

```

1: package jp.ict.aso.model;
2:
3: import java.io.Serializable;
4:
5: public class CalendarHtmlBean implements Serializable {
6:     private int year;
7:     private int month;
8:     private int week;
9:     private int end;
10:    private String weekStr;
11:    private String resultCalendar;
12:    public CalendarHtmlBean() { }
13:    public CalendarHtmlBean(int year, int month) {
14:        this.year=year;
15:        this.month=month;
16:        resultCalendar="";
17:    }
18:    public String getResultCalendar() {
19:        return resultCalendar;
20:    }
21:    public int getYear() {
22:        return year;
23:    }
24:    public int getMonth() {
25:        return month;
26:    }
27:    public int getWeek() {
28:        return week;
29:    }
30:    public int getEnd() {
31:        return end;
32:    }
33:    public void setYear(int year) {
34:        this.year=year;
35:    }
36:    public void setMonth(int month) {
37:        this.month=month;
38:    }
39:    public String getWeekStr() {
40:        return weekStr;
41:    }
42:    // カレンダーの作成
43:    // (仮引数) y:西暦年 m:月 (返却値) なし
44:    public void createCalendar() {
45:        int e, w;
46:        int y=year;
47:        int m=month;
48:        w = week_of_day(y, m, 1); // 月初(1日)の曜日を求める
49:        e = month_last_day(y, m); // 月末日を求める
50:        week=w;
51:        end=e;
52:        //見出し表示
53:        resultCalendar=resultCalendar+"<br>"+getYear()+"年"+getMonth()+"月";
54:
55:        //カレンダー表示
56:        resultCalendar=resultCalendar+"<table border=1>";
57:
58:        resultCalendar=resultCalendar+"<tr><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th><font color=blue>土</font></th><th><font color=red>日</font></th></tr>";
59:        resultCalendar=resultCalendar+"<tr>";
60:
61:        if(w==0) { //1日が日曜日になるときはweek=0なので特別に空打ち
62:            for(int k=1;k<7;k++) {

```

```

63:         resultCalendar=resultCalendar+"<td> </td>";
64:     }
65: }
66: for (int i = 1; i <= w-1; i++) { // 1日まで空白で埋める
67:     resultCalendar=resultCalendar+"<td> </td>";
68: }
69: for (int j = 1; j <= e; j++) { // 最終日まで表示する
70:     //祝祭日は赤
71:     if( is_holiday(m, j) ){
72:         resultCalendar=resultCalendar+"<td align=right><font
color=red>"+j+"</font></td>";
73:     }else if ( (j + w-1) % 7 == 6){ //土曜日は青
74:         resultCalendar=resultCalendar+"<td align=right><font
color=blue>"+j+"</font></td>";
75:     }else if ( (j + w-1) % 7 == 0){ //日曜日は赤
76:         resultCalendar=resultCalendar+"<td align=right><font
color=red>"+j+"</font></td>";
77:     }else{
78:         resultCalendar=resultCalendar+"<td align=right>"+j+"</td>";
79:     }
80:     //日曜日で改行
81:     if ( (j + w-1) % 7 == 0) {
82:         resultCalendar=resultCalendar+"</tr><tr>";
83:     }
84: }
85: resultCalendar=resultCalendar+"</tr></table><BR>";
86: }
87: /*** 曜日を求める ***/
88: /* (仮引数) y:西暦年 m:月 d:日 (返却値) 曜日 0=日 1=月 2=火 3=水 4=木 5=金
6=土
*/
89: public int week_of_day(int y, int m, int d){
90:     int w;
91:
92:     if(m == 1) {
93:         y = y - 1;
94:         m = 13;
95:     }
96:     else if(m == 2) {
97:         y = y - 1;
98:         m = 14;
99:     }
100:    // ツェラーの公式より
101:    w = (5 * y / 4 - y / 100 + y / 400 + (26 * m + 16) / 10 + d) % 7;
102:    switch(w) {
103:        case 0:this.weekStr="日";break;
104:        case 1:this.weekStr="月";break;
105:        case 2:this.weekStr="火";break;
106:        case 3:this.weekStr="水";break;
107:        case 4:this.weekStr="木";break;
108:        case 5:this.weekStr="金";break;
109:        case 6:this.weekStr="土";break;
110:    }
111:    return w;
112: }
113:
114: /*** 月の最終日 ***/
115: /* (仮引数) y:西暦年 m:月 (返却値) 月の最終日 */
116: public int month_last_day(int y, int m){
117:     int d = 0, last[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
118:     if (m == 2) {
119:         // 閏年なら+1
120:         if(is_leap_year(y)) {
121:             d = last[m-1] + 1;

```

```

122:         }else {
123:             d = last[m-1];
124:         }
125:     }
126:     else if (m >= 1 && m <= 12) {
127:         d = last[m-1];
128:     }
129:     return d;
130: }
131:
132: /** 閏年の判定 */
133: /* (仮引数) y: 西暦年 (返却値) 1: 閏年 0: 閏年以外 */
134: public boolean is_leap_year(int y) {
135:     boolean ly = false;
136:     if (y % 4 == 0 && (y % 100 != 0 || y % 400 == 0)) {
137:         ly = true;
138:     }
139:     return ly;
140: }
141: /** 休日の設定 */
142: /* (仮引数) m: 月 d: 日 (返却値) True: 休日 False: 平日 */
143: public boolean is_holiday(int m, int d) {
144:     boolean hd=false;
145:
146:     int holiday[] = {101, 109, 211, 223, 321, 429, 503, 504, 505, 717, 811, 918,
147:                     923, 1009, 1103, 1123};
148:     for (int i=0; i<holiday.length; i++) {
149:         int mm=holiday[i]/100;
150:         int dd=holiday[i]%100;
151:         if (m==mm && d==dd) {
152:             hd=true;
153:         }
154:     }
155:     return hd;
156: }
157:
158: }

```