:         if (stmt != null) {
58:             try {
59:                 stmt.close();
60:             } catch (SQLException sqlEx) {
61:             }
62:         }
63:         if (conn != null) {
64:             try {
65:                 conn.close();
66:             } catch (SQLException sqlEX) {
67:             }
68:         }
69:     }
70:
71:     PrintWriter out = response.getWriter();
72:
73:     out.println("<html>");
74:     out.println("<head><title>" + "練習問題 6-1" + "</title></head>");
75:     out.println("<body>");
76:     out.println("<table border=¥\"1¥>");
77:     out.println("<tr><th>科目 ID</th>" + "<th>科目名</th></tr>");
78:
79:     for (String str : list) {
80:         out.println("<tr>" + str + "</tr>");
81:     }
82:
83:     out.println("</table>");
84:     out.println("</body>");
85:     out.println("</html>");
86: }

```

```
87:
88: }
```

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 rensyu6_2

ファイル SelectServlet2.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.SQLException;
09: import java.sql.Statement;
10: import java.util.ArrayList;
11:
12: import javax.servlet.ServletException;
13: import javax.servlet.annotation.WebServlet;
14: import javax.servlet.http.HttpServlet;
15: import javax.servlet.http.HttpServletRequest;
16: import javax.servlet.http.HttpServletResponse;
17:
18: @WebServlet("/SelectServlet2")
19: public class SelectServlet2 extends HttpServlet {
20:     private static final long serialVersionUID = 1L;
21:
22:     public SelectServlet2() {
23:         super();
24:     }
25:
26:     @Override
27:     protected void doGet(HttpServletRequest request,
28:         HttpServletResponse response)
29:         throws ServletException, IOException {
30:         Connection conn = null;
31:         Statement stmt = null;
32:         ResultSet rs = null;
33:         ArrayList<String> list = new ArrayList<String>();
34:
35:         try {
36:             Class.forName("com.mysql.jdbc.Driver");
37:             conn = DriverManager.getConnection(
38:                 "jdbc:mysql://localhost/sampled", "user", "password");
39:             stmt = conn.createStatement();
40:             rs = stmt
```

```

41:         .executeQuery("SELECT SUBJECTNAME,AVG(SCORE) FROM GRADE,SUBJECT
42: WHERE GRADE.SUBJECTID = SUBJECT.SUBJECTID GROUP BY SUBJECTNAME");
43:
44:         while (rs.next()) {
45:             String s = "<td>" + rs.getString("SUBJECTNAME") + "</td>"
46:                 + "<td>" + rs.getDouble("AVG(SCORE)") + "</td>";
47:             list.add(s);
48:         }
49:
50:     } catch (Exception e) {
51:         e.printStackTrace();
52:     } finally {
53:         if (rs != null) {
54:             try {
55:                 rs.close();
56:             } catch (SQLException sqlEx) {
57:             }
58:         }
59:         if (stmt != null) {
60:             try {
61:                 stmt.close();
62:             } catch (SQLException sqlEx) {
63:             }
64:         }
65:         if (conn != null) {
66:             try {
67:                 conn.close();
68:             } catch (SQLException sqlEX) {
69:             }
70:         }
71:     }
72:
73:     PrintWriter out = response.getWriter();
74:
75:     out.println("<html>");
76:     out.println("<head><title>" + "練習問題 6-2" + "</title></head>");
77:     out.println("<body>");
78:     out.println("<table border=¥\"1¥\">");
79:     out.println("<tr><th>科目名</th>" + "<th>平均点</th></tr>");
80:
81:     for (String str : list) {
82:         out.println("<tr>" + str + "</tr>");
83:     }
84:
85:     out.println("</table>");
86:     out.println("</body>");
87:     out.println("</html>");
88: }
89:
90: }

```

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 rensyu6_3

ファイル index.html

```
01: <!DOCTYPE html>
02:
03: <html lang="ja">
04: <head>
05: <meta charset="UTF-8">
06: <title>練習 6-3</title>
07: </head>
08: <body>
09: <h1>練習 6-3</h1>
10: <h2>科目テーブルに新規科目を登録します。</h2>
11: <form action="InsertServlet1" method="get">
12:   <div>
13:     科目 ID : <input type="text" name="subjectid">
14:   </div>
15:   <div>
16:     科目名 : <input type="text" name="subjectname">
17:   </div>
18:   <div>
19:     <input type="submit" value="登録"><input type="reset">
20:   </div>
21: </form>
22: </body>
23: </html>
```

ファイル InsertServlet1.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.SQLException;
08: import java.sql.Statement;
09:
10: import javax.servlet.ServletException;
11: import javax.servlet.annotation.WebServlet;
12: import javax.servlet.http.HttpServlet;
13: import javax.servlet.http.HttpServletRequest;
14: import javax.servlet.http.HttpServletResponse;
15:
16: @WebServlet("/InsertServlet1")
```

```

17: public class InsertServlet1 extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public InsertServlet1() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request,
26:         HttpServletResponse response)
27:         throws ServletException, IOException {
28:         Connection conn = null;
29:         Statement stmt = null;
30:         int ret = 0;
31:         int id;
32:         String name;
33:
34:         try {
35:             id = Integer.parseInt(request.getParameter("subjectid"));
36:         } catch (NumberFormatException e) {
37:             return;
38:         }
39:         name = request.getParameter("subjectname");
40:
41:         try {
42:             Class.forName("com.mysql.jdbc.Driver");
43:             conn = DriverManager.getConnection(
44:                 "jdbc:mysql://localhost/sampledbs", "user", "password");
45:             stmt = conn.createStatement();
46:             ret = stmt.executeUpdate("INSERT INTO SUBJECT VALUES(" + id + ",¥'"
47:                 + name + "¥'");
48:
49:         } catch (Exception e) {
50:             e.printStackTrace();
51:         } finally {
52:             if (stmt != null) {
53:                 try {
54:                     stmt.close();
55:                 } catch (SQLException sqlEx) {
56:                 }
57:             }
58:             if (conn != null) {
59:                 try {
60:                     conn.close();
61:                 } catch (SQLException sqlEX) {
62:                 }
63:             }
64:         }
65:
66:         PrintWriter out = response.getWriter();

```

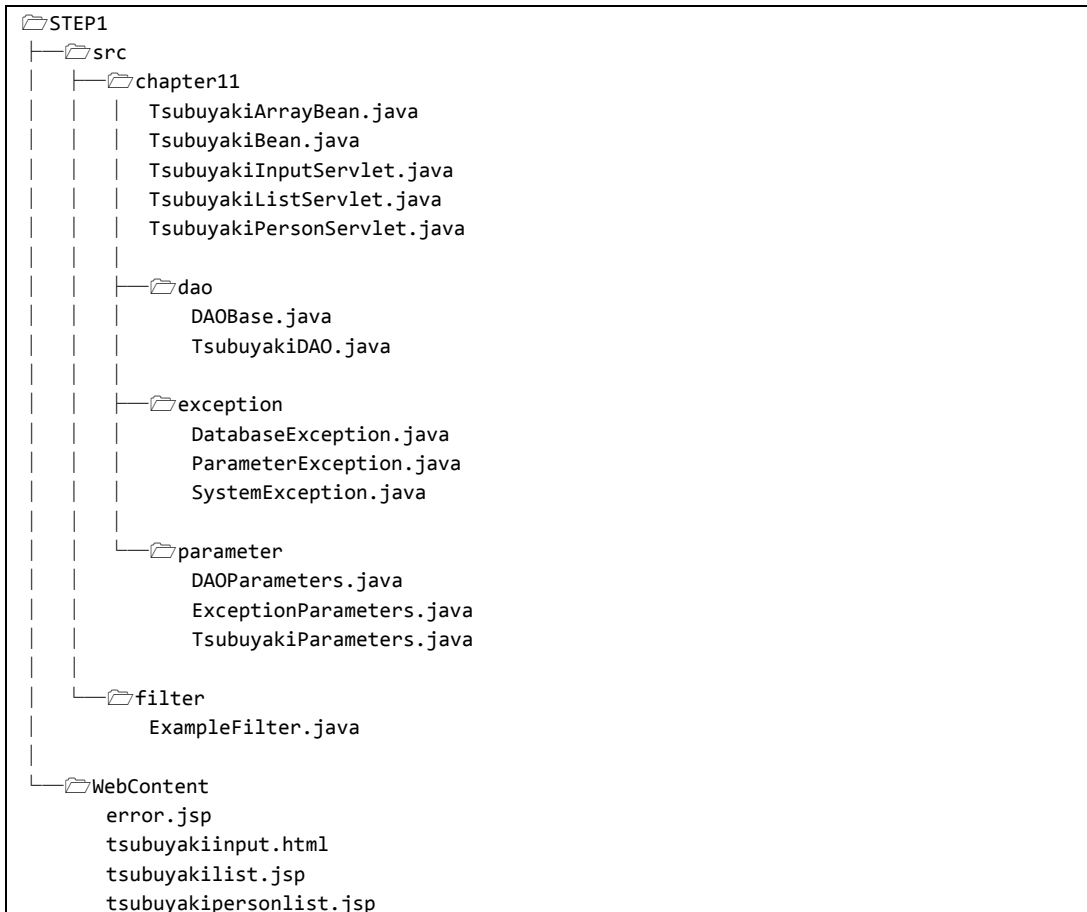
```
67:
68:         out.println("<html>");
69:         out.println("<head><title>練習 6-3</title></head>");
70:         out.println("<body>");
71:         out.println("<h1>練習 6-3</h1>");
72:         out.println("<h2>" + ret + "件追加しました。</h2>");
73:         out.println("<p>");
74:         out.println("科目 ID : " + id + "<br>");
75:         out.println("科目名 : " + name + "<br>");
76:         out.println("</body>");
77:         out.println("</html>");
78:     }
79:
80: }
```

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



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。

SQL(MySQL 用)

```
01: CREATE DATABASE TSUBUYAKI;
02: USE TSUBUYAKI;
03:
04: CREATE TABLE TSUBUYAKI (
05:     NAME VARCHAR(50) NOT NULL,
06:     TSUBUYAKI VARCHAR(250),
07:     DATE DATETIME
08: );
09:
10: GRANT ALL PRIVILEGES ON TSUBUYAKI.*
11: TO user@localhost IDENTIFIED BY 'password';
```

ファイル TsubuyakiArrayBean.java

```
01: package chapter11;
02:
03: import java.io.Serializable;
04: import java.util.ArrayList;
05:
06: public class TsubuyakiArrayBean implements Serializable {
07:
08:     private static final long serialVersionUID = 1L;
09:
10:     private ArrayList<TsubuyakiBean> tsubuyakiArray;
11:
12:     public TsubuyakiArrayBean() {
13:         tsubuyakiArray = new ArrayList<TsubuyakiBean>();
14:     }
15:
16:     public void addTsubuyaki(TsubuyakiBean obj) {
17:         tsubuyakiArray.add(obj);
18:     }
19:
20:     public int getArraySize() {
21:         return tsubuyakiArray.size();
22:     }
23:
24:     public ArrayList<TsubuyakiBean> getTsubuyakiArray() {
25:         return tsubuyakiArray;
26:     }
27:
28:     public void setTsubuyakiArray(ArrayList<TsubuyakiBean> tsubuyakiArray) {
29:         this.tsubuyakiArray = tsubuyakiArray;
30:     }
31:
32: }
```


ファイル TsubuyakiBean.java

```
01: package chapter11;
02:
03: import java.io.Serializable;
04: import java.util.Date;
05:
06: public class TsubuyakiBean implements Serializable {
07:
08:     private static final long serialVersionUID = 1L;
09:
10:     private String name;
11:     private String tsubuyaki;
12:     private Date datetime;
13:
14:     public TsubuyakiBean() {
15:     }
16:
17:     public String getName() {
18:         return name;
19:     }
20:
21:     public void setName(String name) {
22:         this.name = name;
23:     }
24:
25:     public String getTsubuyaki() {
26:         return tsubuyaki;
27:     }
28:
29:     public void setTsubuyaki(String tsubuyaki) {
30:         this.tsubuyaki = tsubuyaki;
31:     }
32:
33:     public Date getDatetime() {
34:         return datetime;
35:     }
36:
37:     public void setDatetime(Date datetime) {
38:         this.datetime = datetime;
39:     }
40:
41: }
```

ファイル TsubuyakiInputServlet.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.TsubuyakiDAO;
14: import chapter11.exception.DatabaseException;
15: import chapter11.exception.SystemException;
16:
17: @WebServlet("/TsubuyakiInputServlet")
18: public class TsubuyakiInputServlet extends HttpServlet {
19:     private static final long serialVersionUID = 1L;
20:
21:     public TsubuyakiInputServlet() {
22:         super();
23:     }
24:
25:     @Override
26:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
27:         throws ServletException, IOException {
28:         try {
29:             String name = request.getParameter("name");
30:             String tsubuyaki = request.getParameter("tsubuyaki");
31:             Date date = new Date();
32:
33:             TsubuyakiBean tsubuyakiBean = new TsubuyakiBean();
34:             tsubuyakiBean.setName(name);
35:             tsubuyakiBean.setTsubuyaki(tsubuyaki);
36:             tsubuyakiBean.setDatetime(date);
37:
38:             TsubuyakiDAO dao = new TsubuyakiDAO();
39:             dao.setTsubuyaki(tsubuyakiBean);
40:
41:             response.sendRedirect("TsubuyakiListServlet");
42:         } catch (SystemException e) {
43:             e.printStackTrace();
44:             HttpSession session = request.getSession();
45:             session.setAttribute("Except", e);
46:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:                 response);
48:         } catch (DatabaseException e) {
49:             e.printStackTrace();
50:         }
```

```

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

```

ファイル TsubuyakiListServlet.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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiListServlet")
17: public class TsubuyakiListServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiListServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         TsubuyakiArrayBean tsubuyakiArray;
29:
30:         try {
31:             TsubuyakiDAO dao = new TsubuyakiDAO();
32:             tsubuyakiArray = dao.getTsubuyakiArray();
33:             HttpSession session = request.getSession();
34:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
35:             getServletContext().getRequestDispatcher("/tsubuyakilist.jsp").forward(
36:                 request, response);
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 TsubuyakiPersonServlet.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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiPersonServlet")
17: public class TsubuyakiPersonServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiPersonServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         TsubuyakiArrayBean tsubuyakiArray;
29:
30:         try {
31:             String name = request.getParameter("name");
32:
33:             TsubuyakiDAO dao = new TsubuyakiDAO();
34:             tsubuyakiArray = dao.getPersonTsubuyakiArray(name);
35:             HttpSession session = request.getSession();

```

```

36:         session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
37:         getServletContext().getRequestDispatcher("/tsubuyakipersonlist.jsp")
38:             .forward(request, response);
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: }

```

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 TsubuyakiDAO.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.TsubuyakiArrayBean;
10:     import chapter11.TsubuyakiBean;
11:     import chapter11.exception.DatabaseException;
12:     import chapter11.exception.SystemException;
13:     import chapter11.parameter.ExceptionParameters;
14:     import chapter11.parameter.TsubuyakiParameters;
15:
16:     public class TsubuyakiDAO extends DAOBase {
17:
18:         public TsubuyakiArrayBean getTsubuyakiArray() throws DatabaseException,
19:             SystemException {
20:             Statement stmt = null;
21:             TsubuyakiArrayBean tsubuyakiArray = new TsubuyakiArrayBean();
22:             this.open();
23:             try {
24:                 stmt = con.createStatement();
25:                 ResultSet rs = stmt.executeQuery(TsubuyakiParameters.SQL_SELECT_ALL);
26:                 while (rs.next()) {
27:                     TsubuyakiBean record = new TsubuyakiBean();
28:                     record.setName(rs.getString(TsubuyakiParameters.NAME));
29:                     record.setTsubuyaki(rs.getString(TsubuyakiParameters.TSUBUYAKI));
30:                     record.setDatetime(rs.getTimestamp(TsubuyakiParameters.DATE));
31:                     tsubuyakiArray.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 tsubuyakiArray;
40: }
41:
42: public TsubuyakiArrayBean getPersonTsubuyakiArray(String name)
43:   throws DatabaseException, SystemException {
44:   PreparedStatement stmt = null;
45:   TsubuyakiArrayBean tsubuyakiArray = new TsubuyakiArrayBean();
46:   this.open();
47:   try {
48:     stmt = con.prepareStatement(TsubuyakiParameters.SQL_SELECT_PERSON);
49:     stmt.setString(1, name);
50:     ResultSet rs = stmt.executeQuery();
51:     while (rs.next()) {
52:       TsubuyakiBean record = new TsubuyakiBean();
53:       record.setName(rs.getString(TsubuyakiParameters.NAME));
54:       record.setTsubuyaki(rs.getString(TsubuyakiParameters.TSUBUYAKI));
55:       record.setDatetime(rs.getTimestamp(TsubuyakiParameters.DATE));
56:       tsubuyakiArray.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 tsubuyakiArray;
65: }
66:
67: public void setTsubuyaki(TsubuyakiBean tsubuyaki) throws DatabaseException,
68:   SystemException {
69:   PreparedStatement stmt = null;
70:   this.open();
71:   try {
72:     stmt = con.prepareStatement(TsubuyakiParameters.SQL_INSERT);
73:     stmt.setString(1, tsubuyaki.getName());
74:     stmt.setString(2, tsubuyaki.getTsubuyaki());
75:     stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
76:       .format(tsubuyaki.getDatetime()));
77:     stmt.executeUpdate();
78:   } catch (SQLException e) {
79:     throw new DatabaseException(
80:       ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
81:   } finally {

```

```

83:         this.close(stmt);
84:     }
85: }
86: }

```

ファイル DatabaseException.java

```

01: package chapter11.exception;
02:
03: public class DatabaseException extends Exception {
04:     private static final long serialVersionUID = 1L;
05:
06:     public DatabaseException() {
07:         super();
08:     }
09:
10:     public DatabaseException(String message, Throwable cause,
11:         boolean enableSuppression, boolean writableStackTrace) {
12:         super(message, cause, enableSuppression, writableStackTrace);
13:     }
14:
15:     public DatabaseException(String message, Throwable cause) {
16:         super(message, cause);
17:     }
18:
19:     public DatabaseException(String message) {
20:         super(message);
21:     }
22:
23:     public DatabaseException(Throwable cause) {
24:         super(cause);
25:     }
26:
27: }

```

ファイル ParameterException.java

```

01: package chapter11.exception;
02:
03: public class ParameterException extends Exception {
04:
05:     private static final long serialVersionUID = 1L;
06:
07:     public ParameterException() {
08:     }
09:
10:     public ParameterException(String message) {
11:         super(message);
12:     }
13:

```



```

14:     public ParameterException(Throwable cause) {
15:         super(cause);
16:     }
17:
18:     public ParameterException(String message, Throwable cause) {
19:         super(message, cause);
20:     }
21:
22:     public ParameterException(String message, Throwable cause,
23:         boolean enableSuppression, boolean writableStackTrace) {
24:         super(message, cause, enableSuppression, writableStackTrace);
25:     }
26:
27: }

```

ファイル SystemException.java

```

01:     package chapter11.exception;
02:
03:     public class SystemException extends Exception {
04:         private static final long serialVersionUID = 1L;
05:
06:         public SystemException() {
07:             super();
08:         }
09:
10:         public SystemException(String message, Throwable cause,
11:             boolean enableSuppression, boolean writableStackTrace) {
12:             super(message, cause, enableSuppression, writableStackTrace);
13:         }
14:
15:         public SystemException(String message, Throwable cause) {
16:             super(message, cause);
17:         }
18:
19:         public SystemException(String message) {
20:             super(message);
21:         }
22:
23:         public SystemException(Throwable cause) {
24:             super(cause);
25:         }
26:
27:     }

```

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 = "jdbc:mysql://localhost/tsubuyaki";
06:     public static final String USERID = "user";
07:     public static final String PASSWORD = "password";
08: }
```

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 TsubuyakiParameters.java

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

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 tsubuyakiinput.html

```
01: <!DOCTYPE html>
02:
03: <html lang="ja">
04: <head>
05: <meta charset="UTF-8">
06: <title>つぶやきアプリ</title>
07: </head>
08: <body>
09: <h1>つぶやきアプリ</h1>
10: <form action="TsubuyakiInputServlet" method="post">
11: <p>
12:     名前:<br> <input type="text" name="name" maxlength="50">
13: </p>
14: <p>
15:     つぶやき:<br>
16:     <textarea rows="5" cols="40" name="tsubuyaki"></textarea>
17: </p>
18: <p>
19:     <input type="submit" value="送信">
20:     <input type="reset">
21:     <input type="button" onclick="location.href='TsubuyakiListServlet'"
22:     value="戻る">
23: </form>
24: </html>
```

ファイル tsubuyakilist.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="TsubuyakiArrayBean"
15:     class="chapter11.TsubuyakiArrayBean" scope="session" />
16: <h1>つぶやきアプリ</h1>
17: <form action="tsubuyakiinput.html" method="get">
18:   <input type="submit" value="書き込み">
19: </form>
20: <table border="1">
21:   <tr>
22:     <th>名前</th>
23:     <th>つぶやき</th>
24:     <th>日時</th>
25:   </tr>
26:   <%
27:     ArrayList<TsubuyakiBean> tsubuyakiArray = TsubuyakiArrayBean
28:       .getTsubuyakiArray();
29:     for (TsubuyakiBean record : tsubuyakiArray) {
30:   %>
31:   <tr>
32:     <td><a href="TsubuyakiPersonServlet?name=<%=record.getName()%>">
33:       <%=record.getName()%>
34:     </a></td>
35:     <td><%=record.getTsubuyaki()%></td>
36:     <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
37:       .format(record.getDatetime())%></td>
38:   </tr>
39:   <%
40:   }
41:   %>
42: </table>
43: </body>
44: </html>
```

ファイル tsubuyakipersonlist.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="TsubuyakiArrayBean"
15:     class="chapter11.TsubuyakiArrayBean" 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="TsubuyakiListServlet" method="get">
22:     <input type="submit" value="戻る">
23: </form></td>
24: </tr>
25: </table>
26: <table border="1">
27: <tr>
28: <th>名前</th>
29: <th>つぶやき</th>
30: <th>日時</th>
31: </tr>
32: <%
33:     ArrayList<TsubuyakiBean> tsubuyakiArray = TsubuyakiArrayBean
34:         .getTsubuyakiArray();
35:     for (TsubuyakiBean record : tsubuyakiArray) {
36: %>
37: <tr>
38: <td><%=record.getName()%></td>
39: <td><%=record.getTsubuyaki()%></td>
40: <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
41:     .format(record.getDatetime())%></td>
42: </tr>
43: <%
44:     }
45: %>
46: </table>
47: </body>
48: </html>
```

第2段階

プロジェクト名 STEP2



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。

SQL(MySQL 用)

```
01: CREATE TABLE USERS (  
02:     USERID CHAR(20) PRIMARY KEY NOT NULL,  
03:     PASSWORD CHAR(20) NOT NULL  
04: );  
05:  
06: CREATE TABLE USER_ROLES (  
07:     USERID CHAR(20) NOT NULL,  
08:     ROLE CHAR(20) NOT NULL,  
09:     PRIMARY KEY (USERID, ROLE)  
10: );  
11:  
12: INSERT INTO USERS VALUES ('u0001', 'pass');  
13: INSERT INTO USERS VALUES ('a0001', 'pass');  
14:  
15: INSERT INTO USER_ROLES VALUES ('u0001', 'users');  
16: INSERT INTO USER_ROLES VALUES ('a0001', 'admins');
```

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:     private static final long serialVersionUID = 1L;  
15:  
16:     public LogoutServlet() {  
17:         super();  
18:     }  
19:  
20:     @Override  
21:     protected void doGet(HttpServletRequest request, HttpServletResponse response)  
22:         throws ServletException, IOException {  
23:         HttpSession session = request.getSession(false);  
24:         if (session != null) {  
25:             session.invalidate();  
26:             response.sendRedirect("TsubuyakiListServlet");  
27:         }  
28:     }  
29: }
```

ファイル TsubuyakiArrayBean.java

```
01: package chapter11;
02:
03: import java.io.Serializable;
04: import java.util.ArrayList;
05:
06: public class TsubuyakiArrayBean implements Serializable {
07:
08:     private static final long serialVersionUID = 1L;
09:
10:     private ArrayList<TsubuyakiBean> tsubuyakiArray;
11:
12:     public TsubuyakiArrayBean() {
13:         tsubuyakiArray = new ArrayList<TsubuyakiBean>();
14:     }
15:
16:     public void addTsubuyaki(TsubuyakiBean obj) {
17:         tsubuyakiArray.add(obj);
18:     }
19:
20:     public int getArraySize() {
21:         return tsubuyakiArray.size();
22:     }
23:
24:     public ArrayList<TsubuyakiBean> getTsubuyakiArray() {
25:         return tsubuyakiArray;
26:     }
27:
28:     public void setTsubuyakiArray(ArrayList<TsubuyakiBean> tsubuyakiArray) {
29:         this.tsubuyakiArray = tsubuyakiArray;
30:     }
31:
32: }
```

ファイル TsubuyakiBean.java

```
01: package chapter11;
02:
03: import java.io.Serializable;
04: import java.util.Date;
05:
06: public class TsubuyakiBean implements Serializable {
07:
08:     private static final long serialVersionUID = 1L;
09:
10:     private String name;
11:     private String tsubuyaki;
12:     private Date datetime;
13: }
```



```

14:     public TsubuyakiBean() {
15:     }
16:
17:     public String getName() {
18:         return name;
19:     }
20:
21:     public void setName(String name) {
22:         this.name = name;
23:     }
24:
25:     public String getTsubuyaki() {
26:         return tsubuyaki;
27:     }
28:
29:     public void setTsubuyaki(String tsubuyaki) {
30:         this.tsubuyaki = tsubuyaki;
31:     }
32:
33:     public Date getDatetime() {
34:         return datetime;
35:     }
36:
37:     public void setDatetime(Date datetime) {
38:         this.datetime = datetime;
39:     }
40:
41: }

```

ファイル TsubuyakiDeleteServlet.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.TsubuyakiDAO;
13:     import chapter11.exception.DatabaseException;
14:     import chapter11.exception.SystemException;
15:
16:     @WebServlet("/TsubuyakiDeleteServlet")
17:     public class TsubuyakiDeleteServlet extends HttpServlet {
18:         private static final long serialVersionUID = 1L;
19:

```

```

20:     public TsubuyakiDeleteServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:         try {
28:             String name = request.getParameter("name");
29:
30:             TsubuyakiDAO dao = new TsubuyakiDAO();
31:             dao.deleteTsubuyaki(name);
32:
33:             response.sendRedirect("TsubuyakiListServlet");
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 TsubuyakiInputServlet.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.TsubuyakiDAO;
14:     import chapter11.exception.DatabaseException;
15:     import chapter11.exception.SystemException;
16:
17:     @WebServlet("/TsubuyakiInputServlet")

```

```

18: public class TsubuyakiInputServlet extends HttpServlet {
19:     private static final long serialVersionUID = 1L;
20:
21:     public TsubuyakiInputServlet() {
22:         super();
23:     }
24:
25:     @Override
26:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
27:         throws ServletException, IOException {
28:         try {
29:             String name = request.getParameter("name");
30:             String tsubuyaki = request.getParameter("tsubuyaki");
31:             Date date = new Date();
32:
33:             TsubuyakiBean tsubuyakiBean = new TsubuyakiBean();
34:             tsubuyakiBean.setName(name);
35:             tsubuyakiBean.setTsubuyaki(tsubuyaki);
36:             tsubuyakiBean.setDatetime(date);
37:
38:             TsubuyakiDAO dao = new TsubuyakiDAO();
39:             dao.setTsubuyaki(tsubuyakiBean);
40:
41:             response.sendRedirect("TsubuyakiListServlet");
42:         } catch (SystemException e) {
43:             e.printStackTrace();
44:             HttpSession session = request.getSession();
45:             session.setAttribute("Except", e);
46:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:                 response);
48:         } catch (DatabaseException e) {
49:             e.printStackTrace();
50:             HttpSession session = request.getSession();
51:             session.setAttribute("Except", e);
52:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
53:                 response);
54:         }
55:     }
56:
57: }

```

ファイル TsubuyakiListServlet.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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiListServlet")
17: public class TsubuyakiListServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiListServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         TsubuyakiArrayBean tsubuyakiArray;
29:
30:         try {
31:             TsubuyakiDAO dao = new TsubuyakiDAO();
32:             tsubuyakiArray = dao.getTsubuyakiArray();
33:             HttpSession session = request.getSession();
34:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
35:             getServletContext().getRequestDispatcher("/tsubuyakilist.jsp").forward(
36:                 request, response);
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: }

```

ファイル TsubuyakiPersonServlet.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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiPersonServlet")
17: public class TsubuyakiPersonServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiPersonServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         TsubuyakiArrayBean tsubuyakiArray;
29:
30:         try {
31:             String name = request.getParameter("name");
32:
33:             TsubuyakiDAO dao = new TsubuyakiDAO();
34:             tsubuyakiArray = dao.getPersonTsubuyakiArray(name);
35:             HttpSession session = request.getSession();
36:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
37:             getServletContext().getRequestDispatcher("/tsubuyakipersonlist.jsp")
38:                 .forward(request, response);
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 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 TsubuyakiDAO.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.TsubuyakiArrayBean;
10: import chapter11.TsubuyakiBean;
11: import chapter11.exception.DatabaseException;
12: import chapter11.exception.SystemException;
13: import chapter11.parameter.ExceptionParameters;
14: import chapter11.parameter.TsubuyakiParameters;
15:
16: public class TsubuyakiDAO extends DAOBase {
17:
18:     public TsubuyakiArrayBean getTsubuyakiArray() throws DatabaseException,
19:         SystemException {
20:         Statement stmt = null;
21:         TsubuyakiArrayBean tsubuyakiArray = new TsubuyakiArrayBean();
22:         this.open();
23:         try {
24:             stmt = con.createStatement();
25:             ResultSet rs = stmt.executeQuery(TsubuyakiParameters.SQL_SELECT_ALL);
26:             while (rs.next()) {
27:                 TsubuyakiBean record = new TsubuyakiBean();
28:                 record.setName(rs.getString(TsubuyakiParameters.NAME));
29:                 record.setTsubuyaki(rs.getString(TsubuyakiParameters.TSUBUYAKI));
30:                 record.setDatetime(rs.getTimestamp(TsubuyakiParameters.DATE));
31:                 tsubuyakiArray.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 tsubuyakiArray;
40:     }
41:
42:     public TsubuyakiArrayBean getPersonTsubuyakiArray(String name)
43:         throws DatabaseException, SystemException {
44:         PreparedStatement stmt = null;
45:         TsubuyakiArrayBean tsubuyakiArray = new TsubuyakiArrayBean();
```

```

46:         this.open();
47:         try {
48:             stmt = con.prepareStatement(TsubuyakiParameters.SQL_SELECT_PERSON);
49:             stmt.setString(1, name);
50:             ResultSet rs = stmt.executeQuery();
51:             while (rs.next()) {
52:                 TsubuyakiBean record = new TsubuyakiBean();
53:                 record.setName(rs.getString(TsubuyakiParameters.NAME));
54:                 record.setTsubuyaki(rs.getString(TsubuyakiParameters.TSUBUYAKI));
55:                 record.setDatetime(rs.getTimestamp(TsubuyakiParameters.DATE));
56:                 tsubuyakiArray.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 tsubuyakiArray;
65:     }
66:
67:     public void setTsubuyaki(TsubuyakiBean tsubuyaki) throws DatabaseException,
68:         SystemException {
69:         PreparedStatement stmt = null;
70:         this.open();
71:         try {
72:             stmt = con.prepareStatement(TsubuyakiParameters.SQL_INSERT);
73:             stmt.setString(1, tsubuyaki.getName());
74:             stmt.setString(2, tsubuyaki.getTsubuyaki());
75:             stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
76:                 .format(tsubuyaki.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 deleteTsubuyaki(String name) throws DatabaseException,
87:         SystemException {
88:         PreparedStatement stmt = null;
89:         this.open();
90:         try {
91:             stmt = con.prepareStatement(TsubuyakiParameters.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:     private static final long serialVersionUID = 1L;
05:
06:     public DatabaseException() {
07:         super();
08:     }
09:
10:     public DatabaseException(String message, Throwable cause,
11:         boolean enableSuppression, boolean writableStackTrace) {
12:         super(message, cause, enableSuppression, writableStackTrace);
13:     }
14:
15:     public DatabaseException(String message, Throwable cause) {
16:         super(message, cause);
17:     }
18:
19:     public DatabaseException(String message) {
20:         super(message);
21:     }
22:
23:     public DatabaseException(Throwable cause) {
24:         super(cause);
25:     }
26:
27: }

```

ファイル ParameterException.java

```

01: package chapter11.exception;
02:
03: public class ParameterException extends Exception {
04:
05:     private static final long serialVersionUID = 1L;
06:
07:     public ParameterException() {
08:     }
09:
10:     public ParameterException(String message) {
11:         super(message);

```

```

12:     }
13:
14:     public ParameterException(Throwable cause) {
15:         super(cause);
16:     }
17:
18:     public ParameterException(String message, Throwable cause) {
19:         super(message, cause);
20:     }
21:
22:     public ParameterException(String message, Throwable cause,
23:         boolean enableSuppression, boolean writableStackTrace) {
24:         super(message, cause, enableSuppression, writableStackTrace);
25:     }
26:
27: }

```

ファイル SystemException.java

```

01: package chapter11.exception;
02:
03: public class SystemException extends Exception {
04:     private static final long serialVersionUID = 1L;
05:
06:     public SystemException() {
07:         super();
08:     }
09:
10:     public SystemException(String message, Throwable cause,
11:         boolean enableSuppression, boolean writableStackTrace) {
12:         super(message, cause, enableSuppression, writableStackTrace);
13:     }
14:
15:     public SystemException(String message, Throwable cause) {
16:         super(message, cause);
17:     }
18:
19:     public SystemException(String message) {
20:         super(message);
21:     }
22:
23:     public SystemException(Throwable cause) {
24:         super(cause);
25:     }
26:
27: }

```

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 = "jdbc:mysql://localhost/tsubuyaki";
06:     public static final String USERID = "user";
07:     public static final String PASSWORD = "password";
08: }
```

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 TsubuyakiParameters.java

```
01: package chapter11.parameter;
02:
03: public class TsubuyakiParameters {
04:     public static final String SQL_SELECT_ALL = "SELECT * FROM TSUBUYAKI ORDER BY DATE
05: DESC";
06:     public static final String SQL_SELECT_PERSON = "SELECT * FROM TSUBUYAKI WHERE NAME
07: = ? ORDER BY DATE DESC";
08:
09:     public static final String SQL_INSERT = "INSERT INTO TSUBUYAKI VALUES(?,?,?)";
10:
11:     public static final String SQL_DELETE = "DELETE FROM TSUBUYAKI WHERE NAME=?";
12:
13:     public static final String NAME = "NAME";
14:     public static final String TSUBUYAKI = "TSUBUYAKI";
15:     public static final String DATE = "DATE";
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 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 login.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>つぶやき ログイン</title>
09: </head>
10: <body>
11: <h1>つぶやき</h1>
12: <h2>ログイン</h2>
13: <form action="j_security_check" method="post">
14:     ログイン ID : <input type="text" name="j_username"><br>
15:     パスワード : <input type="password" name="j_password"><br>
16:     <input type="submit" value="LOGIN"> <input type="reset">
17: </form>
```

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

ファイル loginerror.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>つぶやき ログインエラー</title>
09:     </head>
10:     <body>
11:     <h1>つぶやき</h1>
12:     <h2>ログインエラー</h2>
13:     ログイン ID またはパスワードに誤りがあります。
14:     <br> 再度入力してください。
15:     <br>
16:     <a href=" javascript:history.back();">戻る</a>
17:     </body>
18: </html>
```

ファイル tsubuyakiinput.html

```
01:     <!DOCTYPE html>
02:
03:     <html lang="ja">
04:     <head>
05:     <meta charset="UTF-8">
06:     <title>つぶやきアプリ</title>
07:     </head>
08:     <body>
09:     <h1>つぶやきアプリ</h1>
10:     <form action="TsubuyakiInputServlet" method="post">
11:     <p>
12:     名前:<br> <input type="text" name="name" maxlength="50">
13:     </p>
14:     <p>
15:     つぶやき:<br>
16:     <textarea rows="5" cols="40" name="tsubuyaki"></textarea>
17:     </p>
18:     <p>
19:     <input type="submit" value="送信"> <input type="reset">
20:     <input type="button" onclick="location.href='TsubuyakiListServlet'"
21:     value="戻る">
22:     </form>
23: </html>
```

ファイル tsubuyakilist.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="TsubuyakiArrayBean"
15:     class="chapter11.TsubuyakiArrayBean" scope="session" />
16: <h1>つぶやきアプリ</h1>
17: <%=request.getRemoteUser()%> :
18: <a href="LogoutServlet">ログアウト</a>
19: <br>
20: <form action="tsubuyakiinput.html" method="get">
21: <input type="submit" value="書き込み">
22: </form>
23: <table border="1">
24: <tr>
25: <th>名前</th>
26: <th>つぶやき</th>
27: <th>日時</th>
28: </tr>
29: <%
30:     ArrayList<TsubuyakiBean> tsubuyakiArray = TsubuyakiArrayBean
31:         .getTsubuyakiArray();
32:     for (TsubuyakiBean record : tsubuyakiArray) {
33: %>
34: <tr>
35: <td><a href="TsubuyakiPersonServlet?name=<%=record.getName()%>">
36:     <%=record.getName()%>
37: </a></td>
38: <td><%=record.getTsubuyaki()%></td>
39: <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
40:     .getDatetime())%></td>
41: </tr>
42: <%
43:     }
44: %>
45: </table>
46: </body>
47: </html>
```

ファイル tsubuyakipersonlist.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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
08:     "http://www.w3.org/TR/html4/loose.dtd">
09: <html>
10: <head>
11: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12: <title>つぶやきアプリ</title>
13: </head>
14: <body>
15: <jsp:useBean id="TsubuyakiArrayBean"
16:     class="chapter11.TsubuyakiArrayBean" scope="session" />
17: <h1>つぶやきアプリ</h1>
18: <h2><%=request.getParameter("name")%>さんのつぶやき
19: </h2>
20: <table>
21: <tr>
22: <td><form action="TsubuyakiListServlet" method="get">
23: <input type="submit" value="戻る">
24: </form></td>
25: <%
26:     if (request.isUserInRole(UserRoleParameters.ROLE_ADMINS)) {
27:     %>
28: <td><form action="TsubuyakiDeleteServlet" method="get">
29: <input type="hidden" name="name"
30:     value="<%=request.getParameter("name")%>"> <input
31:     type="submit" value="削除">
32: </form></td>
33: <%
34:     }
35:     %>
36: </tr>
37: </table>
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<TsubuyakiBean> tsubuyakiArray = TsubuyakiArrayBean
46:     .getTsubuyakiArray();
47:     for (TsubuyakiBean record : tsubuyakiArray) {
48:     %>
49: <tr>
```

```

50:         <td><%=record.getName()%></td>
51:         <td><%=record.getTsubuyaki()%></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://java.sun.com/xml/ns/javaee"
04: xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
05: xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
06: http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
07:     <display-name>STEP2</display-name>
08:     <welcome-file-list>
09:         <welcome-file>index.html</welcome-file>
10:         <welcome-file>index.htm</welcome-file>
11:         <welcome-file>index.jsp</welcome-file>
12:         <welcome-file>default.html</welcome-file>
13:         <welcome-file>default.htm</welcome-file>
14:         <welcome-file>default.jsp</welcome-file>
15:     </welcome-file-list>
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>/TsubuyakiListServlet</url-pattern>
29:             <url-pattern>/TsubuyakiPersonServlet</url-pattern>
30:             <url-pattern>/TsubuyakiInputServlet</url-pattern>
31:             <url-pattern>/LogoutServlet</url-pattern>
32:             <url-pattern>/tsubuyakiinput.html</url-pattern>
33:             <url-pattern>/tsubuyakilist.jsp</url-pattern>
34:             <url-pattern>/tsubuyakipersonlist.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: </web-app>

```

ファイル server.xml

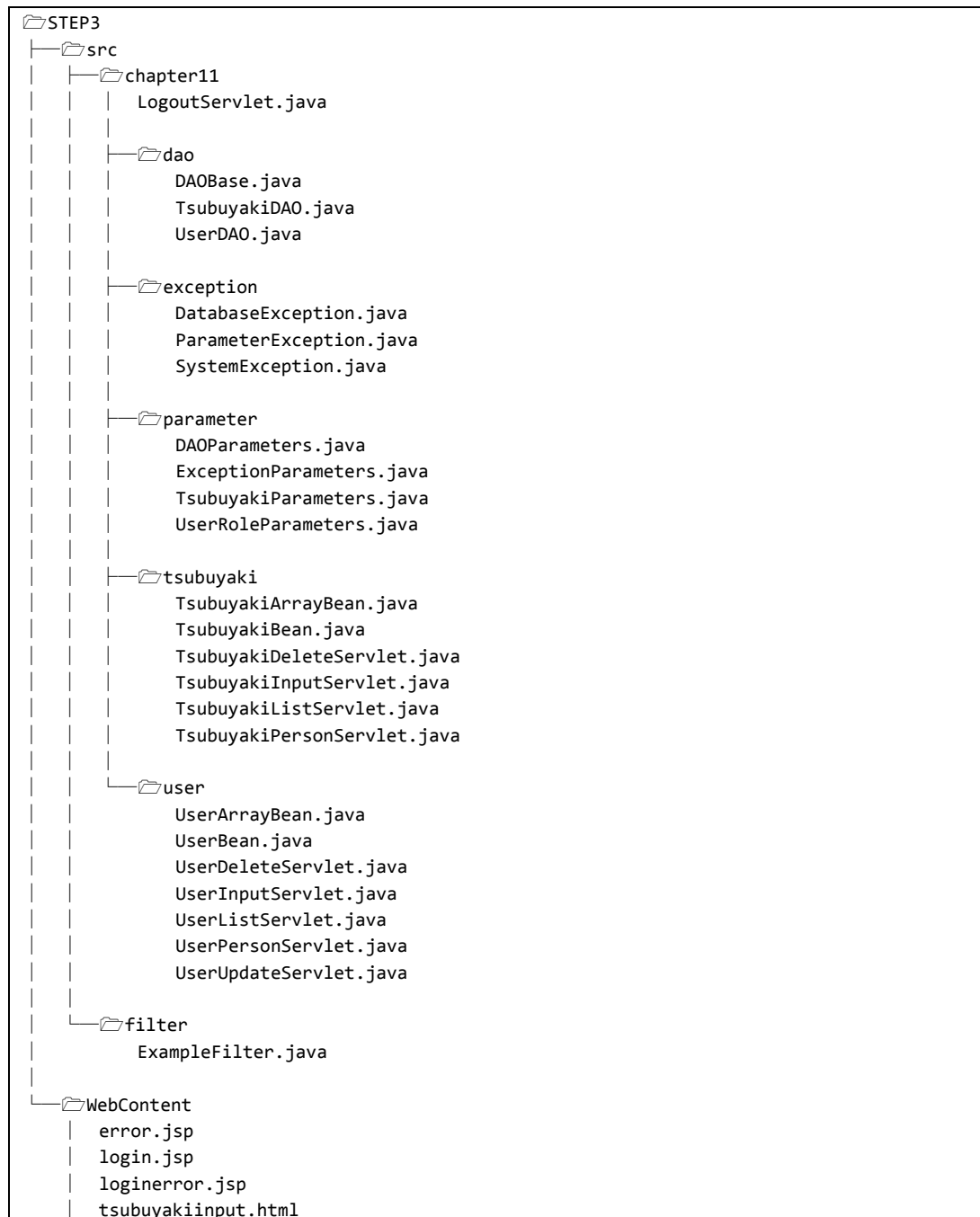
```

01:     (省略)
02:     <Realm className="org.apache.catalina.realm.JDBCRealm"
03:         driverName="com.mysql.jdbc.Driver"
04:         connectionURL="jdbc:mysql://localhost/tsubuyaki"
05:         connectionName="user"
06:         connectionPassword="password"
07:         userTable="users"
08:         userNameCol="userid"
09:         userCredCol="password"
10:         userRoleTable="user_roles"
11:         roleNameCol="role" />
12:     (省略)

```

第3段階

プロジェクト名 STEP3



```
| tsubuyakilist.jsp  
| tsubuyakipersonlist.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:     private static final long serialVersionUID = 1L;
15:
16:     public LogoutServlet() {
17:         super();
18:     }
19:
20:     @Override
21:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
22:         throws ServletException, IOException {
23:         HttpSession session = request.getSession(false);
24:         if (session != null) {
25:             session.invalidate();
26:             response.sendRedirect("TsubuyakiListServlet");
27:         }
28:     }
29: }
```

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 TsubuyakiDAO.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.TsubuyakiParameters;
13:     import chapter11.tsubuyaki.TsubuyakiArrayBean;
14:     import chapter11.tsubuyaki.TsubuyakiBean;
15:
16:     public class TsubuyakiDAO extends DAOBase {
17:
18:         public TsubuyakiArrayBean getTsubuyakiArray() throws DatabaseException,
19:             SystemException {

```

```

20:     Statement stmt = null;
21:     TsubuyakiArrayBean tsubuyakiArray = new TsubuyakiArrayBean();
22:     this.open();
23:     try {
24:         stmt = con.createStatement();
25:         ResultSet rs = stmt.executeQuery(TsubuyakiParameters.SQL_SELECT_ALL);
26:         while (rs.next()) {
27:             TsubuyakiBean record = new TsubuyakiBean();
28:             record.setName(rs.getString(TsubuyakiParameters.NAME));
29:             record.setTsubuyaki(rs.getString(TsubuyakiParameters.TSUBUYAKI));
30:             record.setDatetime(rs.getTimestamp(TsubuyakiParameters.DATE));
31:             tsubuyakiArray.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 tsubuyakiArray;
40: }
41:
42: public TsubuyakiArrayBean getPersonTsubuyakiArray(String name)
43:     throws DatabaseException, SystemException {
44:     PreparedStatement stmt = null;
45:     TsubuyakiArrayBean tsubuyakiArray = new TsubuyakiArrayBean();
46:     this.open();
47:     try {
48:         stmt = con.prepareStatement(TsubuyakiParameters.SQL_SELECT_PERSON);
49:         stmt.setString(1, name);
50:         ResultSet rs = stmt.executeQuery();
51:         while (rs.next()) {
52:             TsubuyakiBean record = new TsubuyakiBean();
53:             record.setName(rs.getString(TsubuyakiParameters.NAME));
54:             record.setTsubuyaki(rs.getString(TsubuyakiParameters.TSUBUYAKI));
55:             record.setDatetime(rs.getTimestamp(TsubuyakiParameters.DATE));
56:             tsubuyakiArray.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 tsubuyakiArray;
65: }
66:
67: public void setTsubuyaki(TsubuyakiBean tsubuyaki) throws DatabaseException,
68:     SystemException {
69:     PreparedStatement stmt = null;

```

```

70:     this.open();
71:     try {
72:         stmt = con.prepareStatement(TsubuyakiParameters.SQL_INSERT);
73:         stmt.setString(1, tsubuyaki.getName());
74:         stmt.setString(2, tsubuyaki.getTsubuyaki());
75:         stmt.setString(3, new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
76:             .format(tsubuyaki.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 deleteTsubuyaki(String name) throws DatabaseException,
87:     SystemException {
88:     PreparedStatement stmt = null;
89:     this.open();
90:     try {
91:         stmt = con.prepareStatement(TsubuyakiParameters.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:     private static final long serialVersionUID = 1L;
05:
06:     public DatabaseException() {
07:         super();

```

```

08:     }
09:
10:     public DatabaseException(String message, Throwable cause,
11:         boolean enableSuppression, boolean writableStackTrace) {
12:         super(message, cause, enableSuppression, writableStackTrace);
13:     }
14:
15:     public DatabaseException(String message, Throwable cause) {
16:         super(message, cause);
17:     }
18:
19:     public DatabaseException(String message) {
20:         super(message);
21:     }
22:
23:     public DatabaseException(Throwable cause) {
24:         super(cause);
25:     }
26:
27: }

```

ファイル ParameterException.java

```

01: package chapter11.exception;
02:
03: public class ParameterException extends Exception {
04:
05:     private static final long serialVersionUID = 1L;
06:
07:     public ParameterException() {
08:     }
09:
10:     public ParameterException(String message) {
11:         super(message);
12:     }
13:
14:     public ParameterException(Throwable cause) {
15:         super(cause);
16:     }
17:
18:     public ParameterException(String message, Throwable cause) {
19:         super(message, cause);
20:     }
21:
22:     public ParameterException(String message, Throwable cause,
23:         boolean enableSuppression, boolean writableStackTrace) {
24:         super(message, cause, enableSuppression, writableStackTrace);
25:     }
26:
27: }

```

ファイル SystemException.java

```
01: package chapter11.exception;
02:
03: public class SystemException extends Exception {
04:     private static final long serialVersionUID = 1L;
05:
06:     public SystemException() {
07:         super();
08:     }
09:
10:     public SystemException(String message, Throwable cause,
11:         boolean enableSuppression, boolean writableStackTrace) {
12:         super(message, cause, enableSuppression, writableStackTrace);
13:     }
14:
15:     public SystemException(String message, Throwable cause) {
16:         super(message, cause);
17:     }
18:
19:     public SystemException(String message) {
20:         super(message);
21:     }
22:
23:     public SystemException(Throwable cause) {
24:         super(cause);
25:     }
26:
27: }
```

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 = "jdbc:mysql://localhost/tsubuyaki";
06:     public static final String USERID = "user";
07:     public static final String PASSWORD = "password";
08: }
```

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 TsubuyakiParameters.java

```

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

ファイル TsubuyakiArrayBean.java

```
01: package chapter11.tsubuyaki;
02:
03: import java.io.Serializable;
04: import java.util.ArrayList;
05:
06: public class TsubuyakiArrayBean implements Serializable {
07:
08:     private static final long serialVersionUID = 1L;
09:
10:     private ArrayList<TsubuyakiBean> tsubuyakiArray;
11:
12:     public TsubuyakiArrayBean() {
13:         tsubuyakiArray = new ArrayList<TsubuyakiBean>();
14:     }
15:
16:     public void addTsubuyaki(TsubuyakiBean obj) {
17:         tsubuyakiArray.add(obj);
18:     }
19:
20:     public int getArraySize() {
21:         return tsubuyakiArray.size();
22:     }
23:
24:     public ArrayList<TsubuyakiBean> getTsubuyakiArray() {
25:         return tsubuyakiArray;
26:     }
27:
28:     public void setTsubuyakiArray(ArrayList<TsubuyakiBean> tsubuyakiArray) {
29:         this.tsubuyakiArray = tsubuyakiArray;
30:     }
31:
32: }
```

ファイル TsubuyakiBean.java

```
01: package chapter11.tsubuyaki;
02:
03: import java.io.Serializable;
04: import java.util.Date;
05:
06: public class TsubuyakiBean implements Serializable {
07:
08:     private static final long serialVersionUID = 1L;
09:
10:     private String name;
```

```

11:     private String tsubuyaki;
12:     private Date datetime;
13:
14:     public TsubuyakiBean() {
15:     }
16:
17:     public String getName() {
18:         return name;
19:     }
20:
21:     public void setName(String name) {
22:         this.name = name;
23:     }
24:
25:     public String getTsubuyaki() {
26:         return tsubuyaki;
27:     }
28:
29:     public void setTsubuyaki(String tsubuyaki) {
30:         this.tsubuyaki = tsubuyaki;
31:     }
32:
33:     public Date getDatetime() {
34:         return datetime;
35:     }
36:
37:     public void setDatetime(Date datetime) {
38:         this.datetime = datetime;
39:     }
40:
41: }

```

ファイル TsubuyakiDeleteServlet.java

```

01: package chapter11.tsubuyaki;
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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiDeleteServlet")

```

```

17: public class TsubuyakiDeleteServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiDeleteServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:         try {
28:             String name = request.getParameter("name");
29:
30:             TsubuyakiDAO dao = new TsubuyakiDAO();
31:             dao.deleteTsubuyaki(name);
32:
33:             response.sendRedirect("TsubuyakiListServlet");
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 TsubuyakiInputServlet.java

```

01: package chapter11.tsubuyaki;
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.TsubuyakiDAO;
14: import chapter11.exception.DatabaseException;

```



```

15: import chapter11.exception.SystemException;
16:
17: @WebServlet("/TsubuyakiInputServlet")
18: public class TsubuyakiInputServlet extends HttpServlet {
19:     private static final long serialVersionUID = 1L;
20:
21:     public TsubuyakiInputServlet() {
22:         super();
23:     }
24:
25:     @Override
26:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
27:         throws ServletException, IOException {
28:         try {
29:             String name = request.getParameter("name");
30:             String tsubuyaki = request.getParameter("tsubuyaki");
31:             Date date = new Date();
32:
33:             TsubuyakiBean tsubuyakiBean = new TsubuyakiBean();
34:             tsubuyakiBean.setName(name);
35:             tsubuyakiBean.setTsubuyaki(tsubuyaki);
36:             tsubuyakiBean.setDatetime(date);
37:
38:             TsubuyakiDAO dao = new TsubuyakiDAO();
39:             dao.setTsubuyaki(tsubuyakiBean);
40:
41:             response.sendRedirect("TsubuyakiListServlet");
42:         } catch (SystemException e) {
43:             e.printStackTrace();
44:             HttpSession session = request.getSession();
45:             session.setAttribute("Except", e);
46:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:                 response);
48:         } catch (DatabaseException e) {
49:             e.printStackTrace();
50:             HttpSession session = request.getSession();
51:             session.setAttribute("Except", e);
52:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
53:                 response);
54:         }
55:     }
56:
57: }

```

ファイル TsubuyakiListServlet.java

```

01: package chapter11.tsubuyaki;
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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiListServlet")
17: public class TsubuyakiListServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiListServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         TsubuyakiArrayBean tsubuyakiArray;
29:
30:         try {
31:             TsubuyakiDAO dao = new TsubuyakiDAO();
32:             tsubuyakiArray = dao.getTsubuyakiArray();
33:             HttpSession session = request.getSession();
34:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
35:             getServletContext().getRequestDispatcher("/tsubuyakilist.jsp").forward(
36:                 request, response);
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: }

```

ファイル TsubuyakiPersonServlet.java

```

01: package chapter11.tsubuyaki;
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.TsubuyakiDAO;
13: import chapter11.exception.DatabaseException;
14: import chapter11.exception.SystemException;
15:
16: @WebServlet("/TsubuyakiPersonServlet")
17: public class TsubuyakiPersonServlet extends HttpServlet {
18:     private static final long serialVersionUID = 1L;
19:
20:     public TsubuyakiPersonServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         TsubuyakiArrayBean tsubuyakiArray;
29:
30:         try {
31:             String name = request.getParameter("name");
32:
33:             TsubuyakiDAO dao = new TsubuyakiDAO();
34:             tsubuyakiArray = dao.getPersonTsubuyakiArray(name);
35:             HttpSession session = request.getSession();
36:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
37:             getServletContext().getRequestDispatcher("/tsubuyakipersonlist.jsp")
38:                 .forward(request, response);
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 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 static final long serialVersionUID = 1L;
09:
10:     private ArrayList<UserBean> userArray;
11:
12:     public UserArrayBean() {
13:         userArray = new ArrayList<UserBean>();
14:     }
15:
16:     public void addUser(UserBean obj) {
17:         userArray.add(obj);
18:     }
19:
20:     public int getArraySize() {
21:         return userArray.size();
22:     }
23:
24:     public ArrayList<UserBean> getUserArray() {
25:         return userArray;
26:     }
27:
28:     public void setUserArray(ArrayList<UserBean> userArray) {
29:         this.userArray = userArray;
30:     }
31:
32: }

```

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 static final long serialVersionUID = 1L;
08:

```

```

09:     private String userid;
10:     private String passwd;
11:     private String role;
12:
13:     public UserBean() {
14:     }
15:
16:     public String getUserid() {
17:         return userid;
18:     }
19:
20:     public void setUserid(String userid) {
21:         this.userid = userid;
22:     }
23:
24:     public String getPasswd() {
25:         return passwd;
26:     }
27:
28:     public void setPasswd(String passwd) {
29:         this.passwd = passwd;
30:     }
31:
32:     public String getRole() {
33:         return role;
34:     }
35:
36:     public void setRole(String role) {
37:         this.role = role;
38:     }
39:
40: }

```

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:

```

```

16:  @WebServlet("/UserDeleteServlet")
17:  public class UserDeleteServlet extends HttpServlet {
18:      private static final long serialVersionUID = 1L;
19:
20:      public UserDeleteServlet() {
21:          super();
22:      }
23:
24:      @Override
25:      protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:          throws ServletException, IOException {
27:          try {
28:              String userid = request.getParameter("userid");
29:
30:              UserDAO dao = new UserDAO();
31:              dao.deleteUser(userid);
32:
33:              response.sendRedirect("UserListServlet");
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 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:     private static final long serialVersionUID = 1L;
19:
20:     public UserInputServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:         try {
28:             String userid = request.getParameter("userid");
29:             String passwd = request.getParameter("passwd");
30:             String role = request.getParameter("role");
31:
32:             UserBean userBean = new UserBean();
33:             userBean.setUserid(userid);
34:             userBean.setPasswd(passwd);
35:             userBean.setRole(role);
36:
37:             UserDAO dao = new UserDAO();
38:             dao.addUser(userBean);
39:
40:             response.sendRedirect("UserListServlet");
41:         } catch (SystemException e) {
42:             e.printStackTrace();
43:             HttpSession session = request.getSession();
44:             session.setAttribute("Except", e);
45:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
46:                 response);
47:         } catch (DatabaseException e) {
48:             e.printStackTrace();
49:             HttpSession session = request.getSession();
50:             session.setAttribute("Except", e);
51:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
52:                 response);
53:         }
54:     }
55:
56: }

```

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:     private static final long serialVersionUID = 1L;
19:
20:     public UserListServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         UserArrayBean userArray;
29:
30:         try {
31:             UserDAO dao = new UserDAO();
32:             userArray = dao.getUserArray();
33:             HttpSession session = request.getSession();
34:             session.setAttribute("UserArrayBean", userArray);
35:             getServletContext().getRequestDispatcher("/userlist.jsp").forward(
36:                 request, response);
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: }

```

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:     private static final long serialVersionUID = 1L;
19:
20:     public UserPersonServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         UserBean user = null;
29:
30:         try {
31:             String userid = request.getParameter("userid");
32:
33:             UserDAO dao = new UserDAO();
34:             user = dao.getUser(userid);
35:             HttpSession session = request.getSession();
36:             session.setAttribute("UserBean", user);
37:             getServletContext().getRequestDispatcher("/userperson.jsp").forward(
38:                 request, response);
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 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:     private static final long serialVersionUID = 1L;
19:
20:     public UserUpdateServlet() {
21:         super();
22:     }
23:
24:     @Override
25:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:
28:         UserBean user = null;
29:
30:         try {
31:             String userid = request.getParameter("userid");
32:             String passwd = request.getParameter("passwd");
33:             String role = request.getParameter("role");
34:
35:             user = new UserBean();
36:             user.setUserid(userid);
37:             user.setPasswd(passwd);
38:             user.setRole(role);
39:
40:             UserDAO dao = new UserDAO();
41:             dao.updateUser(user);
42:
43:             response.sendRedirect("UserListServlet");

```

```

44:         } catch (SystemException e) {
45:             e.printStackTrace();
46:             HttpSession session = request.getSession();
47:             session.setAttribute("Except", e);
48:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
49:                 response);
50:         } catch (DatabaseException e) {
51:             e.printStackTrace();
52:             HttpSession session = request.getSession();
53:             session.setAttribute("Except", e);
54:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
55:                 response);
56:         }
57:     }
58: }
59: }

```

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 login.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>つぶやきアプリ ログイン</title>
09: </head>

```

```

10: <body>
11:   <h1>つぶやきアプリ</h1>
12:   <h2>ログイン</h2>
13:   <form action="j_security_check" method="post">
14:     ログイン ID : <input type="text" name="j_username"><br> パスワード : <input
15:       type="password" name="j_password"><br> <input
16:       type="submit" value="LOGIN"> <input type="reset">
17:   </form>
18: </body>
19: </html>

```

ファイル loginerror.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>つぶやきアプリ ログインエラー</title>
09: </head>
10: <body>
11:   <h1>つぶやきアプリ</h1>
12:   <h2>ログインエラー</h2>
13:   ログイン ID またはパスワードに誤りがあります。
14:   <br> 再度入力してください。
15:   <br>
16:   <a href="javascript:history.back();">戻る</a>
17: </body>
18: </html>

```

ファイル tsubuyakiinput.html

```

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

```

```

17:     </p>
18:     <p>
19:         <input type="submit" value="送信"> <input type="reset">
20:         <input type="button" onclick="location.href='TsubuyakiListServlet'"
21:             value="戻る">
22:     </form>
23: </html>

```

ファイル tsubuyakilist.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.tsubuyaki.*"%>
06: <%@ page import="chapter11.parameter.*"%>
07: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
08:     "http://www.w3.org/TR/html4/loose.dtd">
09: <html>
10: <head>
11: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12: <title>つぶやきアプリ</title>
13: </head>
14: <body>
15:     <jsp:useBean id="TsubuyakiArrayBean"
16:         class="chapter11.tsubuyaki.TsubuyakiArrayBean" scope="session" />
17:     <h1>つぶやきアプリ</h1>
18:     <%=request.getRemoteUser()%> :
19:     <a href="LogoutServlet">ログアウト</a>
20:     <%
21:         if (request.isUserInRole(UserRoleParameters.ROLE_ADMINS)) {
22:     %>
23:     <a href="UserListServlet">ユーザ管理</a>
24:     <%
25:         }
26:     %>
27:
28:     <br>
29:     <form action="tsubuyakiinput.html" method="get">
30:         <input type="submit" value="書き込み">
31:     </form>
32:     <table border="1">
33:     <tr>
34:         <th>名前</th>
35:         <th>つぶやき</th>
36:         <th>日時</th>
37:     </tr>
38:     <%
39:         ArrayList<TsubuyakiBean> tsubuyakiArray = TsubuyakiArrayBean
40:             .getTsubuyakiArray();
41:         for (TsubuyakiBean record : tsubuyakiArray) {

```

```

42:     %>
43:     <tr>
44:         <td><a href="TsubuyakiPersonServlet?name=<%=record.getName()%>">
45:             <%=record.getName()%>
46:         </a></td>
47:         <td><%=record.getTsubuyaki()%></td>
48:         <td><%=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(record
49:             .getDatetime())%></td>
50:     </tr>
51:     <%
52:     }
53:     %>
54: </table>
55: </body>
56: </html>

```

ファイル tsubuyakipersonlist.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.tsubuyaki.*"%>
06: <%@ page import="chapter11.parameter.*"%>
07: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
08:     "http://www.w3.org/TR/html4/loose.dtd">
09: <html>
10: <head>
11: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12: <title>つぶやきアプリ</title>
13: </head>
14: <body>
15: <jsp:useBean id="TsubuyakiArrayBean"
16:     class="chapter11.tsubuyaki.TsubuyakiArrayBean" scope="session" />
17: <h1>つぶやきアプリ</h1>
18: <h2><%=request.getParameter("name")%>さんのつぶやき
19: </h2>
20: <table>
21: <tr>
22:     <td><form action="TsubuyakiListServlet" method="get">
23:         <input type="submit" value="戻る">
24:     </form></td>
25:     <%
26:         if (request.isUserInRole(UserRoleParameters.ROLE_ADMINS)) {
27:         %>
28:     <td><form action="TsubuyakiDeleteServlet" method="get">
29:         <input type="hidden" name="name"
30:             value="<%=request.getParameter("name")%>"> <input
31:             type="submit" value="削除">
32:     </form></td>

```

```

33:     <%
34:     }
35:     %>
36: </tr>
37: </table>
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<TsubuyakiBean> tsubuyakiArray = TsubuyakiArrayBean
46:       .getTsubuyakiArray();
47:     for (TsubuyakiBean record : tsubuyakiArray) {
48:   %>
49:   <tr>
50:     <td><%=record.getName()%></td>
51:     <td><%=record.getTsubuyaki()%></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 userInput.html

```

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

```

```

20:         <input type="button" onclick="location.href='./UserListServlet'"
21:             value="戻る">
22:     </form>
23: </body>
24: </html>

```

ファイル userList.jsp

```

01: <%@ page language="java" contentType="text/html; charset=UTF-8"
02:     pageEncoding="UTF-8"%>
03: <%@ page import="java.util.ArrayList"%>
04: <%@ page import="chapter11.user.*"%>
05: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
06:     "http://www.w3.org/TR/html4/loose.dtd">
07: <html>
08: <head>
09: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10: <title>つぶやきアプリ</title>
11: </head>
12: <body>
13: <jsp:useBean id="UserArrayBean" class="chapter11.user.UserArrayBean"
14:     scope="session" />
15: <h1>つぶやきアプリ</h1>
16: <h2>ユーザ管理</h2>
17: <%=request.getRemoteUser()%> :
18: <a href="LogoutServlet">ログアウト</a>
19: <a href="TsubuyakiListServlet">つぶやきへ戻る</a>
20: <table border="1">
21: <tr>
22: <th>ユーザ ID</th>
23: <th>ロール</th>
24: <td colspan=2><input type="button"
25:     onclick="location.href='userinput.html'" value="ユーザ作成"></td>
26: </tr>
27: <%
28:     ArrayList<UserBean> userArray = UserArrayBean.getUserArray();
29:     for (UserBean record : userArray) {
30: %>
31: <tr>
32: <td><%=record.getUserid()%></td>
33: <td><%=record.getRole()%></td>
34: <td><a href="UserPersonServlet?userid=<%=record.getUserid()%>">編集</a></td>
35: <td><a href="UserDeleteServlet?userid=<%=record.getUserid()%>">削除</a></td>
36: </tr>
37: <%
38:     }
39: %>
40: </table>
41: </body>
42: </html>

```


ファイル userperson.jsp

```
01: <%@ page language="java" contentType="text/html; charset=UTF-8"
02:     pageEncoding="UTF-8"%>
03: <%@ page import="chapter11.parameter.*"%>
04: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
05:     "http://www.w3.org/TR/html4/loose.dtd">
06: <html>
07: <head>
08: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
09: <title>つぶやきアプリ</title>
10: </head>
11: <body>
12: <jsp:useBean id="UserBean" class="chapter11.user.UserBean"
13:     scope="session" />
14: <h1>つぶやきアプリ</h1>
15: <h2>ユーザ情報変更</h2>
16:
17: <form action="UserUpdateServlet" method="post">
18:     ユーザ ID:<input type="hidden" name="userid"
19:         value="<%=UserBean.getUserid()%>"><%=UserBean.getUserid()%><br>
20:     パスワード:<input type="password" name="passwd"
21:         value="<%=UserBean.getPasswd()%>"><br>
22:     <%
23:         String users = "checked";
24:         String admins = "";
25:         if (UserRoleParameters.ROLE_ADMINS.equalsIgnoreCase(UserBean.getRole())) {
26:             users = "";
27:             admins = "checked";
28:         }
29:     %>
30:     ロール:<input type="radio" name="role" value="users" <%=users%>>
31:     <%=UserRoleParameters.ROLE_USERS%>
32:     <input type="radio" name="role" value="admins" <%=admins%>>
33:     <%=UserRoleParameters.ROLE_ADMINS%>
34:     <br><input type="submit" value="変更"> <input type="reset">
35:     <input type="button" onclick="location.href='./UserListServlet'"
36:         value="戻る">
37: </form>
38: </body>
39: </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://java.sun.com/xml/ns/javaee"
04:     xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
05:     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```

06: http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
07: <display-name>STEP3</display-name>
08: <welcome-file-list>
09:   <welcome-file>index.html</welcome-file>
10:   <welcome-file>index.htm</welcome-file>
11:   <welcome-file>index.jsp</welcome-file>
12:   <welcome-file>default.html</welcome-file>
13:   <welcome-file>default.htm</welcome-file>
14:   <welcome-file>default.jsp</welcome-file>
15:   <welcome-file>TsubuyakiListServlet</welcome-file>
16: </welcome-file-list>
17: <security-constraint>
18:   <web-resource-collection>
19:     <web-resource-name>FORM 認証</web-resource-name>
20:     <url-pattern>/*</url-pattern>
21:   </web-resource-collection>
22:   <auth-constraint>
23:     <role-name>admins</role-name>
24:   </auth-constraint>
25: </security-constraint>
26: <security-constraint>
27:   <web-resource-collection>
28:     <web-resource-name>FORM 認証</web-resource-name>
29:     <url-pattern>/TsubuyakiListServlet</url-pattern>
30:     <url-pattern>/TsubuyakiPersonServlet</url-pattern>
31:     <url-pattern>/TsubuyakiInputServlet</url-pattern>
32:     <url-pattern>/LogoutServlet</url-pattern>
33:     <url-pattern>/tsubuyakiinput.html</url-pattern>
34:     <url-pattern>/tsubuyakilist.jsp</url-pattern>
35:     <url-pattern>/tsubuyakipersonlist.jsp</url-pattern>
36:     <url-pattern>/error.jsp</url-pattern>
37:   </web-resource-collection>
38:   <auth-constraint>
39:     <role-name>admins</role-name>
40:     <role-name>users</role-name>
41:   </auth-constraint>
42: </security-constraint>
43: <login-config>
44:   <auth-method>FORM</auth-method>
45:   <form-login-config>
46:     <form-login-page>/login.jsp</form-login-page>
47:     <form-error-page>/loginerror.jsp</form-error-page>
48:   </form-login-config>
49: </login-config>
50: <security-role>
51:   <role-name>users</role-name>
52: </security-role>
53: <security-role>
54:   <role-name>admins</role-name>
55: </security-role>

```

56: </web-app>