FORM 4 PASCAL PROGRAMMING
Unit 4: Conditional Statements

March, 01

4.1 BOOLEAN EXPRESSION

Like the arithmetic expression, Boolean expression can also be evaluated to give result. But the possible values of the result can only be TRUE or FALSE.

4.1.1 Relational Operators 關係運算子

RELATIONAL OPERATOR	MEANING	BOOLEAN EXPRESSION
<=	less than or equal to	X <= Y 4 <= 4
<	less than	X < Y 1 < 8
>=	greater than or equal to	X >= Y 9 >= 5
>	greater than	X >= Y 8 >=7
=	equal to	X = Y 4 = 4
<>	not equal to	X <> Y 2 <> 5

- The result of a relational expression is either TRUE or FALSE.
- Variables and constants are used in a relational expression:

  e.g.
  ```
  X = 'GOODBYE'
  Y = 9
  A <> B
  ```
- The arithmetic (+, - , *, /) can be placed in a relational expression and it is always performed before the relational operators (<, >, =,etc.)

  e.g.
  ```
  2*A >= B        ⇔    (2*A) >= B
  B/2 < 5 + A     ⇔    (B/2) < (5 + A)
  ```

4.1.2 Logical Operators 邏輯運算子 (NOT, AND , OR)

NOT	
T	F
F	T

AND	T	F
T	T	F
F	F	F

OR	T	F
T	T	T
F	T	F

- e.g. Determine the result of the following logical expression, if A = 5, B = 0, C = 6, D = 6.

```
      NOT ((A >= B) OR NOT (C = D))

=     NOT ((5 >= 0) OR NOT (6 = 6))
=     NOT (T OR  NOT (T))
=     NOT (T OR F)
=     NOT (T)
=     F
```

4.2 COMPOUND STATEMENT 複合語句

- a sequence of ***one or more statements*** starting with BEGIN and ending with END
- NOTES: no semicolon(;) after reserved word BEGIN or before the reserved word END

e.g.

```
.
.
begin
  x := 1;
  y := 2;
  sum := x + y;
  writeln (sum)
end;
.
.
```

4.3 CONDITIONAL STATEMENTS 條件語句

- Conditional statements are used to alter the sequence of execution of statements within the program. It greatly increases the ability of the language to handle the decision-making problem.

- The conditional statements provided by PASCAL are as follows:

IF...THEN...
IF..THEN..ELSE..
CASE ...OF

4.3.1 `IF...THEN` *statement*

***Syntax*:**

Simple Statement

```
if boolean expression then
  statement;
```

Compound Statement

```
if boolean expression then
begin
  statement1;
  statement2;
        :
        :
  statementN
end;
```

Example:

```
if (a > max) then
begin
  max := a;
  index := index + 1
end;
```

- The statement(s) will be executed only when the boolean expression is evaluated to be TRUE.
- For the case that when the expression is evaluated to be FALSE, the statement(s) will not be executed at all.

4.3.2 `IF...THEN...ELSE` *statement*

***Syntax*:**

Simple Statement

```
if boolean expression then
  statement_a
else
  statement_b;
```

Compound Statement

```
if boolean expression then
  begin
    statement1;
        :
        :
    statementN
  end
else
  begin
    statement1;
    ststement2;
         :
    statementM
  end;
```

Example:

```
if (a >= b) then
  begin
    a := b - 10;
    c := a * b
  end
else
  begin
    a := b -20;
    c := a * 100
  end;
```

- The boolean expression is evaluated first;
- If it is evaluated to be TRUE, then the compound statement A, composed of 2 statements in the example, is executed and compound statement B will not be executed;
- If it is evaluated to be FALSE, then compound statement B, composed of 2 statements here, is executed and compound statement A will not be executed;
- REMEMBER: SEMI-COLON MUST NOT APPEAR AFTER END AND BEFORE ELSE!!!

4.3.3 CASE-statement

- To avoid using complicated nested-if statements, case-statement can be used to make the program to be clear and to reduce logical errors.

Syntax:

```
case selector of                              {no begin}
    a, b, c       : compound statement;
    d             : compound statement;
       :
       :
    n             : compound statement;
    else            compound statement;   {optional}
end;
```

Example:

```
case grade of
    'a'..'d' : begin    {represent 'a', 'b', 'c', 'd'}
                 writeln('pass');
                 writeln('keep it up')
               end;
    'e'      : writeln('just pass!');
    'f', 'u' : writeln('fail');
    else       writeln('error in grades')
end;
```

- The selector is evaluated first;
- the selector will be matched with various values (from 'a' to 'n') sequentially;
- If the selector equals to any listed value, then only the associated compound statement is executed. Other statements will not be executed;
- If none of the listed values are matched with the selector, then compound statement of ELSE will be executed. Without ELSE statement, nothing will be performed by CASE statement;

4.4 MORE EXAMPLES ON CONDITINAL STATEMENTS

```
program U3;
(* ================  program description ================ *)
(*  test if a number is even or odd                       *)
(* ========================================================= *)
var
   number, rem : integer;

begin
   writeln('please input any number for testing : ');
   readln(number);
   rem := number mod 2;
   if rem = 1 then
      writeln(number,' is odd')
   else
      writeln(number,' is even')
end.
```

```
program processmarks;
 mark : integer;
begin
  write( 'enter exam mark: ' );
  readln( mark );

  if mark >= 90 then
    writeln( 'a' )
  else if mark >=80 then
    writeln( 'b' )
  else if mark >= 70 then
    writeln( 'c' )
  else if mark >= 60 then
    writeln( 'd' )
  else if mark >= 50 then
    writeln( 'e' )
  else
    writeln( 'f' )
end.

program daysinmonth;
 month, noofdays : integer;
begin
  write( 'enter the month number: ' );
  readln( month );

  case month of
    1, 3, 5, 7,
      8, 10, 12 : noofdays := 31;
                2 : noofdays := 28;
    4, 6, 9, 11 : noofdays := 30
  end;

  writeln( 'there are ', noofdays, ' days in this month.' )
end.


program outputday;
(* ================  program description ================ *)
(*  according to input number and output suitable string. *)
(* ===================================================== *)
var
   index : integer;

begin
   writeln('please input the number( 1 - 7 ) ');
   readln(index);
   case index of
      1: writeln('today is sunday.');
      2: writeln('today is monday.');
      3: writeln('today is tuesday.');
      4: writeln('today is wednesday.');
      5: writeln('today is thursday.');
      6: writeln('today is friday.');
      7: writeln('today is saturday.')
      else
         writeln('input error!')
   end
end.
```

e n d o f u n i t 4