

# 第7章 控制结构

## 知识要点

- 各种控制结构的含义和使用的语句
- 过程的概念、分类、定义及调用方法

## 学习任务

- 了解各种控制结构的含义，过程的概念和分类
- 理解各种控制结构所使用语句的功能，函数过程和子程序过程的定义、调用方式
- 掌握各种控制结构、函数过程和子程序过程的使用方法



# 7.1 条件判断结构

## 7.1.1 If...Then...Else语句

### 1. 单行式

语法:

```
If 条件表达式 Then 语句1 [Else 语句2]
```

功能:

当条件表达式成立时，执行关键字Then后面的语句1，否则执行关键字Else后面的语句2。

### 2. 区块式

语法:

```
If 条件表达式 Then  
    语句序列1  
[Else  
    语句序列2]  
End If
```



## 功能:

当条件表达式成立时，执行关键字Then后面的语句序列1，否则，执行关键字Else后面的语句序列2。无论执行的是语句序列1，还是语句序列2，执行完以后都要执行End If 后面的语句。

## 7.1.2 If...Then...ElseIf语句

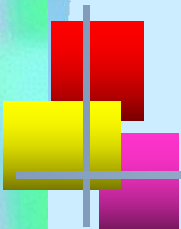
### 语法:

```
If 条件表达式1 Then
    语句序列1
[ElseIf 条件表达式2 Then
    语句序列2]
    .
    .
    .
[Else
    语句序列n+1]
End If
```



## 功能:

首先测试条件表达式1，如果其值为True，则执行语句序列1，然后跳过关键字ElseIf至End If之间的语句，而执行关键字End If后面的语句；反之，如果条件表达式1的值为False，则测试条件表达式2，依次类推，直到找到一个值为True的条件表达式，并执行其后面的语句序列，然后接着执行End If后面的语句；如果条件表达式的值都不是True，则执行关键字Else后面的语句序列n+1，然后接着执行End If后面的语句。



### 7.1.3 Select Case语句

#### 语法:

```
Select Case表达式  
    [Case 取值1  
        语句序列1]  
    [Case 取值2  
        语句序列2]  
    .  
    .  
    .  
    [Case Else  
        语句序列n+1]  
End Select
```

#### 功能:

先计算表达式，然后，将表达式的值依次与语法中的每个Case关键字后面的取值进行比较，如果相等，就执行该Case后面的语句序列；如果都不相等，则执行Case Else子语句后面的语句序列。无论执行的是哪一个语句序列，执行完后都要接着执行关键字End Select后面的语句。



## 7.2 循环程序结构

### 7.2.1 For...Next语句

语法:

```
For 计数变量 = 初值 To 终值 [Step 增量值]
    语句序列
[Exit For]      循环体
    语句序列
Next [计数变量]
```

功能:

重复执行For语句和Next语句之间的语句序列。

### 7.2.2 For Each...Next语句

语法:

```
For Each 变量 In 数组
    语句序列
[Exit For]      循环体
    语句序列
Next [变量]
```



## 功能:

变量每取数组中的一个元素, 都重复执行关键字For Each和Next之间的语句序列。

## 7.2.3 Do...Loop语句

### 1. 当型Do...Loop语句

#### 语法:

```
Do [While 条件表达式]
    [语句序列]
[Exit Do]      循环体
[语句序列]
Loop [While 条件表达式]
```

#### 功能:

当条件表达式成立时, 重复执行关键字Do和关键字Loop之间的语句序列, 当条件表达式不成立时, 则结束循环, 转去执行关键字Loop后面的语句。



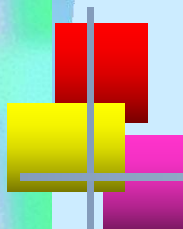
## 2. 直到型Do...Loop语句

### 语法:

```
Do [Until 条件表达式]
    [语句序列]
[Exit Do]      循环体
[语句序列]
Loop [Until 条件表达式]
```

### 功能:

重复执行关键字Do和关键字Loop之间的语句序列，直到条件不成立时，则结束循环，转去执行关键字Loop后面的语句。





## 7.2.4 While...Wend语句

### 语法:

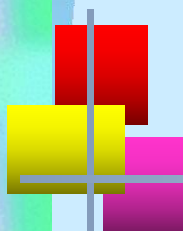
```
While 条件表达式  
    语句序列  
Wend
```

### 功能:

当条件成立时，重复执行语句序列，否则，转去执行关键字Wend后面的语句。

## 7.3 控制结构的嵌套

可以把一个控制结构放入另一个控制结构之中，这称为控制结构的嵌套。



## 7.4 过程

### 7.4.1 Function过程

#### 1. Function过程的定义

语法:

```
[Public|Private][Static]Function过程名[类型说明符]  
[(形参表)][As 类型说明词]  
    语句序列  
    [Exit Function]  
    语句序列  
    函数名=表达式  
End Function
```

功能:

声明Function过程的名称、形式参数，以及构成该过程的语句序列。



## 2. Function过程的创建

创建Function过程有两种方法。

(1) 执行“工具”菜单中的“添加过程”命令。

(2) 在“代码”窗口中直接输入Function语句并回车，系统就会自动

为其加上End Function语句，然后在两条语句之间输入过程所需语句即可。

## 3. Function过程的调用

Function过程一旦被声明，就可以像标准函数那样在程序中调用了，即在表达式中写上该过程的名称及相应的实在参数即可。



## 7.4.2 Sub过程

### 1. Sub过程的定义

语法:

```
[Public|Private][Static]Sub 过程名[形式参数表]  
    语句序列  
    [Exit Sub]    过程体  
    语句序列  
End Sub
```

功能:

声明Sub过程的名称、形式参数, 以及构成该过程的语句序列。

### 2. Sub过程的调用

#### (1) 使用关键字Call

语法:

```
Call 过程名[实在参数表]
```



(2) 省略关键字Call。

语法：

过程名[实在参数表]

### 7.4.3 Static选项

### 7.4.4 参数传递

1. 形式参数与实在参数

2. 参数传递方式

(1) 按值传递

(2) 按地址传递

(3) 命名传递

参数格式为：

〈形式参数〉:〈实在参数〉

### 7.4.5 Property过程

### 7.4.6 Event过程



# 思考与练习

## 填空题

- ✦ Visual Basic程序，按其语句代码执行的先后顺序，可以分为\_\_\_\_\_结构、\_\_\_\_\_结构和\_\_\_\_\_结构。
- ✦ 条件判断结构可以使用\_\_\_\_\_语句、\_\_\_\_\_语句和\_\_\_\_\_语句。
- ✦ If...Then...Else语句是\_\_\_\_\_语句的特例。
- ✦ 在Select Case语句中，当不止一个Case后面的取值与表达式的值相匹配时，执行\_\_\_\_\_ Case后面的语句序列？
- ✦ 实现循环程序结构，可以使用\_\_\_\_\_语句、\_\_\_\_\_语句、\_\_\_\_\_语句和\_\_\_\_\_语句。
- ✦ 在Visual Basic语言中，过程可以分为\_\_\_\_\_过程、\_\_\_\_\_过程、\_\_\_\_\_过程和\_\_\_\_\_过程。
- ✦ 参数传递有\_\_\_\_\_方式、\_\_\_\_\_方式和\_\_\_\_\_方式，常量默认采用\_\_\_\_\_方式；变量默认采用\_\_\_\_\_方式；表达式默认采用\_\_\_\_\_方式。



# 思考与练习

## 简答题

- ✦ 在Select Case语句中，关键字Case后面的取值的格式有哪几种？试举例说明。
- ✦ 简述For...Next语句的执行过程。
- ✦ 在Do...Loop语句中，根据条件表达式前面使用的关键字的不同，可将Do...Loop语句分为哪两种形式？在每一种形式中，根据表达式所在位置的不同，又可将Do...Loop语句分为哪两种形式？这两种形式的Do...Loop语句在执行时有什么区别？
- ✦ 结构化程序设计具有哪些优点？
- ✦ Sub过程和Function过程有哪几种创建方法？

