

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

แผนปฏิบัติการสอนสัปดาห์ที่ 8

วิชา 030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม ระดับ: ปริญญาตรี

เรื่อง การควบคุมโปรแกรมด้วยคำสั่งต่าง ๆ

เวลา: บรรยาย 120 นาที

ปฏิบัติ 120 นาที

ก. วัตถุประสงค์การสอน	รายละเอียดตามที่จะพบไว้ใน
1. if else, case 2. for do 3. while 4. repeat until 5. break, continue, exit และ abort	หน้า 105-112

ข. การจัดการเรียนการสอน

เวลา - นาที	0	60	120	180	240
วัตถุประสงค์	1, 2, 3, 4, 5				
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
พัก					
ทำแบบฝึกหัด					
ให้เนื้อหา					
สรุปเนื้อหา					
ทดสอบและเก็บคะแนน					
ประเมินผล	พิจารณาจากผลการทดสอบในช่วงสุดท้าย				
วิธีการสอน:	บรรยาย				
	ถาม - ตอบ				
	ทำแบบฝึกหัด				
	บรรยาย และแสดงให้ดู				
	ทดสอบ				
สื่อการสอน:	คอมพิวเตอร์				
	Presentations				

Tip

- เนื้อหาในบทนี้เป็นการบรรยายในภาคทฤษฎี ให้ค้น.จด Note ลงในเอกสารประกอบคำบรรยาย และเขียน Code ทดสอบเพื่อให้เข้าใจในเนื้อหาดียิ่งขึ้น

Procedure and Keyword

Procedure	Abort	Aborts the current processing with a silent exception
Keyword	Begin	Keyword that starts a statement block
Function	BeginThread	Begins a separate thread of code execution
Procedure	Break	Forces a jump out of a single loop
Keyword	Case	A mechanism for acting upon different values of an Ordinal
Procedure	Continue	Forces a jump to the next iteration of a loop
Keyword	Do	Defines the start of some controlled action
Keyword	DownTo	Prefixes an decremental for loop target value
Keyword	Else	Starts false section of if, case and try statements
Keyword	End	Keyword that terminates statement blocks
Procedure	EndThread	Terminates a thread with an exit code
Variable	ErrorAddr	Sets the error address when an application terminates
Keyword	Except	Starts the error trapping clause of a Try statement
Procedure	Exit	Exit abruptly from a function or procedure
Keyword	Finally	Starts the unconditional code section of a Try statement
Keyword	For	Starts a loop that executes a finite number of times
Keyword	Function	Defines a subroutine that returns a value
Keyword	Goto	Forces a jump to a label, regardless of nesting
Procedure	Halt	Terminates the program with an optional dialog
Keyword	If	Starts a conditional expression to determine what to do next
Keyword	Implementation	Starts the implementation (code) section of a Unit
Keyword	Interface	Used for Unit external definitions, and as a Class skeleton
Function	IsMultiThread	Returns true if the code is running multiple threads
Keyword	Of	Linking keyword used in many places
Keyword	On	Defines exception handling in a Try Except clause
Keyword	Procedure	Defines a subroutine that does not return a value
Keyword	Program	Defines the start of an application
Keyword	Raise	Raise an exception
Keyword	Repeat	Repeat statements until a termination condition is met
Procedure	RunError	Terminates the program with an error dialog
Keyword	Then	Part of an if statement - starts the true clause
Keyword	To	Prefixes an incremental for loop target value

Keyword	Try	Starts code that has error trapping
Type	TThreadFunc	Defines the function to be called by BeginThread
Keyword	Unit	Defines the start of a unit file - a Delphi module
Keyword	Until	Ends a Repeat control loop
Keyword	Uses	Declares a list of Units to be imported
Keyword	While	Repeat statements whilst a continuation condition is met

<http://www.delphibasics.co.uk/ByFunction.asp?Main=PgmStruct&Sub=Control>

1. if else, case

if else

```
if Condition then Statement else Statement;
case Expression of
    Case clauses
    ...
else
    Statements
end;
try
```

```
Statements
...
except
    Exception statements
...
else
    Statements
end
```

Example

```
begin
    // Note the lack of a ';' after the 'then' clause
    if False
    then ShowMessage('True')
    Else ShowMessage('False');

    // Nested if statements - Delphi sensibly manages
    associations
```

```
if true then
    if false then
        ShowMessage('Inner then satisfied')
    Else
        ShowMessage('Inner else satisfied')
    Else
        ShowMessage('Outer else satisfied')
end;
```

Results

```
False
    Inner else satisfied
```

<http://www.delphibasics.co.uk/RTL.asp?Name=IF>

Case

```
1 case Ordinal expression of
    Ordinal value {Ordinal value...} : Statement;
{...}
```

```
else Statement;
end;
2 type Name = record
```

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

Declarations ...

case {Tag :} Ordinal type of

Ordinal value {Ordinal value...} : (Declarations);

{...}

end;

// Procedure to show the colour of a passed

procedure TForm1.ShowColour(colour : TPrimary);

begin

*// Use a case statement to see the colour of the
passed var*

*// Note how important the else clause is, even though
we have*

// apparently covered all TPrimary values!

Case colour of

Red : ShowMessage('The colour is Red');

Green : ShowMessage('The colour is Green');

Blue : ShowMessage('The colour is Blue');

Yellow : ShowMessage('The colour is Yellow');

else ShowMessage('The colour is Unknown!');

end;

end;

Results

The colour is Unknown!

The colour is Green

17 mod 2 is 1

<http://www.delphibasics.co.uk/RTL.asp?Name=Case>

2. for do

1 for Variable := Integer Expression to/downto Integer Expression do Statement;

2 for Variable := Char Expression to/downto Char Expression do Statement;

3 for Variable := Enum Expression to/downto Enum Expression do Statement;

Example

var

i : Integer;

begin

// Loop 5 times

For i := 1 to (10 div 2) do

ShowMessage('i = '+IntToStr(i));

end;

Results

i = 1

i = 2

i = 3

i = 4

i = 5

<http://www.delphibasics.co.uk/RTL.asp?Name=For>

3. while

While Expression do Statement;

Example

```
var
  num, sqrNum : Integer;

begin
  num := 1;
  sqrNum := num * num;
  // Display squares of integers until we reach 100 in
value
  While sqrNum <= 100 do
  begin
```

```
    // Show the square of num
    ShowMessage(IntToStr(num)+' squared =
'+IntToStr(sqrNum));
    // Increment the number
    Inc(num);
    // Square the number
    sqrNum := num * num;
  end;
end;
```

Results

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
```

```
6 squared = 36
7 squared = 49
8 squared = 64
9 squared = 81
10 squared = 100
```

<http://www.delphibasics.co.uk/RTL.asp?Name=While>

4. repeat until

Repeat

```
Statement1;
{Statement2;
...}
Until Expression
```

Example

```
var
  num, sqrNum : Integer;

begin
  num := 1;
```

```
  sqrNum := num * num;

  // Display squares of integers until we reach 100 in
value
  Repeat
```

```
// Show the square of num
ShowMessage(IntToStr(num)+' squared =
'+IntToStr(sqrNum));

// Increment the number
Inc(num);
```

```
// Square the number
sqrNum := num * num;

until sqrNum > 100;

end;
```

Results

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
```

```
6 squared = 36
7 squared = 49
8 squared = 64
9 squared = 81
10 squared = 100
```

<http://www.delphibasics.co.uk/RTL.asp?Name=Repeat>

5. break, continue, exit และ abort

Break

```
var
  i : Integer;
  s : string;

begin
  s := '';
  // A big loop
  for i := 1 to 10 do
```

```
begin
  s := s + IntToStr(i) + ' ';
  // Exit loop when a certain condition is met
  if Random(4) = 2 then Break;
end;

ShowMessage('i = ' + IntToStr(i));

ShowMessage('s = ' + s);

end;
```

Results

```
i = 6
s = 1 2 3 4 5 6
```

<http://www.delphibasics.co.uk/RTL.asp?Name=Break>

Continue

```
var
  i : Integer;
  s : string;

begin
  s := '';
  // A big loop
```

```
for i := 1 to 9 do
begin
  // Skip loop processing for certain values of i
  if (i = 3) or (i = 7) then Continue;

  s := s + IntToStr(i);
  s := s + ' ';
```

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

```
end;  
  
// Show the string created by the above loop
```

```
ShowMessage('s = '+s);  
end;
```

Results

s = 1 2 4 5 6 8 9

<http://www.delphibasics.co.uk/RTL.asp?Name=Continue>

Exit

```
begin  
    // Ask the user for their name  
    ShowMessage('Name = '+AskForName);  
end;  
  
// Ask the user for first and second names  
function TForm1.AskForName: string;  
var  
    firstName, secondName : string;  
begin  
    Result := 'Lazy person';
```

```
repeat  
    if not InputQuery('Test program', 'First name :',  
        firstName)  
    then Exit;  
    if not InputQuery('Test program', 'Second name :',  
        secondName)  
    then Exit;  
    until (firstName <> '') or (secondName <> '');  
    Result := firstName + ' ' + secondName;  
end;
```

Results

If the user cancels from the first or second name dialogs, the ShowMessage dialog gives:

Name = Lazy person

<http://www.delphibasics.co.uk/RTL.asp?Name=Exit>

Abort

```
begin  
    // Enter a try block  
  
    Try  
        ShowMessage('Before abort');  
        Abort;  
        ShowMessage('After abort');
```

```
Except  
    On E : Exception do ShowMessage(E.Message + '  
exception occurred');  
end;  
  
ShowMessage('After try');  
end;
```

Results

Before abort

Operation aborted exception occurred

After try

<http://www.delphibasics.co.uk/RTL.asp?Name=Abort>

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

Example

[Code\week7\FX_ControlStatement_1](#)

Exercise

- อธิบายข้อแตกต่างระหว่างคำสั่ง *for*, *while*, *repeat*
- เขียนโปรแกรมตัดเกรดนักศึกษาที่มีคะแนนดิบอยู่ใน Array ขนาด 1X10 ให้เกรดตามช่วงคะแนนดังนี้
A(80-100), B(60-79), C(50-59), F(0-49)

Assignment

- เขียนโปรแกรมรับข้อมูลจาก User จำนวน 10 รอบจากการ Click ที่ปุ่ม ในการรับจะห้ามการรับค่าในรอบที่ 3 ทำให้เหลือการรับจริง 9 รอบ และถ้า User ป้อน "E" ให้ออกจากการรับข้อมูล ถ้า CheckBox1 มีการ Checked ไว้ ให้ออกจาก Procedure OnClick ทันทีด้วยคำสั่ง Exit;

Answer Sheet

[Code\week7\Ans_Grade_1](#)

[Code\week7\Ans_Break_1](#)