全国高等职业教育示范专业规划教材

计算机专业

VB数据库项目设计模块化教程

VB SHUJUKU XIANGMU SHEJI MOKUAIHUA JIAOCHENG

刘玉山　刘宝山　主编

机械工业出版社
CHINA MACHINE PRESS

赠电子课件

全国高等职业教育示范专业规划教材

计算机专业

VB 数据库项目设计模块化教程

主　编　刘玉山　刘宝山

副主编　王　蓉

参　编　戴　瑞　刘　涌

刘彩霞　冯文田

机械工业出版社

以工作过程为导向开发职业教育课程，在教学中模拟真实的工作情境，是职业教育行之有效的途径。基于这一理念，本书力求将计算机专业传统的“VB 程序设计”和“数据库应用”两门课程进行整合，融入“软件工程”的思想，采用行业设计规范和流程，以软件开发工作过程为导向，确立软件设计任务，提炼可供课堂教学的设计任务，突出可视化编程模块化特点。

本书共分 13 章，每章以具体项目为导引，通过“项目需求分析”对项目进行描述，在“项目设计”过程中实现程序设计，在“技术要点小结”中提炼技术要点，通过【模拟项目演练】提供学生实训的机会。

本书可作为高职高专计算机专业的教材，也可作为其他层次的学生和程序设计爱好者，特别是数据库信息管理系统开发设计人员的参考用书。

本书配有电子教案，凡使用本书作为教材的教师可登录机械工业出版社教材服务网 www.cmpedu.com 下载。如有问题请致信 cmpgaozhi@sina.com,或致电 010-88379375 联系营销人员。

图书在版编目（CIP）数据

VB 数据库项目设计模块化教程/刘玉山，刘宝山主编. —北京：机械工业出版社，2009.8

全国高等职业教育示范专业规划教材. 计算机专业

ISBN 978-7-111-27333-2

Ⅰ.V... Ⅱ.①刘... ②刘... Ⅲ.①BASIC 语言—程序设计—高等学校：技术学校—教材 ②关系数据库—数据库管理系统—程序设计—高等学校：技术学校—教材 Ⅳ.TP312 TP311.138

中国版本图书馆 CIP 数据核字（2009）第 127184 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：李大国 责任印制：洪汉军

三河市国英印务有限公司印刷

2009 年 8 月第 1 版第 1 次印刷

184mm × 260mm · 8.5 印张 · 209 千字

0001—4000 册

标准书号：ISBN 978-7-111-27333-2

定价：16.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

销售服务热线电话：（010）68326294

购书热线电话：（010）88379639 88379641 88379643

编辑热线电话：（010）68354423

前　言

Visual Basic 以其强大的功能，简单、直观、易学的特点，一直是众多程序设计初学者首选的软件开发平台，也是计算机爱好者走进面向对象程序设计殿堂的捷径。数据库系统作为信息管理系统开发不可或缺的部分，已成为广大程序设计人员必须掌握的内容之一。因此，这两门课程在计算机专业中具有十分重要的地位。然而，单独学习这两门课程，不仅需要过多地投入学习成本，而且很难使学生深刻认识它们之间的有机联系，使学生在程序开发过程中“编程找不到对象，数据库不知如何操作”。为克服传统教学带来的弊端，我们对“VB 程序设计”和“数据库应用”这两门课程进行整合并编写了本书。

职业教育的课程开发需要以工作过程为导向，强调在教学中模拟真实的工作情境，采用行业规范和流程，采取任务驱动教学方式，实现模块化项目教学。因此，在教材编写过程中，我们打破了传统的教学模式，强化了工作过程为导向的课程开发理念，采用了 MIS 系统开发的一般流程，并融入了软件工程的思想。

全书共分 13 章，以学生比较熟悉的“学生成绩管理系统”作为案例，并将其有机地划分成“三大模块”、“25 个子模块”，从而构成学生学习和实训的设计项目。每章以具体项目为导引，通过“项目需求分析”对项目进行描述，使学生明确具体任务；在“项目设计”过程中学习和掌握程序设计和数据库设计方法和技能，实现程序设计；在“技术要点小结”中提炼技术要点，方便学生对知识的归纳和总结；通过“模拟项目演练”为学生提供实训的机会，促进学生设计能力的提高。

通过本书的学习，不但可以使学生掌握“VB 程序设计”和“数据库应用”两门课程所包含的知识内容，而且可以使学生体会到软件工程和 MIS 系统开发的一般方法，初步建立系统开发的思想，奠定学生进行毕业设计的基础和培养学生软件开发的实际能力。

为方便教学，本书配套有多媒体课件及每个任务设计的源代码。

本书是作者多年从事计算机教学工作的结晶，不仅包含着作者个人的努力，而且蕴含着作者同事的支持和家人的一份辛劳。

尽管本书在正式出版之前作为校本教材有过成功的应用经历，但面对广大的读者难免会顾此失彼，存在不妥之处，恳请广大读者批评指正。

编　者

目　录

前言

第1章　Visual Basic、Access 及 SQL Server 导引 1

1.1　Visual Basic 概述 1
1.2　数据库概述 15
1.3　Access 概述 17
1.4　SQL Server 概述 19
1.5　SQL 数据查询语言概述 29
【模拟项目演练】 33

第2章　学生成绩管理系统总体设计 35

2.1　学生成绩管理系统的定义 35
2.2　学生成绩管理系统功能描述 37
2.3　学生成绩管理系统的数据结构 38

第3章　登录对话框模块设计 40

3.1　项目需求分析 40
3.2　项目设计 41
3.3　技术要点小结 45
【模拟项目演练】 48

第4章　管理员授权用户模块设计 51

4.1　项目需求分析 51
4.2　项目设计 52
4.3　技术要点小结 54
【模拟项目演练】 55

第5章　教师基本情况添加模块设计 56

5.1　项目需求分析 56
5.2　项目设计 57
5.3　技术要点小结 60
【模拟项目演练】 61

第6章　教师基本情况查询和浏览模块设计 62

6.1　项目需求分析 62
6.2　项目设计 65
6.3　技术要点小结 70
【模拟项目演练】 71

第7章　教师基本情况修改模块设计 72

7.1　项目需求分析 72
7.2　项目设计 74
7.3　技术要点小结 77
【模拟项目演练】 78

第8章　教师基本情况删除模块设计 80

8.1　项目需求分析 80
8.2　项目设计 82
8.3　技术要点小结 84
【模拟项目演练】 84

第9章　学生基本情况管理模块设计 85

9.1　项目需求分析 85
9.2　项目设计 89
9.3　技术要点小结 95
【模拟项目演练】 96

第10章　教师任务书查询功能模块设计 99

10.1　项目需求分析 99
10.2　项目设计 100
10.3　技术要点小结 107
【模拟项目演练】 107

第11章　学生成绩添加模块设计 114

11.1　项目需求分析 114
11.2　项目设计 116
11.3　技术要点小结 118
【模拟项目演练】 119

第12章　备份和恢复模块设计 121

12.1　项目需求分析 121
12.2　项目设计 122
12.3　技术要点小结 123

第13章　学生成绩管理系统集成和打包 126

13.1　学生成绩管理系统的集成 126
13.2　学生成绩管理系统的编译和打包 128
13.3　技术要点小结 129

第1章 Visual Basic、Access 及 SQL Server 导引

1.1 Visual Basic 概述

Visual Basic 是一种可视化的、面向对象和事件驱动方式的结构化高级程序设计语言，可用于开发 Windows 环境下的各类应用程序。Visual Basic 通过图形对象（包括窗体、控件、菜单等）来设计应用程序。图形对象的建立和使用都十分简单，只需要为数不多的几行程序就可以控制这些图形对象。

Visual Basic 是采用事件驱动编程机制的计算机语言之一。事件驱动是一种适用于图形用户界面（GUI）的编程方式。传统的编程是面向过程、按规定顺序进行的，程序设计人员总是要关心什么时候发生什么事情。对于现代的计算机应用来说，必须根据用户的需求安排程序的执行，而这实际上就是事件驱动程序所要解决的问题。

用事件驱动方式设计程序时，程序员不必给出按精确次序执行的每个步骤，只是编写响应用户动作的程序。例如选择命令，移动鼠标，用鼠标单击某个图标等。与传统的面向过程的语言不同，在用 Visual Basic 设计应用程序时，要编写的不是大段的程序代码，而是由若干个微小程序，这些微小程序由用户启动的事件来激发，从而大大降低了编程的难度和工作量，提高了程序的开发效率。

Visual Basic 的主要特点有：

（1）可视化编程。

（2）事件驱动的编程机制。

（3）面向对象的设计方法。

（4）结构化的程序设计语言。

（5）强大的数据库管理功能。

（6）友好的帮助系统。

1.1.1 VB 运行环境及基本概念

Visual Basic 的集成开发环境（IDE）是开发 Visual Basic 应用程序的开发设计平台，熟练掌握 Visual Basic 集成开发环境是开发应用程序的基础。Visual Basic 的集成开发环境及其几个主要组成部分，如图 1-1 所示。

VB 程序设计是围绕窗体设计进行的，因此窗体设计器是 VB 程序设计的中心。工具箱是可以使用的控件集合，通过工具箱可以将控件添加到窗体上，从而实现对程序运行的控制。属性窗口是对控件属性进行静态设置和控制的工具，如图 1-2 所示，通过属性窗口可以设置控件的初始状态。代码窗口是编程窗口，如图 1-3 所示，通过代码窗口可以对控件的属性进行动态控制，改变控件的状态，即进行控件事件编程，也可以在代码窗口编写普通过程和函数的编程。

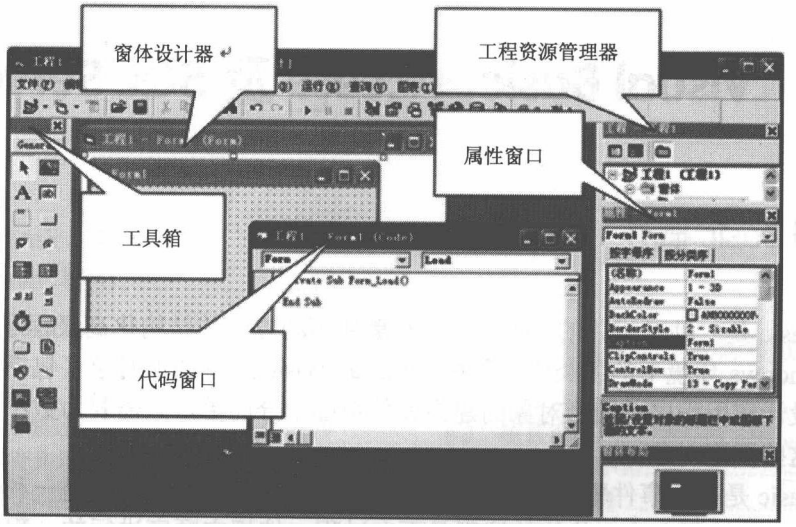

图 1-1 Visual Basic 的集成开发环境

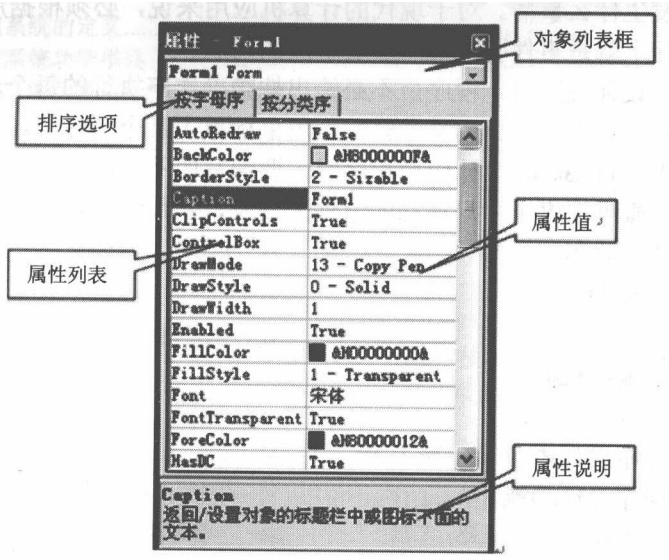

图 1-2 属性窗口

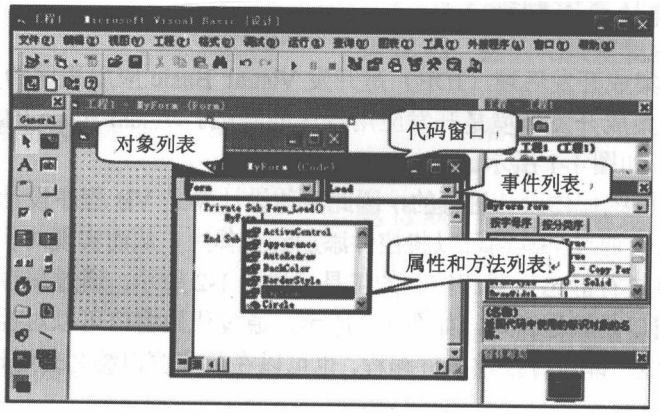

图 1-3 代码窗口

1.1.2　VB 语言基础

在 VB 程序设计中，基本的设计机制就是：为对象事件编写事件过程，在事件响应中填写改变对象属性的语句，通过调用对象的方法完成特定功能。

1．创建 VB 应用程序的主要步骤

（1）**新建工程**　用户界面通常是由对象，即窗体和控件组成。所有的控件都放在窗体上，每个窗体最多可容纳 255 个控件。程序中的所有信息都要通过窗体显示出来。在窗体上建立所需的控件，程序运行后，将在屏幕上显示由窗体和控件组成的用户界面。

要建立新的窗件，可用“工程”→“添加窗体”命令。

（2）**设计应用程序界面**　在窗体上按应用程序要求将控件调整到适当大小，放置到相应的位置上。

（3）**设置窗体和控件的属性**　设置窗体和控件的属性是通过“属性窗口”进行的。

为了使界面设计清晰而有条理，通常在设计前将界面中所需要的对象及其属性画成一个表格，然后按照这个表来设计界面。

另外，窗体的大小及每个控件的位置、大小属性均可根据需要任意调整，标题及输出字体的属性也可任意改变。

（4）**编写事件驱动代码**　VB 应用程序采用事件驱动编程机制，因此大部分程序都是针对窗体中各个控件所能支持的方法或事件编写的，这样的程序称为事件处理过程。

每个事件对应一个事件处理过程。

为了指明某个对象的操作，必须在方法或属性前加上对象名，中间用句点（.）隔开，如 Text1.Text。如果不指出对象名，则针对当前窗体进行操作。

（5）**程序的运行和调试**

1）解释运行。解释运行需要系统有 Visual Basic 环境，应用程序不能独立运行。当程序装入内存后，可通过“运行”→“启动”命令来实现。如果想退出程序，可单击“结束程序”按钮。

2）生成可执行文件。要使程序能在 Windows 环境下独立运行，必须建立可执行文件，即.EXE 文件。执行“文件”→“生成.EXE”命令，输入文件名后确定。

（6）**文件的保存**　在生成的 VB 程序中，常包含以下文件：

工程文件（.vbp）：包含了一个应用程序的所有文件，由若干窗体和模块组成。

窗体文件（.frm）：包含控件及属性、事件过程和自定义过程。

标准模块文件（.bas）：包含不与具体的窗体或控件相关联的代码，如全局变量声明，自定义函数或子程序过程。

类模块文件（.cls）：可以看成是没有界面的控件，每个类模块定义了一个类，可以在窗体模块中定义类的对象，调用类模块中的过程。

保存文件的步骤：

1）保存窗体文件。

2）保存标准模块文件。

3）保存类模块文件。

4）保存工程文件。

也可执行“文件”→“工程另存为”命令，直接保存工程文件。此时会自动将与该工程

相关的各类文件一起保存。

需要装入程序时，可执行“文件”→“打开工程”命令。只需装入工程文件，就可以自动把与该工程有关的其他文件装入内存。

（7）**应用程序运行的操作序列**

1）启动应用程序，加载和显示窗体。

2）窗体或窗体上的控件接收事件，事件可由用户引发，也可由系统引发，还可以通过代码间接引发。

3）如果相应的事件过程中存在代码，则执行该代码。

4）应用程序等待下一次事件。

2．Visual Basic 的数据类型

Visual Basic 6.0 提供的基本数据类型主要有字符串型数据和数值型数据，此外还提供了字节型、货币型、对象型、日期型、布尔型和变体型数据类型。

（1）**字符型**（String）**数据** 字符型是一个字符序列，由 ASCII 字符组成，包括标准的 ASCII 字符和扩展的 ASCII 字符。在 VB 中，字符型是放在双引号内的若干个字符，其中长度为 0（即不含任可字符）的字符串称为空字符串。如：

"Visual Basic 程序设计"　"控件"　"123456"　"Lbs@bttc.cn"　" "

（2）**数值型数据** Visual Basic 的数值型数据分为整数和浮点数两类。其中，整数又分为整型(Integer)和长整型(Long)；浮点数分为单精度浮点数(Single)和双精度浮点数(Double)。如：

1234　54321　123.45　1.2345e2　1.2e-127

（3）**货币型**（Currency）**数据** 货币型数据小数点前最多有 15 位数，小数点后只保留 4 位数，超过 4 位的小数，系统按四舍五入自动截取。如：

1234704345　13258.3962

（4）**日期型**（Date）**数据** 日期型数据表示法有两种：一种是以数字符号“#”括起来的格式化表示法，如#January 1,1993#或#1 Jan 93#；另一种是以数字序列表示，小数点左边是日期，右边是时间，如 2.5 表示 1900-1-1 12:00:00。有关数字序列与日期型数据的对应关系，可通过下面代码给出：

```
Private Sub Form_Click ()
    Dim D As Date
    D=2.5
    Print D
End Sub
```

（5）**布尔型**（Boolean）**数据** 布尔型数据是表示真假的数据，用于表示逻辑判断的结果。取值只有真（True）和假（False）两个值。

（6）**变体型**（Variant）**数据** 变体数据类型是一种可变的数据类型，可以表示任何值，包括数值、字符串、日期/时间等。

注意：若未设置强制声明，则 VB 允许使用没有声明的变量，其会自动创建该变量并赋予这个变量 Variant 类型，其初值为 empty。

更多关于数据类型的内容请参阅 MSDN 技术资源库的“数据类型”主题。

3．常量和变量

（1）**常量**　常量是程序运行中不可改变的量。Visual Basic 系统中常量分为直接常量、用户声明的符号常量和系统预定义常量。

1）直接常量。直接常量也称为常数。不同类型的直接常量表现形式不同，如 123、-78.9、"程序设计"、#04/12/2008#、True 分别是整型、浮点型、字符串型、日期型和布尔型常量。

2）符号常量。符号常量是命名的数据项，其值和类型由定义时确定，作用是增加程序代码的可读性，提高程序调试的效率。

一般格式为：

Const 常量名＝表达式[,常量名＝表达式]...

3）系统常量。除了用户自定义的符号常量外，VB 系统提供了系统常量，用户可以直接引用。如系统的颜色常量 vbBlack、vbRed 和 vbGreen。

更多的系统常量及应用请参阅 MSDN 的“常量”主题。

（2）**变量**　Visual Basic 用变量来储存数据值。每个变量都有一个名字和相应的数据类型，通过名字来引用一个变量，而数据类型则决定了该变量的储存方式。变量是程序中数据的临时存放场所，可以保存程序运行时用户输入的数据、特定运算的结果以及要在窗体上显示的一段数据等。变量的值在程序运行中是可以变化的。

1）变量的声明。变量的声明就是定义变量名和变量的数据类型。Visual Basic 系统声明变量分为显示声明和隐式声明两种。

① 显式声明。显示声明变量的格式：

Dim|Static 变量名[As 类型][,变量名[As 类型]]

如：

```
Dim x As Integer '定义 x 为整型变量
Dim str As String '定义 str 为变长字符串变量
Dim a As Integer,b As Double '定义 a 为整型变量，b 为双精度浮点型变量
```

② 隐式声明。如果不进行显式声明而通过赋值语句直接使用的变量，或省略了[As 类型]短语的变量，其类型为变体类型（Variant）。

注意：如果在程序的开始处写入如下语句：

```
Opting Explicit
```

则程序中所有变量必须进行显式声明，否则当有未定义的变量出现或已定义的变量名发生拼写错误时，系统都会提出警告。

2）变量的作用域。变量的作用域就是变量引用的有效范围。在 Visual Basic 中，变量通常分为局部变量、模块变量和全局变量。

① 局部变量。局部变量又称过程变量。在 Sub 过程中使用 Dim 或 Static 定义的变量属于局部变量，其有效范围在其所声明的过程内部。

使用 Static 定义的变量与 Dim 定义的变量不同之处在于：在执行一个过程结束时，其所用到的 Static 变量的值会保留，下次再调用此过程时，变量的初值是上次调用结束时被保留的值；而 Dim 定义的变量在过程结束时不保留，每次调用时需要重新初始化。

② 模块变量。Visual Basic 程序由窗体模块、标准模块和类模块 3 种模块组成。模块包括过程和声明两部分，在模块的声明部分使用 Private 或 Dim 声明的变量的有效作用范围是模块内部的任何过程，称为模块级变量。

③ 全局变量。全局变量就是可以在整个程序的任何模块、任何过程中使用的变量。在模块的声明部分使用 Public 声明的变量就是全局变量。

4．表达式和运算符

表达式是把常量、变量、函数以及关键字通过运算符按照一定规则组合起来的式子。运算符包括算术运算符、关系运算符、字符串运算符和逻辑运算符。

（1）**算术运算符和表达式** Visual Basic 提供的算术运算符见表 1-1，操作数是数值型数据。

表 1-1 算术运算符

运 算 符	含 义	表达式举例	结 果
+	加	2+3	5
-	减	5-3	2
*	乘	6*3	18
/	除	7/3	2.333333
\	整除	8\3	2
Mod	求余数	25 mod 3	1
^	幂	2^3	8

注：1．除法运算，结果是单精度保留 6 位小数，双精度保留 14 位小数；2．整除和求余数运算，操作数一般是整型或长整型，若为浮点型，则系统会将其四舍五入转换为整型或长整型再进行运算；3．整除运算结果取整不四舍五入。

（2）**字符串运算符和表达式** Visual Basic 有两个字符串连接符：“&”和“+”，用于将两个字符串连接成一个字符串。如：

“程序设计”&“语言” 结果为：“程序设计语言”

123 & 456 结果为：“123 456”

123 &"asd" 结果为：“123asd”

当将上面表达式中的“&”连接符换成“+”运算符时，第一个表达式结果为：“程序设计语言”，第二个表达式的结果是：579，第三个表达式的结果是出错。可以看出，“&”连接符不论两个操作数是字符串还是数值，都可以连接；“+”运算符只有当两个操作数都是字符串时才起连接作用。

（3）**关系运算符和表达式** 关系运算符用于对两个操作数进行比较，如果关系成立，则结果为 True；如果关系不成立，则结果为 False，且两个操作数同为数值型数据或同为字符串数据才能进行比较。常见的关系运算符见表 1-2。

表 1-2 常见的关系运算符

运 算 符	含 义	表 达 式	结 果
>	大于	2+3>8	False
>=	大于等于	5-3>=2	True
<	小于	"3wad"<"3wbf"	True
<=	小于等于	7/3<=3	True
=	等于	"abc"="ABC"	False
<>	不等于	"abc"<>"ABC"	True

注：1．数值型数据比较的是数值的大小；2．字符串比较的是两个字符串中第一个不相同的字符其 ASCII 码的大小。

5. 分支结构与循环结构

（1）**分支结构 If-Then-Else 语句** 语句形式：

```
If<表达式> Then
  <语句块 1>
Else
  <语句块 2>
End If
```

语句的功能是根据条件确定执行不同的语句块：<表达式>为真时，执行<语句块 1>；<表达式>为假时，执行<语句块 2>。

（2）**多分支 If-Then-ElseIf 语句** 语句形式：

```
If<表达式 1> Then
 <语句块 1>
ElseIf <表达式 2> Then
  <语句块 2>
      ……
ElseIf <表达式 n> Then
  <语句块 n>
[Else
  <语句块 n+1> ]
End If
```

语句的功能是根据不同的条件确定执行不同的语句块：当<表达式 i>为真时，执行<语句块 i>。

VB 中 If-Then-ElseIf 语句的条件表达式和语句块的个数没有限制。当选择的情况较多时，VB 提供了 Select Case 语句可以方便简洁地处理多分支的控制结构。

语句形式：

```
Select  Case<测试表达式>
Case    <表达式 1>
        <语句块 1>
Case    <表达式 2>
        <语句块 2>
      ……
Case    <表达式 n>
        <语句块 n>
 [Case  Else
        <语句块 n+1> ]
End  Select
```

语句的功能是根据不同的条件确定执行不同的语句块，即<表达式 i>为真时，执行<语句块 i>。

执行过程说明：

1）首先计算<测试表达式>的值。

2）然后用这个值与<表达式 1>、<表达式 2>、…、<表达式 n>的值相比较。

3）若与表达式 i 的值相匹配，则执行<语句块 i>；执行完语句块 i 后，则结束 Select Case 语句，不再与后面的表达式进行比较，开始执行 End Select 语句后面的语句。

4）当测试表达式的值与后面所有的表达式都不匹配时，若有 Case Else 语句，则执行 Case Else 语句后面的<语句块 n+1>，然后则结束 Select Case 语句；若没有 Case Else 语句，则直接结束 Select Case 语句。

Select Case 语句中的表达式写法有：

1）一个确定的值。例如，

```
Case    1'表示测试表达式的取值为 1
```

2）表达式。例如，

```
Case   a+5   '表示测试表达式的取值为 a+5，a 的值必须是确定的
```

3）用逗号分隔的一组值。例如，

```
Case   1,3,5   '表示测试表达式在 1，3，5 中的取值
```

4）表达式 1 To 表达式 2。例如，

```
Case   20 To 30 '表示测试表达式的取值在 20 至 30 之间
```

5）Is 关系表达式。例如，

```
Case   Is<5    '表示测试表达式的取值在小于 5 的范围，Is 代表测试表达式的值
```

比较两种语句的区别，Select Case 语句的结构清晰，可读性好，但要求多分支的条件表达式应该是相同的，只是不同的值进入不同的分支；当根据不同的判断条件确定进入不同的分支时，只能选择 If-Then-Elseif 语句。

（3）**循环**　所谓循环，就是重复地执行某些操作。在程序设计中，表现为从某处开始规律地反复执行某一程序块，重复执行的程序块称为“循环体”。VB 的循环结构及相应语句表示如下：

循环结构
- 计数型循环　For…Next 语句
- 条件型循环
  - While…Wend 语句
  - Do…Loop 语句

1）计数型循环（For…Next）。在知道要执行多少次循环时，最好用 For…Next 循环结构。

格式：

```
For  <循环变量>=<初值>  To  <终值>  [Step <增量>]
                [<循环体 1>]
                [<Exit For>]
                [<循环体 2>]
Next            [<循环变量>]
```

其中：

①“循环变量”用作循环计数器的数值型变量，“初值”、“终值”均是数值表达式，用于表示循环变量的变动范围。

②“Step”也是一个数值表达式，其值可以是正数（递增循环），也可以是负数（递减循环），但不能为 0。若步长为 1，可略去不写。

2）然后用这个值与<表达式 1>、<表达式 2>、…、<表达式 n>的值相比较。

3）若与表达式 i 的值相匹配，则执行<语句块 i>；执行完语句块 i 后，则结束 Select Case 语句，不再与后面的表达式进行比较，开始执行 End Select 语句后面的语句。

4）当测试表达式的值与后面所有的表达式都不匹配时，若有 Case Else 语句，则执行 Case Else 语句后面的<语句块 n+1>，然后则结束 Select Case 语句；若没有 Case Else 语句，则直接结束 Select Case 语句。

Select Case 语句中的表达式写法有：

1）一个确定的值。例如，

```
Case    1'表示测试表达式的取值为 1
```

2）表达式。例如，

```
Case   a+5   '表示测试表达式的取值为 a+5，a 的值必须是确定的
```

3）用逗号分隔的一组值。例如，

```
Case   1,3,5   '表示测试表达式在 1，3，5 中的取值
```

4）表达式 1 To 表达式 2。例如，

```
Case   20 To 30 '表示测试表达式的取值在 20 至 30 之间
```

5）Is 关系表达式。例如，

```
Case   Is<5    '表示测试表达式的取值在小于 5 的范围，Is 代表测试表达式的值
```

比较两种语句的区别，Select Case 语句的结构清晰，可读性好，但要求多分支的条件表达式应该是相同的，只是不同的值进入不同的分支；当根据不同的判断条件确定进入不同的分支时，只能选择 If-Then-Elseif 语句。

（3）**循环**　所谓循环，就是重复地执行某些操作。在程序设计中，表现为从某处开始规律地反复执行某一程序块，重复执行的程序块称为“循环体”。VB 的循环结构及相应语句表示如下：

循环结构
- 计数型循环　For…Next 语句
- 条件型循环
  - While…Wend 语句
  - Do…Loop 语句

1）计数型循环（For…Next）。在知道要执行多少次循环时，最好用 For…Next 循环结构。

格式：

```
For  <循环变量>=<初值>  To  <终值>  [Step <增量>]
                [<循环体 1>]
                [<Exit For>]
                [<循环体 2>]
Next            [<循环变量>]
```

其中：

①“循环变量”用作循环计数器的数值型变量，“初值”、“终值”均是数值表达式，用于表示循环变量的变动范围。

②“Step”也是一个数值表达式，其值可以是正数（递增循环），也可以是负数（递减循环），但不能为 0。若步长为 1，可略去不写。

③ 循环次数=INT（（终值-初值）/步长）+1。

④ “Exit For”用于中途退出循环，一般与 If 语句联用。

功能：

重复执行 For 和 Next 之间的循环体，重复执行的次数由循环变量来控制。该语句主要用于已知循环次数的循环控制。

2）条件型循环。在很多情况下并不知道循环的次数，VB 提供了条件控制的循环结构，相应语句为 While…Wend 和 Do…Loop。

3）While…Wend 语句。语句的格式为：

While <条件表达式>

[<循环体>]

Wend

功能：当条件表达式的值为 True 时，重复执行循环体；为 False 时，跳出循环，执行 Wend 语句的下一条语句。

必须先给 While 条件中的变量赋值即初始化，在循环体中要有能改变循环条件值的语句，让循环条件表达式最终取 False 值，以结束循环，否则有可能造成“死循环”。

4）Do…Loop 语句。Do…Loop 循环结构较为灵活，有当型（即 While 型）和直到型（即 Until）两种结构。当型结构是条件为真时，执行循环体；直到型结构是条件为真时，结束循环体。根据测试条件在循环体的先后，又分为先判断后执行型和先执行后判断型，二者的区别在于前者循环体有可能一次也不执行，而后者循环体至少执行一次。

具体格式如下：

① Do While…Loop：

```
Do While <循环条件>
    [<循环体 1>]
    [<Exit  Do>]
    [<循环体 2>]
Loop
```

属于当型的先判断后执行结构。

② Do…Loop While：

```
Do
    [<循环体 1>]
    [<Exit  Do>]
    [<循环体 2>]
Loop While <循环条件>]
```

属于当型的先执行后判断结构。

③ Do Until…Loop ：

```
Do Until <循环条件>]
    [<循环体 1>]
    [<Exit  Do>]
    [<循环体 2>]
Loop
```

属于直到型的先判断后执行结构。注意该结构中条件为 True 时，结束循环。

④ Do…Loop Until：

Do

[<循环体 1>]

[<Exit　Do>]

[<循环体 2>]

Loop Until <循环条件>]

属于直到型的先判断后执行结构。注意该结构中条件为 True 时结束循环。

（4）**多重循环**　多重循环就是指循环嵌套，即在一个循环体内有包含另一个或多个完整的循环结构。例如，可以在 For 循环中包含 While 循环、Do 循环或 For 循环。在多重循环中，外面的大循环称为外层循环，里面的小循环称为内层循环。

循环嵌套，应注意以下问题：

1）内层循环一定要包含在外层循环内。

2）内、外层循环不能交叉使用。

3）各层循环的控制变量名应不相同，以免造成混乱。

4）外层循环变量取值一次，内层循环变量取值一遍。

5）内层循环体内的变量取初值，一般应放在内循环之前，外层循环之内。

6．数组与数组元素

我们把具有相同名字、相同数据类型、不同下标的一组变量称为数组。数组中的每一个数组元素是由数组名和带圆括号的下标组成的。数组用于保存大量的、逻辑上有联系的、相同数据类型的数据。例如，一个班级有 30 名学生，若用数组 stu 来表示这 30 个学生的成绩，则可用 stu（1）表示序号为 1 的学生的成绩，stu（2）表示序号为 2 的学生的成绩……

数组名的取名规则和简单变量相同，如 AI、BI、TZ 等均可作为数组名。在同一过程中数组名不应与简单变量名相同。

数组下标：在 VB 中必须把下标放在一对紧跟在数组名后的圆括号中，不能把下标变量 S（7）写成 S7，后者是一个普通的 VB 变量名。下标必须为等于或大于零的整数，否则舍去小数部分自动取整。

下标的作用是指出某个数组元素在数组中的位置，Stu（7）代表了 Stu 数组中的第七个数组元素。

下标的最小值称为下标下界，最大值称为下标上界。由下标的上下界可以确定数组中元素的个数。数组元素的个数称为数组的大小。

数组的特点：数据中的元素在类型上是一致的；数组元素在内存空间上是连续存放的。

（1）**数组的数据类型**　与一般变量类型一样，数组的数据类型也分为单精度、双精度、整数、字符串等。

由于数组是同类数据的集合，因此数组中的所有数组元素应具有相同的数据类型，但如果数组类型是 Variant 时，则数组元素能够为不同的数据类型。

（2）**数组的维数**　只有一个下标的数组称为一维数组，其数组元素称为单下标变量，其下标又称为索引。有两个下标的数组称为二维数组，其数组元素称为双下标变量。

VB 中至多可以使用 16 维的数组。

（3）**数组的定义** 数组必须先声明才使用。声明时要指定数组的类型与数组名。定义一维数组的格式为：

Public | Private | Dim | Static <数组名>（<下标上界>） [AS <数据类型>]

定义二维数组的格式为：

Public | Private | Dim | Static <数组名>（<上界 1>, <上界 2>） [AS <数据类型>]

说明：1）每一维的下标下界值从 0 算起，若要改变下界值可使用 Option Base 语句。即可在窗体或标准模块中，定义数组前将数组下标的默认值下界设置为 1。

Option Base 的语句格式是：

Option Base 1

2）数组一旦定义，就可使用其数组元素，但下标不能超过定义时规定的范围。

3）下标值为长整型，取值范围为（-2147483648～2147483647）。

4）若缺省 As <数据类型>，默认为变体型（variant）。

（4）**静态数组的使用** 要访问数组元素，其格式为：

数组名（下标）

访问数组元素时的下标值必须在所定义的上下界范围内，否则将导致越界错误。

数组的常见操作是数组元素的遍历，利用 For 循环的循环变量和数组元素的下标之间的联系可以很好地处理这类操作。常采用 For 循环结构和赋值语句或 Inputbox 函数完成数组输入，采用 For 循环结构和 Print 方法实现数组输出。

7．过程

VB 中有两类过程：事件过程和通用过程。事件过程是对发生的事件进行处理的代码。

通用过程是解决多个事件调用相同的程序而独立出来的代码。通用过程分为两类，即函数过程和 Sub 过程。

（1）**函数过程（Function 过程）** 函数过程是标准模块中位于 Function 语句与 End Function 语句之间的一系列语句。函数中的这些语句完成某些操作，一般是处理文本、进行输入或计算一个值。通过将函数名和必要的参数一起置于一条程序语句中，可以执行或称作调用该函数。换句话说，使用函数过程与使用内置函数，如 Time、Int 或 Str 等的方法完全相同。

提示：在标准模块中声明的函数在默认状态下是公用函数，它们可在任何事件过程中使用。

函数的基本语法为：

Function 函数名（[参数列表]）[As 数据类型]

　　函数体

End Function

注：函数可以有一个类型。

下列语法成分十分重要：

“函数名”是在模块中要创建函数的函数名称。

“参数列表”为可选项，由函数中用到的一系列参数组成（参数之间用逗号隔开）。

“As 数据类型”为可选项，用于指定函数返回值的数据类型（默认类型为变体类型）。

“函数体”是完成函数功能的一组语句。

函数总是用“函数名”返回给调用过程一个值。因此，函数中的最后一行语句往往是将函数的最终计算结果放入“函数名”中的赋值语句。