
Java/バイブルシリーズ Webアプリケーション構築 Java7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:     protected void doGet(HttpServletRequest request,
21:         HttpServletResponse response)
22:         throws ServletException, IOException {
23:         PrintWriter out = response.getWriter();
24:         out.println("<html>");
25:         out.println("<title>九九の表</title>");
26:         out.println("<body>");
27:         out.println("<h3>九九の表</h3>");
28:         out.println("<table border=¥\"3¥\">");
29:         for (int i = 0; i < 10; i++) {
30:             out.println("<TR>");
31:             for (int j = 0; j < 10; j++) {
32:                 if (i == 0) {
33:                     if (j == 0) {
34:                         out.println("<TD> </TD>");
35:                     } else {
36:                         out.println("<TH> " + j + "</TH>");
37:                     }
38:                 } else {
39:                     if (j == 0) {
40:                         out.println("<TH> " + i + "</TH>");
41:                     } else {
```

```

42:         out.println("<TD align='right'>" + (i * j)
43:             + "</TD>");
44:     }
45: }
46: }
47:     out.println("</TR>");
48: }
49: out.println("</table>");
50: out.println("<p>以上");
51: out.println("</body></html>");
52: }
53: }

```

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:     protected void doGet(HttpServletRequest request,
24:         HttpServletResponse response)
25:         throws ServletException, IOException {
26:         // TODO Auto-generated method stub
27:         PrintWriter out = response.getWriter();
28:         out.println("<html>");

```

```

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

```

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 AccessCnt = 0;
19:
20:     /**
21:      * @see HttpServlet#HttpServlet()
22:      */
23:     public AccessCounterServlet() {
24:         super();

```

```

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

```

練習4ー1 じゃんけん

プロジェクト名 rensyu2_4

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:  protected void doGet(HttpServletRequest request,
23:      HttpServletResponse response)
24:      throws ServletException, IOException {
25:      // TODO Auto-generated method stub
26:      /**
27:       * 人が選んだ手を受け取る
28:       */
29:      String teHito = request.getParameter("Te");
30:
31:      /**
32:       * コンピュータの手を決める
33:       */
34:      double jankenPon = Math.random();
35:      String tePGM = "";
36:      if (jankenPon <= 0.3d) {
37:          tePGM = "g";
38:      } else if (jankenPon <= 0.6d) {
39:          tePGM = "c";
40:      } else {
41:          tePGM = "p";
42:      }
43:
44:      /**
45:       * 勝負の判定
46:       */
47:      String kekka = "";
48:      if (tePGM.equals(teHito)) {
49:          kekka = "あいこ";
50:      } else if ((tePGM.equals("g")) && (teHito.equals("c"))) {
51:          kekka = "PGM";
52:      } else if ((tePGM.equals("c")) && (teHito.equals("p"))) {
53:          kekka = "PGM";
54:      } else if ((tePGM.equals("p")) && (teHito.equals("g"))) {
55:          kekka = "PGM";
56:      } else {
57:          kekka = "人";
58:      }
59:      if (kekka.equals("あいこ")){
60:          kekka = "引き分け";
61:      }else if (kekka.equals("PGM")){
62:          kekka = "プログラムの勝ち";
63:      }else{
64:          kekka = "あなたが勝ち";
65:      }
66:
67:      PrintWriter out = response.getWriter();
68:      out.println("<html>");

```

```

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

```

ファイル Janken.html

```

01: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:     "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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:     protected void doGet(HttpServletRequest request,
027:         HttpServletResponse response)
028:         throws ServletException, IOException {
029:         // TODO Auto-generated method stub
030:         /**
031:          * 人が選んだ手を受け取る
032:          */
033:         String teHito = request.getParameter("Te");
034:
035:         /**
036:          * コンピュータの手を決める
037:          */
038:         double jankenPon = Math.random();
039:         String tePGM = "";
040:         if (jankenPon <= 0.3d) {
041:             tePGM = "g";
042:         } else if (jankenPon <= 0.6d) {
043:             tePGM = "c";
044:         } else {
045:             tePGM = "p";
046:         }
047:
```

```

048:     kaisu++;
049:     /**
050:      * 勝負の判定
051:      */
052:     String kekka = "";
053:     if (tePGM.equals(teHito)) {
054:         kekka = "あいこ";
055:     } else if ((tePGM.equals("g")) && (teHito.equals("c"))) {
056:         kekka = "PGM";
057:     } else if ((tePGM.equals("c")) && (teHito.equals("p"))) {
058:         kekka = "PGM";
059:     } else if ((tePGM.equals("p")) && (teHito.equals("g"))) {
060:         kekka = "PGM";
061:     } else {
062:         kekka = "人";
063:     }
064:     if (kekka.equals("あいこ")) {
065:         kekka = "引き分け";
066:     } else if (kekka.equals("PGM")) {
067:         winPGM++;
068:         kekka = "プログラムの勝ち";
069:     } else {
070:         winHito++;
071:         kekka = "あなたが勝ち";
072:     }
073:
074:     PrintWriter out = response.getWriter();
075:     out.println("<html>");
076:     out.println("<head>");
077:     out.println("<title>じゃんけん結果</title>");
078:     out.println("</head>");
079:     out.println("<body>");
080:     out.println("<h1>じゃんけん結果 (" + kaisu + "回目) </h1>");
081:     out.println("あなたが選んだ手 : " + teHenkan(teHito) + "("
082:         + winHito + "勝)<br>");
083:     out.println("プログラムが選んだ手 : " + teHenkan(tePGM) + "("
084:         + winPGM + "勝)<br>");
085:     out.println("勝負の結果は、" + kekka + "でした！");
086:     out.println("<p><a href=¥\"Janken2.html¥\">もう一度やる</a>");
087:     out.println("</body></html>");
088: }
089:
090: String teHenkan(String te) {
091:     switch (te) {
092:         case "g":
093:             return "グー";
094:         case "c":
095:             return "チョキ";
096:         default:
097:             return "パー";

```

```

098:     }
099:
100:     }
101: }

```

ファイル Janken2.html

```

01: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:     "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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">チョキ</A>
13: <p><A href="JankenServlet2?Te=p">パー</A>
14: </body>
15: </html>

```

練習5 数当て

プロジェクト名 rensyu2.5

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:  protected void doGet(HttpServletRequest request,
26:   HttpServletResponse response)
27:   throws ServletException, IOException {
28:   // TODO Auto-generated method stub
29:   if (kaisu == 0) {
30:     targetKazu = (int) (Math.random() * 9d) + 1;
31:   }
32:   kaisu++;
33:   String kazu_s = request.getParameter("kazu");
34:   int EnteredKazu = Integer.parseInt(kazu_s);
35:
36:   String kekka = "はずれ～！";
37:   if (EnteredKazu == targetKazu) {
38:     kekka = "当たり";
39:   }
40:
41:   PrintWriter out = response.getWriter();
42:   out.println("<html>");
43:   out.println("<head>");
44:   out.println("<title>数当てゲーム結果</title>");
45:   out.println("</head>");
46:   out.println("<body>");
47:   out.println("<h1>数当てゲーム結果</h1>");
48:   out.println(kaisu + "回目 あなたが選んだ数 : " + EnteredKazu + "<br>");
49:   out.println("<br>");
50:   out.println("結果は、" + kekka);
51:   out.println("<p><a href='\"kazuate.html\"'>もう一度やる</a>");
52:   out.println("</body></html>");
53:
54:   if (EnteredKazu == targetKazu) {
55:     kaisu = 0;
56:   }
57:
58: }
59: }

```

ファイル kazuate.html

```

01: <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:   "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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）を当ててみましょう。
12: <p>
13: <a href="KazuServlet?kazu=1">1</a>&nbsp;
14: <a href="KazuServlet?kazu=2">2</a>&nbsp;
15: <a href="KazuServlet?kazu=3">3</a>&nbsp;
16: <a href="KazuServlet?kazu=4">4</a>&nbsp;
17: <a href="KazuServlet?kazu=5">5</a>&nbsp;
18: <a href="KazuServlet?kazu=6">6</a>&nbsp;
19: <a href="KazuServlet?kazu=7">7</a>&nbsp;
20: <a href="KazuServlet?kazu=8">8</a>&nbsp;
21: <a href="KazuServlet?kazu=9">9</a>&nbsp;
22: </body>
23: </html>

```

練習6 単純な電卓

プロジェクト名 rensyu2.6

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:     /**
18:      * @see HttpServlet#HttpServlet()
19:      */
20:     public KeisanServlet() {
21:         super();
22:         // TODO Auto-generated constructor stub
23:     }
24:
25:     /**
26:      * @see HttpServlet#doGet(HttpServletRequest request,

```

```

27:      *      HttpServletResponse response)
28:      */
29:  protected void doGet(HttpServletRequest request,
30:      HttpServletResponse response)
31:      throws ServletException, IOException {
32:      // TODO Auto-generated method stub
33:      String sKazu1 = request.getParameter("kazu1");
34:      int kazu1 = Integer.parseInt(sKazu1);
35:      String sKazu2 = request.getParameter("kazu2");
36:      int kazu2 = Integer.parseInt(sKazu2);
37:      String syori = request.getParameter("syurui");
38:      PrintWriter out = response.getWriter();
39:      out.println("<html>");
40:      out.println("<head><title>四則演算結果</title></head>");
41:      out.println("<body>");
42:      out.println("<h1>四則演算の結果</h1>");
43:      out.println("<blockquote><table>");
44:      out.println("<tr><td>数字 1 つ目<td><td>数字 2 つ目<td><td>結果");
45:      out.println("<tr align=¥\"center¥\"><td>\" + kazu1 + \"<td>");
46:      switch (syori) {
47:      case "tasu": out.print("+");break;
48:      case "hiku": out.print("-");break;
49:      case "kake": out.print("x");break;
50:      case "wari": out.print("÷");break;
51:      }
52:      out.println("<td>\" + kazu2 + \"<td>=<td>");
53:      switch (syori) {
54:      case "tasu":out.print(kazu1 + kazu2);break;
55:      case "hiku":out.print(kazu1 - kazu2);break;
56:      case "kake":out.print(kazu1 * kazu2);break;
57:      case "wari":out.print(kazu1 / kazu2);break;
58:      }
59:      out.println("</table></blockquote>");
60:      out.println("</body></html>");
61:  }
62:
63: }

```

ファイル Keisan.html

```

01:  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:      "http://www.w3.org/TR/html4/loose.dtd">
03:  <html>
04:  <head>
05:  <meta http-equiv="Content-Type" content="text/html; 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 align="center">数字 1 つ目<br>
13:     <INPUT size="5" type="text" name="kazu1" value="30">
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">÷
18:     <td align="center">数字 2 つ目<br>
19:     <INPUT size="5" type="text" name="kazu2" value="20">
20:     <td align="center">
21:     <INPUT type="submit" name="SyoriHajime" value="">
22: </table>
23: </form>
24: </blockquote>
25: </body>
26: </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;  
02:   charset=UTF-8" 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 height=30 width=30 bgcolor=¥"#"  
23:           + colorHTML + "¥"><br></td>");  
24:       }  
25:     <%>  
26:   </tr>  
27: </table>  
28: </body>  
29: </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 height=50 width=50 bgcolor=¥"#"
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: import javax.servlet.http.Cookie;
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 getCookie
14:  */
15: @WebServlet("/ReadCookieServlet")
16: public class ReadCookieServlet 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) throws ServletException,
26:         IOException {
27:         response.setContentType("text/html;charset=¥utf-8¥");
28:         PrintWriter out = response.getWriter();
29:         boolean bl1 = true;
30:         String strcnt;
31:         int intcnt;
32:         out.println("<html><head><title>クッキー情報</title></head><body>");
33:         out.println("<h2>クッキー情報</h2>");
34:         Cookie[] cookies = request.getCookies();
35:         if (cookies != null) {
36:             for (Cookie cook : cookies) {
37:                 if (cook.getName().equals("myname")) {
38:                     out.println("<p>" + cook.getValue() + "さん" + "</p>");
39:                 }
40:                 if (cook.getName().equals("cookiecnt")) {
41:                     strcnt = cook.getValue();
```

```

42:         intcnt = Integer.parseInt(cook.getValue()) + 1;
43:         strcnt = Integer.toString(intcnt);
44:         out.println("<p>" + "こんにちは" + strcnt + "回目の訪問ですね" + "</p>");
45:         Cookie cookcnt = new Cookie("cookiecnt", strcnt);
46:         // クッキー保存時間 60秒 *60分*24時間*30日の1ヶ月
47:         cookcnt.setMaxAge(60 * 60 * 24 * 30);
48:         response.addCookie(cookcnt);
49:         bl1 = false;
50:     }
51: }
52: }
53: if (bl1) {
54:     out.println("<p>初めての訪問ですね。</p>");
55:     String url2 = "/rensyu4_1/CookieInfo.jsp";
56:     out.println("<p>" + "My 情報を登録しましょう" + "</p>");
57:     out.println("<a href=¥" + url2 + "¥>登録</a>");
58: }
59: out.println("</body></html>");
60: }
61: }
62: }

```

ファイル WriteCookie.java

```

01: package pac1;
02:
03: import java.io.IOException;
04: import java.io.PrintWriter;
05:
06: import javax.servlet.ServletException;
07: import javax.servlet.annotation.WebServlet;
08: import javax.servlet.http.Cookie;
09: import javax.servlet.http.HttpServlet;
10: import javax.servlet.http.HttpServletRequest;
11: import javax.servlet.http.HttpServletResponse;
12:
13: /**
14:  * Servlet implementation class addCookie
15:  */
16: @WebServlet("/WriteCookie")
17: public class WriteCookie 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 doPost(HttpServletRequest request,
26:         HttpServletResponse response) throws ServletException,
27:         IOException {

```

```

28: response.setContentType("text/html;charset=utf-8");
29: PrintWriter out = response.getWriter();
30: boolean bl1 = false;
31:
32: // クッキー保存時間 60 秒 *60 分*24 時間*30 日の 1 ヶ月
33: int setmaxtime = 60 * 60 * 24 * 30;
34:
35: out.println("<html><head><title>クッキー書き込み</title></head><body>");
36:
37: // パラメータの取得
38: String myno = request.getParameter("myno");
39: String myname = request.getParameter("myname");
40: String mymail = request.getParameter("mymail");
41: try {
42:     if (myno != null && myno.length() != 0) {
43:         bl1 = true;
44:     }
45:     Integer.parseInt(myno);
46: } catch (NumberFormatException e) {
47:     bl1 = false;
48: }
49: if (bl1) {
50:     // クッキー情報書き込み
51:     Cookie cookmyno = new Cookie("myno", myno);
52:     cookmyno.setMaxAge(setmaxtime);
53:     response.addCookie(cookmyno);
54:
55:     Cookie cookmyname = new Cookie("myname", myname);
56:     cookmyname.setMaxAge(setmaxtime);
57:     response.addCookie(cookmyname);
58:
59:     Cookie cookemail = new Cookie("mymail", mymail);
60:     cookemail.setMaxAge(setmaxtime);
61:     response.addCookie(cookemail);
62:
63:     Cookie cookcnt = new Cookie("cookiecnt", "1");
64:     cookcnt.setMaxAge(setmaxtime);
65:     response.addCookie(cookcnt);
66:
67:     // クッキー情報の表示
68:     out.println("<h2>クッキー情報</h2>");
69:     out.println("<p>" + "MyNo: " + request.getParameter("myno")
70:         + "</p>");
71:     out.println("<p>" + "MyName: " + request.getParameter("myname")
72:         + "</p>");
73:     out.println("<p>" + "MyMail: " + request.getParameter("mymail")
74:         + "</p>");
75:     out.println("<p>" + "クッキーを書きました" + "</p>");
76: } else {
77:     out.println("<h2>クッキー情報</h2>");
78:

```

```

79:         out.println("<p>MyNo を入力してください</p>");
80:         String url2 = "/rensyu4_1/CookieInfo.jsp";
81:         out.println("<a href=¥\"\" + url2 + \"¥\">戻る</a>");
82:     }
83:     out.println("</body></html>");
84: }
85: }

```

ファイル ClearCookie.java

```

01: package pac1;
02:
03: import java.io.IOException;
04: import java.io.PrintWriter;
05: import javax.servlet.http.Cookie;
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 resetCookie
14:  */
15: @WebServlet("/ClearCookie")
16: public class ClearCookie 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) throws ServletException,
26:         IOException {
27:         response.setContentType("text/html;charset=¥utf-8¥");
28:         PrintWriter out = response.getWriter();
29:
30:         Cookie cookmyno = new Cookie("myno", "");
31:         cookmyno.setMaxAge(0);
32:         response.addCookie(cookmyno);
33:
34:         Cookie cookmyname = new Cookie("myname", "");
35:         cookmyname.setMaxAge(0);
36:         response.addCookie(cookmyname);
37:
38:         Cookie cookemail = new Cookie("mymail", "");
39:         cookemail.setMaxAge(0);
40:         response.addCookie(cookemail);

```

```

41:
42:     Cookie cookcnt = new Cookie("cookiecnt", "");
43:     cookcnt.setMaxAge(0);
44:     response.addCookie(cookcnt);
45:
46:     out.println("<html><head><title>クッキーリセット情報</title></head><body>");
47:     out.println("<h2>クッキー情報</h2>");
48:     out.println("<p>クッキー情報は削除しました</p>");
49:     out.println("</body></html>");
50: }
51: }

```

ファイル CookieInfo.js

```

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>クッキー情報登録</title>
09: </head>
10: <body>
11: <h3>My 情報登録</h3>
12: <form method="POST" action="WriteCookie" >
13:     MyNo
14:     <input type="text" name="myno" size="10" value=""/>
15:     MyName
16:     <input type="text" name="myname" size="20" value=""/>
17:     MyMail
18:     <input type="text" name="mymail" size="20" value=""/>
19:     <input type="submit" value="登録" />
20: </form>
21: </body>
22: </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 PaymentRecord.java

```
01: package pac1;
02:
03: public class PaymentRecord {
04:     int payment;    // 入金額
05:     String comment; // コメント
06:
07:     public void setPayment(int value) {
08:         payment = value;
09:     }
10:
11:     public int getPayment() {
12:         return payment;
13:     }
14:
15:     public void setComment(String value) {
16:         comment = value;
17:     }
18:
19:     public String getComment() {
20:         return comment;
21:     }
22:
23: }
```

ファイル PettyCashbookBean.java

```
01: package pac1;
02:
03: import java.io.Serializable;
04: import java.util.ArrayList;
05:
06: public class PettyCashbookBean implements Serializable {
07:
08:     private ArrayList<PaymentRecord> paymentHistoryArray
09:         = new ArrayList<PaymentRecord>(); // 貯金の履歴
10:
11:     public PettyCashbookBean() {
12:     }
13:
14:     public ArrayList<PaymentRecord> getPaymentHistoryArray() {
15:         return paymentHistoryArray;
16:     }
17: }
```

```

16:     }
17:
18:     public void addPaymentHistoryArray(PaymentRecord payment) {
19:         paymentHistoryArray.add(payment);
20:     }
21:
22:     public int getTotalPayment() {
23:         int total = 0;
24:
25:         for (PaymentRecord rec : paymentHistoryArray) {
26:             total += rec.getPayment();
27:         }
28:         return total;
29:     }
30:
31: }

```

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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02: "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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.PaymentRecord" %>
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<PaymentRecord> paymentHistoryArray =
21:      pettyCashbookBean.getPaymentHistoryArray();
22:
23:      int i = 1;
24:      for(PaymentRecord 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:     PettyCashbookBean pettyCashbookBean = new PettyCashbookBean();
16:
17:     public PettyCashbookServlet() {
18:         super();
19:     }
20:
21:     @Override
22:     protected void doGet(HttpServletRequest request,
23:         HttpServletResponse response) throws ServletException,
24:         IOException {
25:
26:         int payment;    // 入金額
27:         String comment; // コメント
28:
29:         /* Formから入力された値の取り出し */
30:         try {
31:             payment = Integer.parseInt(request.getParameter("Payment"));
32:             comment = request.getParameter("Comment");
33:
34:         } catch (NumberFormatException e) {
35:             getServletContext().getRequestDispatcher("/ErrorInput.html")
36:                 .forward(request, response);
37:             return;
38:         }
39:
40:         /* 入金レコードの作成 */
41:         PaymentRecord paymentRec = new PaymentRecord();
42:         paymentRec.setPayment(payment);
43:         paymentRec.setComment(comment);
44:
45:         /* Beans 実行 */
46:         pettyCashbookBean.addPaymentHistoryArray(paymentRec); // 入金レコードを Beans へ追加
47:
48:         /* セッションに Beans を登録し、JSP で出力 */
49:         HttpSession session = request.getSession();
50:         session.setAttribute("pettyCashbookBean", pettyCashbookBean);
51:         getServletContext().getRequestDispatcher(
52:             "/PettyCashbook_output.jsp").forward(request, response);
53:
54:     }
55: }

```

第6章 練習問題の模範解答

練習1 科目表

プロジェクト名 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。

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_b", "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.getString("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。

プロジェクト名 rensyu6_3

ファイル index.html

```
01:      <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:      "http://www.w3.org/TR/html4/loose.dtd">
03:      <html>
04:      <head>
05:      <meta http-equiv="Content-Type" content="text/html; 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

```
STEP1
├── src
│   ├── chapter11
│   │   ├── TsubuyakiArrayBean.java
│   │   ├── TsubuyakiBean.java
│   │   ├── TsubuyakiInputServlet.java
│   │   ├── TsubuyakiListServlet.java
│   │   └── TsubuyakiPersonServlet.java
│   ├── dao
│   │   ├── DAOBase.java
│   │   └── TsubuyakiDAO.java
│   ├── exception
│   │   ├── DatabaseException.java
│   │   ├── ParameterException.java
│   │   └── SystemException.java
│   ├── parameter
│   │   ├── DAOParameters.java
│   │   ├── ExceptionParameters.java
│   │   └── TsubuyakiParameters.java
│   └── filter
│       └── ExampleFilter.java
└── WebContent
    ├── error.jsp
    ├── TsubuyakiInput.html
    ├── TsubuyakiList.jsp
    └── TsubuyakiPersonList.jsp
```

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 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:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:         try {
28:             String name = request.getParameter("name");
29:             String tsubuyaki = request.getParameter("tsubuyaki");
30:             Date date = new Date();
31:
32:             TsubuyakiBean tsubuyakiBean = new TsubuyakiBean();
33:             tsubuyakiBean.setName(name);
34:             tsubuyakiBean.setTsubuyaki(tsubuyaki);
35:             tsubuyakiBean.setDatetime(date);
36:
37:             TsubuyakiDAO dao = new TsubuyakiDAO();
38:             dao.setTsubuyaki(tsubuyakiBean);
39:
40:             response.sendRedirect("TsubuyakiListServlet");
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:
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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         TsubuyakiArrayBean tsubuyakiArray;
28:
29:         try {
30:             TsubuyakiDAO dao = new TsubuyakiDAO();
31:             tsubuyakiArray = dao.getTsubuyakiArray();
32:             HttpSession session = request.getSession();
33:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
34:             getServletContext().getRequestDispatcher("/TsubuyakiList.jsp").forward(
35:                 request, response);
36:         } catch (SystemException e) {
37:             e.printStackTrace();
38:             HttpSession session = request.getSession();
39:             session.setAttribute("Except", e);
40:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
41:                 response);
42:         } catch (DatabaseException e) {
43:             e.printStackTrace();
44:             HttpSession session = request.getSession();
45:             session.setAttribute("Except", e);
46:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:                 response);
48:         }
49:     }
50: }
51:

```

ファイル 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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         TsubuyakiArrayBean tsubuyakiArray;
28:
29:         try {
30:             String name = request.getParameter("name");
31:
32:             TsubuyakiDAO dao = new TsubuyakiDAO();
33:             tsubuyakiArray = dao.getPersonTsubuyakiArray(name);
34:             HttpSession session = request.getSession();
35:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
36:             getServletContext().getRequestDispatcher("/TsubuyakiPersonList.jsp")
37:                 .forward(request, response);
38:         } catch (SystemException e) {
39:             e.printStackTrace();
40:             HttpSession session = request.getSession();
41:             session.setAttribute("Except", e);
42:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:                 response);
44:         } catch (DatabaseException e) {
45:             e.printStackTrace();
46:             HttpSession session = request.getSession();
47:             session.setAttribute("Except", e);
48:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
49:                 response);
50:         }
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:         }
```

```

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

```

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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02: "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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。

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:         protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:             throws ServletException, IOException {
26:             try {
27:                 String name = request.getParameter("name");
28:
29:                 TsubuyakiDAO dao = new TsubuyakiDAO();
30:                 dao.deleteTsubuyaki(name);
31:
32:                 response.sendRedirect("TsubuyakiListServlet");
33:             } catch (SystemException e) {
34:                 e.printStackTrace();
35:                 HttpSession session = request.getSession();
36:                 session.setAttribute("Except", e);
37:                 getServletContext().getRequestDispatcher("/error.jsp").forward(request,

```

```

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

```

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:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:         try {
28:             String name = request.getParameter("name");
29:             String tsubuyaki = request.getParameter("tsubuyaki");
30:             Date date = new Date();
31:
32:             TsubuyakiBean tsubuyakiBean = new TsubuyakiBean();
33:             tsubuyakiBean.setName(name);
34:             tsubuyakiBean.setTsubuyaki(tsubuyaki);
35:             tsubuyakiBean.setDatetime(date);
36:

```

```

37:         TsubuyakiDAO dao = new TsubuyakiDAO();
38:         dao.setTsubuyaki(tsubuyakiBean);
39:
40:         response.sendRedirect("TsubuyakiListServlet");
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 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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         TsubuyakiArrayBean tsubuyakiArray;

```

```

28:
29:     try {
30:         TsubuyakiDAO dao = new TsubuyakiDAO();
31:         tsubuyakiArray = dao.getTsubuyakiArray();
32:         HttpSession session = request.getSession();
33:         session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
34:         getServletContext().getRequestDispatcher("/TsubuyakiList.jsp").forward(
35:             request, response);
36:     } catch (SystemException e) {
37:         e.printStackTrace();
38:         HttpSession session = request.getSession();
39:         session.setAttribute("Except", e);
40:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
41:             response);
42:     } catch (DatabaseException e) {
43:         e.printStackTrace();
44:         HttpSession session = request.getSession();
45:         session.setAttribute("Except", e);
46:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:             response);
48:     }
49: }
50: }

```

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:         protected void doGet(HttpServletRequest request, HttpServletResponse response)

```

```

25:         throws ServletException, IOException {
26:
27:         TsubuyakiArrayBean tsubuyakiArray;
28:
29:         try {
30:             String name = request.getParameter("name");
31:
32:             TsubuyakiDAO dao = new TsubuyakiDAO();
33:             tsubuyakiArray = dao.getPersonTsubuyakiArray(name);
34:             HttpSession session = request.getSession();
35:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
36:             getServletContext().getRequestDispatcher("/TsubuyakiPersonList.jsp")
37:                 .forward(request, response);
38:         } catch (SystemException e) {
39:             e.printStackTrace();
40:             HttpSession session = request.getSession();
41:             session.setAttribute("Except", e);
42:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:                 response);
44:         } catch (DatabaseException e) {
45:             e.printStackTrace();
46:             HttpSession session = request.getSession();
47:             session.setAttribute("Except", e);
48:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
49:                 response);
50:         }
51:     }
52: }
53: }

```

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 = "root";
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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:     "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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>

```

プロジェクト名 STEP3

```
STEP3
├── src
│   ├── chapter11
│   │   ├── LogoutServlet.java
│   │   ├── dao
│   │   │   ├── DAOBase.java
│   │   │   ├── TsubuyakiDAO.java
│   │   │   └── UserDAO.java
│   │   ├── exception
│   │   │   ├── DatabaseException.java
│   │   │   ├── ParameterException.java
│   │   │   └── SystemException.java
│   │   ├── parameter
│   │   │   ├── DAOParameters.java
│   │   │   ├── ExceptionParameters.java
│   │   │   ├── TsubuyakiParameters.java
│   │   │   └── UserRoleParameters.java
│   │   ├── tsubuyaki
│   │   │   ├── TsubuyakiArrayBean.java
│   │   │   ├── TsubuyakiBean.java
│   │   │   ├── TsubuyakiDeleteServlet.java
│   │   │   ├── TsubuyakiInputServlet.java
│   │   │   ├── TsubuyakiListServlet.java
│   │   │   └── TsubuyakiPersonServlet.java
│   │   └── user
│   │       ├── UserArrayBean.java
│   │       ├── UserBean.java
│   │       ├── UserDeleteServlet.java
│   │       ├── UserInputServlet.java
│   │       ├── UserListServlet.java
│   │       ├── UserPersonServlet.java
│   │       └── UserUpdateServlet.java
│   └── filter
│       └── ExampleFilter.java
└── WebContent
    ├── error.jsp
    ├── login.jsp
    ├── loginerror.jsp
    └── TsubuyakiInput.html
```

```
| 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, SystemException {
18:         Statement stmt = null;
19:         UserArrayBean userArray = new UserArrayBean();
20:         this.open();
21:         try {
22:             stmt = con.createStatement();
23:             ResultSet rs = stmt.executeQuery(UserRoleParameters.SQL_SELECT_ALL);
24:             while (rs.next()) {
25:                 UserBean record = new UserBean();
26:                 record.setUserid(rs.getString(UserRoleParameters.USERID));
27:                 record.setRole(rs.getString(UserRoleParameters.ROLE));
28:                 userArray.addUser(record);
29:             }
30:         } catch (SQLException e) {
31:             throw new DatabaseException(
32:                 ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
33:         } finally {
34:             this.close(stmt);
35:         }
36:         return userArray;
37:     }
38:
39:     public UserBean getUser(String userid) throws DatabaseException,
40:         SystemException {
41:         UserBean record = null;
42:         PreparedStatement stmt = null;
43:         this.open();
44:         try {
45:             stmt = con.prepareStatement(UserRoleParameters.SQL_SELECT_PERSON);
46:             stmt.setString(1, userid);
47:             ResultSet rs = stmt.executeQuery();
48:             rs.next();
49:             record = new UserBean();
```

```

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

```

```

100:         throw new DatabaseException(
101:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
102:     } finally {
103:         this.close(stmt);
104:     }
105:
106: }
107:
108: public void deleteUser(String userid) throws DatabaseException,
109:     SystemException {
110:     PreparedStatement stmt = null;
111:     this.open();
112:     try {
113:         stmt = con.prepareStatement(UserRoleParameters.SQL_DELETE_ROLE);
114:         stmt.setString(1, userid);
115:         stmt.executeUpdate();
116:         stmt = con.prepareStatement(UserRoleParameters.SQL_DELETE_USER);
117:         stmt.setString(1, userid);
118:         stmt.executeUpdate();
119:     } catch (SQLException e) {
120:         throw new DatabaseException(
121:             ExceptionParameters.DATABASE_CONNECTION_EXCEPTION_MESSAGE, e);
122:     } finally {
123:         this.close(stmt);
124:     }
125:
126: }
127:
128: }

```

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 = "root";
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 PASSWD=? 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 = "passwd";
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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:         try {
27:             String name = request.getParameter("name");
28:
29:             TsubuyakiDAO dao = new TsubuyakiDAO();
30:             dao.deleteTsubuyaki(name);
31:
32:             response.sendRedirect("TsubuyakiListServlet");
33:         } catch (SystemException e) {
34:             e.printStackTrace();
35:             HttpSession session = request.getSession();
36:             session.setAttribute("Except", e);
37:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
38:                 response);
39:         } catch (DatabaseException e) {
40:             e.printStackTrace();
41:             HttpSession session = request.getSession();
42:             session.setAttribute("Except", e);
43:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
44:                 response);
45:         }
46:     }
47:
48: }

```

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:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
26:         throws ServletException, IOException {
27:         try {
28:             String name = request.getParameter("name");
29:             String tsubuyaki = request.getParameter("tsubuyaki");
30:             Date date = new Date();
31:
32:             TsubuyakiBean tsubuyakiBean = new TsubuyakiBean();
33:             tsubuyakiBean.setName(name);
34:             tsubuyakiBean.setTsubuyaki(tsubuyaki);
35:             tsubuyakiBean.setDatetime(date);
36:
37:             TsubuyakiDAO dao = new TsubuyakiDAO();
38:             dao.setTsubuyaki(tsubuyakiBean);
39:
40:             response.sendRedirect("TsubuyakiListServlet");
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 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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         TsubuyakiArrayBean tsubuyakiArray;
28:
29:         try {
30:             TsubuyakiDAO dao = new TsubuyakiDAO();
31:             tsubuyakiArray = dao.getTsubuyakiArray();
32:             HttpSession session = request.getSession();
33:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
34:             getServletContext().getRequestDispatcher("/TsubuyakiList.jsp").forward(
35:                 request, response);
36:         } catch (SystemException e) {
37:             e.printStackTrace();
38:             HttpSession session = request.getSession();
39:             session.setAttribute("Except", e);
40:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
41:                 response);
42:         } catch (DatabaseException e) {
43:             e.printStackTrace();
44:             HttpSession session = request.getSession();
45:             session.setAttribute("Except", e);
46:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:                 response);
48:         }
49:     }
50: }

```

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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         TsubuyakiArrayBean tsubuyakiArray;
28:
29:         try {
30:             String name = request.getParameter("name");
31:
32:             TsubuyakiDAO dao = new TsubuyakiDAO();
33:             tsubuyakiArray = dao.getPersonTsubuyakiArray(name);
34:             HttpSession session = request.getSession();
35:             session.setAttribute("TsubuyakiArrayBean", tsubuyakiArray);
36:             getServletContext().getRequestDispatcher("/TsubuyakiPersonList.jsp")
37:                 .forward(request, response);
38:         } catch (SystemException e) {
39:             e.printStackTrace();
40:             HttpSession session = request.getSession();
41:             session.setAttribute("Except", e);
42:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:                 response);
44:         } catch (DatabaseException e) {
45:             e.printStackTrace();
46:             HttpSession session = request.getSession();
47:             session.setAttribute("Except", e);
48:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
49:                 response);
50:         }
51:     }
52: }
53: }

```

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: protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:     throws ServletException, IOException {
26:     try {
27:         String userid = request.getParameter("userid");
28:
29:         UserDao dao = new UserDao();
30:         dao.deleteUser(userid);
31:
32:         response.sendRedirect("UserListServlet");
33:     } catch (SystemException e) {
34:         e.printStackTrace();
35:         HttpSession session = request.getSession();
36:         session.setAttribute("Except", e);
37:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
38:             response);
39:     } catch (DatabaseException e) {
40:         e.printStackTrace();
41:         HttpSession session = request.getSession();
42:         session.setAttribute("Except", e);
43:         getServletContext().getRequestDispatcher("/error.jsp").forward(request,
44:             response);
45:     }
46: }
47:
48: }

```

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:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:         try {
27:             String userid = request.getParameter("userid");
28:             String passwd = request.getParameter("passwd");
29:             String role = request.getParameter("role");
30:
31:             UserBean userBean = new UserBean();
32:             userBean.setUserid(userid);
33:             userBean.setPasswd(passwd);
34:             userBean.setRole(role);
35:
36:             UserDAO dao = new UserDAO();
37:             dao.addUser(userBean);
38:
39:             response.sendRedirect("UserListServlet");
40:         } catch (SystemException e) {
41:             e.printStackTrace();
42:             HttpSession session = request.getSession();
43:             session.setAttribute("Except", e);
44:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
45:                 response);
46:         } catch (DatabaseException e) {
47:             e.printStackTrace();
48:             HttpSession session = request.getSession();
49:             session.setAttribute("Except", e);
50:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
51:                 response);
52:         }
53:     }
54:
55: }

```

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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         UserArrayBean userArray;
28:
29:         try {
30:             UserDAO dao = new UserDAO();
31:             userArray = dao.getUserArray();
32:             HttpSession session = request.getSession();
33:             session.setAttribute("UserArrayBean", userArray);
34:             getServletContext().getRequestDispatcher("/UserList.jsp").forward(
35:                 request, response);
36:         } catch (SystemException e) {
37:             e.printStackTrace();
38:             HttpSession session = request.getSession();
39:             session.setAttribute("Except", e);
40:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
41:                 response);
42:         } catch (DatabaseException e) {
43:             e.printStackTrace();
44:             HttpSession session = request.getSession();
45:             session.setAttribute("Except", e);
46:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
47:                 response);
48:         }
49:     }
50: }

```

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:     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         UserBean user = null;
28:
29:         try {
30:             String userid = request.getParameter("userid");
31:
32:             UserDAO dao = new UserDAO();
33:             user = dao.getUser(userid);
34:             HttpSession session = request.getSession();
35:             session.setAttribute("UserBean", user);
36:             getServletContext().getRequestDispatcher("/UserPerson.jsp").forward(
37:                 request, response);
38:         } catch (SystemException e) {
39:             e.printStackTrace();
40:             HttpSession session = request.getSession();
41:             session.setAttribute("Except", e);
42:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
43:                 response);
44:         } catch (DatabaseException e) {
45:             e.printStackTrace();
46:             HttpSession session = request.getSession();
47:             session.setAttribute("Except", e);
48:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
49:                 response);
50:         }
51:     }
52: }
53:

```

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:     protected void doPost(HttpServletRequest request, HttpServletResponse response)
25:         throws ServletException, IOException {
26:
27:         UserBean user = null;
28:
29:         try {
30:             String userid = request.getParameter("userid");
31:             String passwd = request.getParameter("password");
32:             String role = request.getParameter("role");
33:
34:             user = new UserBean();
35:             user.setUserid(userid);
36:             user.setPasswd(passwd);
37:             user.setRole(role);
38:
39:             UserDAO dao = new UserDAO();
40:             dao.updateUser(user);
41:
42:             response.sendRedirect("UserListServlet");
43:         } catch (SystemException e) {
44:             e.printStackTrace();
45:             HttpSession session = request.getSession();
46:             session.setAttribute("Except", e);
47:             getServletContext().getRequestDispatcher("/error.jsp").forward(request,
48:                 response);
49:         } catch (DatabaseException e) {
```

```

50:         e.printStackTrace();
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 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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02:     "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02: "http://www.w3.org/TR/html4/loose.dtd">
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; 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:     </p>
23: </form>
24: </body>
</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="password"
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>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>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>

```