

## 13 多分岐構文 (case 文)

今まで、条件によって実行する文を選択する分岐構文は、二分岐構文の if 文だけを用いてきました。今回は場合分けが多いときに用いる多分岐構文の case 文を用いるプログラムを作ります。

### 13.1 Tanjoubi.dpr (誕生日の曜日)

#### 13.1.1 実行例

今日は 2005 年 11 月 21 日です  
誕生日として未来の日付を入れると終わります

あなたの誕生日 [YYYY MM DD] ? 1998 10 26  
平成 10 年 10 月 26 日は今日より 2583 日前で月曜日です

あなたの誕生日 [YYYY MM DD] ? 2005 12 31

#### 13.1.2 プログラム

```

1 program Tanjoubi;
2 {$APPTYPE CONSOLE}
3 uses
4   SysUtils;
5
6 function DaysFrom010101(Year,Month,Day : Integer) : Integer;
7   (* 西暦 1 年 1 月 1 日 (月) からの日数 *)
8   begin
9     if Month <= 2
10      then begin
11          Month := Month+12;
12          Year := Year-1;
13        end;
14    DaysFrom010101 := Year*365
15                      + Year div 4 - Year div 100 + Year div 400 // 閏年の調整
16                      + Trunc(Month*30.6-91.3) // 月の先頭
17                      + Day // 日
18                      - 306; // 01/01/01 を 1 に
19  end; {DaysFrom010101}
20
21 function IntToYoubi(Code : Integer) : String;
22   { 曜日のコード (0~6) を文字列に変換する }
23   begin
24     case Code of
25       0 : IntToYoubi := '日';
26       1 : IntToYoubi := '月';
27       2 : IntToYoubi := '火';
28       3 : IntToYoubi := '水';
29       4 : IntToYoubi := '木';
30       5 : IntToYoubi := '金';
31       6 : IntToYoubi := '土';
32     end;
33  end; {IntToYoubi}
34
35 procedure SeirekiToHourei(Seireki : Integer; var Gengou : String; var Nen : Integer);
36   { 西暦 邦暦 (明治, 大正, 昭和, 平成) }
```

```
37  begin
38      case Seireki of
39          1868..1911 : begin
40              Gengou := '明治';
41              Nen := Seireki-1867;
42          end;
43          1912..1925 : begin
44              Gengou := '大正';
45              Nen := Seireki-1911;
46          end;
47          1926..1988 : begin
48              Gengou := '昭和';
49              Nen := Seireki-1925;
50          end;
51          1989..2006 : begin
52              Gengou := '平成';
53              Nen := Seireki-1988;
54          end;
55          else      begin
56              Gengou := '西暦';
57              Nen := Seireki;
58          end;
59      end;
60  end; {HourekiToSeireki}
61
62  var
63      Y,M,D : Word;
64      Today,Birthday : Integer;
65      Gengou : String;
66      Houreki : INteger;
67  begin {Main}
68      DecodeDate(Now,Y,M,D);
69      WriteLn('今日は', Y, '年', M, '月', D, '日です');
70      Today := DaysFrom010101(Y,M,D);
71      WriteLn('誕生日として未来の日付を入れると終わります');
72      repeat
73          WriteLn;
74          Write('あなたの誕生日 [YYYY MM DD] ? ');
75          ReadLn(Y,M,D);
76          SeirekiToHoureki(Y,Gengou,Houreki);
77          Birthday := DaysFrom010101(Y,M,D);
78          Write(Gengou, Houreki, '年', M, '月', D, '日は');
79          Write('今日より', Today-Birthday, '日前で');
80          WriteLn(IntToYoubi(Birthday mod 7), '曜日です');
81      until Birthday > Today;
82  end.
```

// 今日の日付

### 13.1.3 解説

case 文は次の構造をしています。

```
case 式 of
  値 1 : 文 1 ;
  値 2 : 文 2 ;

  値 n : 文 n ;
  else 文 e ;
end
```

“式”を計算して、その値が“値 1”～“値 n”の中にあると、対応する“文 k”を実行します。その中に値がないときは“文 e”を実行します。

38～59 行のように、実行したい文が複数あるときは begin と end でくくって 1 つの複合文にします。

else の行は省略することができます。その場合、値のリストの中に式の値がないと何も実行しないで次に進みます。

13.1.4 改良

つぎの事項も出力するように改良しなさい。

- (1) 誕生年の<sup>えと</sup>干支も書く。

十干	十二支
甲(きのえ)	子(ね)
乙(きのと)	丑(うし)
丙(ひのえ)	寅(とら)
丁(ひのと)	卯(う)
戊(つちのえ)	辰(たつ)
己(つちのと)	巳(み)
庚(かのえ)	午(うま)
辛(かのと)	未(ひつじ)
壬(みずのえ)	申(さる)
癸(みずのと)	酉(とり)
	戌(いぬ)
	亥(い)

十干は西暦を 10 で割った余りで決まり, 十二支は 12 で割った余りで決まります。今年(2015年)は乙(きのと)酉(とり)です。

- (2) 誕生日の十二宮も書く。

3月21日 ~ 4月19日	<sup>おひつじ</sup> 牡羊座	Aries
4月20日 ~ 5月20日	<sup>おうし</sup> 牡牛座	Taurus
5月21日 ~ 6月21日	<sup>ふたご</sup> 双子座	Gemini
6月22日 ~ 7月22日	<sup>かに</sup> 蟹座	Cancer
7月23日 ~ 8月22日	<sup>しし</sup> 獅子座	Leo
8月23日 ~ 9月22日	<sup>おとめ</sup> 乙女座	Virgo
9月23日 ~ 10月23日	<sup>てんびん</sup> 天秤座	Libra
10月24日 ~ 11月22日	<sup>さそり</sup> 蠍座	Scorpius
11月23日 ~ 12月21日	<sup>いて</sup> 射手座	Sagittarius
12月22日 ~ 1月19日	<sup>やぎ</sup> 山羊座	Capricorn
1月20日 ~ 2月18日	<sup>みずがめ</sup> 水瓶座	Aquarius
2月19日 ~ 3月20日	<sup>うお</sup> 魚座	Pisces

M\*100 + D で場合分けすると簡単です。