计算机等级考试二级 VB 基础教程

1.1 Visual Basic 概述

1.Visual Basic 是什么

Visual Basic(简称 VB)是 Microsoft 公司开发的一种通用的基于对象的程序设计语言。

“Visual” 指的是开发图形用户界面 (GUI) 的方法——不需编写大量代码去描述界面元素的外观和位置，而只要把预先建立的对象 add 到屏幕上的一点即可。

“Basic”指的是 BASIC (Beginners All-Purpose Symbolic Instruction Code) 语言，一种在计算技术发展历史上应用得最为广泛的语言。Visual Basic 在原有 BASIC 语言的基础上进一步发展，至今包含了数百条语句、函数及关键词，其中很多和 Windows GUI 有直接关系。专业人员可以用 Visual Basic 实现其它任何 Windows 编程语言的功能，而初学者只要掌握几个关键词就可以建立实用的应用程序。Visual Basic Scripting Edition (VBScript)是广泛使用的脚本语言，它是 Visual Basic 语言的子集，可嵌入 HTML 语言中，用于网页设计，如 ASP(Active Server Page)文件。

VB 简单易学，通用性强，用途广泛。

2.VB 的功能

VB 可以用于可以开发多媒体、数据库、网络、图形等方面的应用程序。

数据访问特性允许对包括 Microsoft SQL Server 和其它企业数据库在内的大部分数据库格式建立数据库和前端应用程序，以及可调整的服务器端部件。

有了 ActiveX(TM) 技术就可使用其它应用程序提供的功能，例如 Microsoft Word 字处理器，Microsoft Excel 电子数据表及其它 Windows 应用程序。

Internet 能力强大，很容易在应用程序内通过 Internet 或 intranet 访问文档和应用程序，或者创建 Internet 服务器应用程序。

已完成的应用程序是使用 Visual Basic 虚拟机真正 .exe 文件，可以自由发布。

3.VB 的发展

VB 是伴随 Windows 操作系统而发展的，在中国使用较广的版本有 VB4.0、VB5.0、VB6.0。

VB4.0 是为配合 WIN95 的问世于 1995 年推出的，既可用于编写 WIN3.X 平台的 16 位应用程序也可编写 WIN95 平台的 32 位应用程序；VB5.0 主要用于编写 WIN95 平台的 32 位应用程序，较之 VB4.0 主要扩展了数据库、ActiveX 和 Internet 方面的功能；VB6.0 是与 WIN98 配合于 1998 年推出的，进一步加强了数据库、Internet 和创建控件方面的功能。

4.VB 中的几个常用术语

工程(Project)：是指用于创建一个应用程序的文件的集合。
对象(object)：可控制的某个东西,VB 中主要有两类对象:窗体和控件。
窗体(form)：应用程序的用户界面，即 windows。
控件(control)：指的是各种按扭、标签、文本框等。
属性(property)：是指对象的特性，如大小、标题或颜色。

ActiveX:ActiveX 是基于 component object model(com)的可视化控件结构的商标　　名称.它是一种封装技术,提供封装 COM 组件并将其置入应用程序(如 WEB 浏览器)的一种方法.

Components Object Model (COM)：　是软件组件互相通讯的一种方式。它是一种二进制和网络标准，允许任意两个组件互相通讯，而不管它们是在什么计算机上运行(只要计算机是相连的)，不管各计算机运行的是什么操作系统(只要该操作系统支持 COM)，也不管该组件是用什么语言编写的。

5. VB 的系统特性

(1)工程限制

1)代码限制：

可被加载到窗体、类或标准模块的代码总数限于 65,534 行。一行代码限于 1023 个字节。在一行中的实际文本之前最多只能有 256 个空格的前导，在一个逻辑行中最多只能有 25 个续行符 (_)。

2)过程、类型和变量

对每个模块的过程数没有限制。每个过程可包含至多 64K 的代码。如果过程或模块超过这一限制，Visual Basic 便产生编译时间错误。如果遇到这种错误，可将特别大的过程分割成若干个较小的过程，或将模块级声明移到另一模块，来避免此类错误发生。

Visual Basic 用表来保存代码中的标识符名(变量、过程、常量等)。每个表限于 64K。

3)动态链接库声明表

每个窗体和代码模块使用一个描述动态链接库入口点的结构的表。每个结构约 40 个字节，表的大小限于 64K，形成每个模块大约有 1500 个声明。

4)工程名表

整个应用程序用一张包含所有名称的表。这些名称包括： 常量名 、 变量名 、 自定义的类型定义名、 模块名、 DLL 过程声明名。对工程名表总大小没有限制，但是区分大小写的条目不得超过 32K。如果超过了 32K 的限制，可以在不同的模块中重新使用 private 标识符以限制区分大小写的条目数到达 32K。

5)输入表

在不同的模块中每引用一个标识符，便在输入表中创建一个条目。每一个这样的入口最小是 24 字节，但限于 64K，这样每个模块大约可以有 2000 个引用。

6)模块条目表

该表中每个模块最多达 125 个字节，但有 64K 的总限制，这样每个工程大约可以产生 400 个模块。VB 中的变量名不多于 255 个字符，而窗体、控件、模块和类名不多于 40 个字符。

(2)工程文件格式

VB 在创建和编译工程时要产生许多文件，分为：设计时文件、杂项开发文件和运行时文件。

设计时文件是工程的建造块，例如基本模块 (.bas) 和窗体模块 (.frm)。

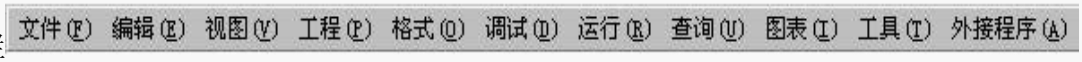
杂项文件是由 Visual Basic 开发环境中的各种不同的进程和函数产生的，例如打包和展开向导从属文件 (.dep)。

1.2 VB 6.0 的集成开发环境简介

VB 集成开发环境 (IDE——Integrated Developing Environment) 由以下元素组成：

1. 标题栏

用于显示正在开发或调试的工程名和系统的工作状态（设计态、运行态、中止态）。

2. 菜单栏 

用于显示所使用的 Visual Basic 命令。VB6.0 标准菜单包括：

3. 工具栏 

在编程环境下用于快速访问常用命令。缺省情况下，启动 VB 后显示“标准”工具栏，附加的编辑、窗体设计和调试的工具 栏可以从“视图”菜单上的“工具栏”命令中移进或移出。

4. 窗体设计器

用来设计应用程序的界面。启动 VB 后，窗体设计器中自动出现一个名为 Form1 的空白窗体，可以在该窗体中添加控件、图形和图片等来创建所希望的外观，窗体的外观设计好后，从菜单中选择“文件”→“保存窗体”→在保存对话框中给出合适的文件名（注意扩展名），并选择所需的保存位置→确定。需要再设计另一个窗体时，单击工具栏上的“添加窗体”按钮即可。

5. 控件（工具）箱

由一组控件按钮组成，用于设计时在窗体中放置控件。除了缺省的工具箱布局之外，还可以通过从上下文菜单中选定“添加选项卡”并在结果选项卡中添加控件来创建自定义布局。

6. 弹出式（上下文）菜单

在要使用的对象上单击鼠标右键即可打开快捷菜单，其上会出现与当前对象相关的经常执行的操作，以加快操作速度。

7. 工程管理器窗口

用于浏览工程中所包含的窗体和模块，还可以从中查看代码、查看对象。

8. 属性窗口

是 VB 中一个比较复杂的窗口，其中列出了对选定窗体和控件的属性设置值。VB 中正是通过改变属性来改变对象的特征，如大小、标题或颜色。

9. 对象浏览器

列出工程中有效的对象，并提供在编码中漫游的快速方法。可以使用“对象浏览器”浏览在 VB 中的对象和其它应用程序，查看对那些对象有效的方法和属性，并将代码过程粘贴进自己的应用程序。

10. 对象浏览器

右击窗体中的对象、从工程管理器中点“查看代码”按钮。

11. 窗体布局窗口

Form Layout window 允许使用表示屏幕的小图象来布置应用程序中各窗体的位置。

12. 立即、本地和监视窗口这些附加窗口是为调试应用程序提供的，它们只在 IDE 之中运行应用程序时才有效。

1.3 面向对象程序设计的基本概念

1.基本术语

(1) 对象和对象类

对象是代码和数据的组合，可以作为一个单位来处理。对象可以是应用程序的一部分，比如可以是控件或窗体。整个应用程序也是一个对象。

VB 中的每个对象都是用类定义的。用饼干模子和饼干之间的关系作比，就会明白对象和它的类之间的关系。饼干模子是类。它确定了每块饼干的特征，比如大小和形状。用类创建对象,对象就是饼干。类是面向对象程序设计的核心技术，可以理解成一种定义了对象行为和外观的模板；把对象看作是类的原原本本的复制品，

类具有继承性、封装性、多态性、抽象性。

(2) 属性

属性是对对象特性的描述，VB 为每一类对象都规定了若干属性，设计中可以改变具体对象的属性值。比如窗体的背景颜色、高度与宽度。

(3) 事件(Event)

事件是发生在对象上的动作。事件的发生不是随意的，某些事件仅发生在某些对象上。

在 VB 中事件的调用形式是：

Private Sub 对象名_事件名

(事件内容)

End Sub

(4) 方法(Method)

方法指的是控制对象动作行为的方式。它是对象本身内含的函数或过程，它也是一个动作，是一个简单的不必知道细节的无法改变的事件，但不称作事件；同样，方法也不是随意的，一些对象有一些特定的方法。在 VB 里方法的调用形式是：

对象名.方法名

2. 属性、方法和事件之间的关系

VB 对象具有属性、方法和事件。属性是描述对象的数据；方法告诉对象应做的事情；事件是对象所产生的事情，事件发生时可以编写代码进行处理。

VB 的窗体和控件是具有自己的属性、方法和事件的对象。可以把属性看作一个对象的性质，把方法看作对象的动作，把事件看作对象的响应。

日常生活中的对象，如小孩玩的气球同样具有属性、方法和事件。气球的属性包括可以看到的一些性质，如它的直径和颜色。其它一些属性描述气球的状态(充气的或未充气的)或不可见的性质，如它的寿命。通过定义，所有气球都具有这些属性；这些属性也会因气球的不同而不同。

气球还具有本身所固有的方法和动作。如：充气方法(用氦气充满气球的动作)，放气方法(排出气球中的气体)和上升方法(放手让气球飞走)。所有的气球都具备这些能力。

气球还有预定义的对某些外部事件的响应。例如，气球对刺破它的事件响应是放气，对放手事件的响应是升空。

在 VB 程序设计中，基本的设计机制就是：改变对象的属性、使用对象的方法、为对象事件编写事件过程。程序设计时要做的工作就是决定应更改哪些属性、调用哪些方法、对哪些事件作出响应，从而得到希望的外观和行为。

3.事件驱动模型

在传统的或“过程化”的应用程序中，应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。从第一行代码执行程序并按应用程序中预定的路径执行，必要时调用过程。

在事件驱动的应用程序中，代码不是按照预定的路径执行，而是在响应不同的事件时执行不同的代码片段。事件可以由用户操作触发、也可以由来自操作系统或其它应用程序的消息触发、甚至由应用程序本身的消息触发。这些事件的顺序决定了代码执行的顺序，因此应用程序每次运行时所经过的代码的路径都是不同的。
因为事件的顺序是无法预测的，所以在代码中必须对执行时的“各种状态”作一定的假设。当作出某些假设时(例如，假设在运行来处理某一输入字段的过程之前，该输入字段必须包含确定的值)，应该组织好应用程序的结构，以确保该假设始终有效(例如，在输入字段中有值之前禁止使用启动该处理过程的命令按钮)。

在执行中代码也可以触发事件。例如，在程序中改变文本框中的文本将引发文本框的 Change 事件。如果 Change 事件中包含有代码，则将导致该代码的执行。如果原来假设该事件仅能由用户的交互操作所触发，则可能会产生意料之外的结果。正因为这一原因，所以在设计应用程序时理解事件驱动模型并牢记在心是非常重要的。

4.交互式开发

传统的应用程序开发过程可以分为三个明显的步骤：编码、编译和测试代码。但是 Visual Basic 与传统的语言不同，它使用交互式方法开发应用程序，使三个步骤之间不再有明显的界限。

VB 在编程者输入代码时便进行解释，即时捕获并突出显示大多数语法或拼写错误。看起来就象一位专家在监视代码的输入。

除即时捕获错误以外，VB 也在输入代码时部分地编译该代码。当准备运行和测试应用程序时，只需极短时间即可完成编译。如果编译器发现了错误，则将错误突出显示于代码中。这时可以更正错误并继续编译，而不需从头开始。

由于 VB 的交互特性，代码运行的效果可以在开发时进行测试，而不必等到编译完成以后。

1．4　利用 VB 开发应用程序的一般步骤

一个 VB 程序也称为一个工程，由窗体、标准模块、自定义控件及应用所需的环境设置组成。开发步骤一般如下：

1. 创建程序的用户界面

2. 设置界面上各个对象的属性

3. 编写对象响应事件的程序代码

4. 保存工程

5. 测试应用程序，排除错误

6. 创建可执行程序

2.1 VB 用户界面设计基础

1. 概述

界面的设计有两步：先绘制控件，然后确定控件属性。

绘制控件：在工具箱里单击想画的控件，在窗体里按下鼠标并拖曳，然后松开鼠标即可。确定属性：先选中控件，然后按 F4 键或单击工具栏上的属性窗口进入属性(Properties)窗口，再在属性窗口中找到要设置的属性并进行设置。

2. 常用属性的设置

(1)Name 属性

对象都有名字，计算机把名字看成对象于对象之间的根本差异，因此在同一窗体里不许出现重名的情况(除非这是一个控件数组)，且名字不得超过 40 个字。

在简单的程序里，给控件命名不是很必要，完全可以使用控件 Name 属性的缺省值。例如 Text1。但在有几十个控件的复杂窗体里，就很难区分它们。所以，VB 推荐由三个小写字母的前缀和一个第一个字母为大写的描述性单词组成的名字。例如 cmdMyButton 是一个命令按钮(前缀是 cmd)

(2)Caption 属性

Caption 即标题，是可以在对象外观上直接看见的文本，可以长达 255 字符，包括空格和标点符号，比如一个叫 cmdOk 的命令钮，它的 Caption 属性就可以是“Ok”。 注意：并不是所有的对象都有此属性，比如文本框、图片框、线条等就没有。

为按钮设置热键：在设置 Caption 属性时，在需要加下划线的字母前加上 “&” 符号，例如 “&File”，输出的就是“File”，这样就可以通过按 ALT 键和标题上那个带下划线的字母来选取它了，不必为此编任何代码。

Name 和 caption 的比较：

1) Name 是系统用来识别对象的，编程时需要用它来指代各对象；Caption 是给用户看的，提示用户该对象的作用；

2) Name 可以采用系统默认的名称，但 Caption 应该根据实际情况改成意义明了的名词；

3)所有对象都有 Name，但不一定都有 Caption；

(3)Top，Left 属性

这两个属性决定对象的位置。只有两种情况需要在属性窗口里设置这两个属性：第一种是用户没有鼠标，第二种是程序员需要十分精确地设定这两个值。当选中对象，单击并拖曳它的时候，便在修改这两个值了。

(4)Height，Width 属性

这两个属性决定了对象的大小，当选中控件时，它周围出现八个小黑方块，把鼠标指向这些方块，鼠标指针将变成一个双向的箭头，这时按下鼠标并拖曳它，即可改变控件的大小，也就改变了 Height，Width 属性。

1. 窗体的属性

(1)设置属性的方法

1）在设计态通过属性窗口设置

直接在属性窗口中选择或输入既可。

2）在程序代码中改变属性值

代码中的格式为： 对象名 . 属性 = 属性值

例：Form1.BackColor=RGB（255，0，0）

2. 窗体的常用方法

（1）Hide 方法

用以隐藏 MDIForm 或 Form 对象，但不能使其卸载。

语法：object.Hide

object 所在处代表一个对象表达式，其值为“应用于”列表中的一个对象。如果省略 object，则带有焦点的窗体就认为是该 object。

说明：

隐藏窗体时，它就从屏幕上被删除，并将其 Visible 属性设置为 False。 用户将无法访问隐藏窗体上的控件，但是对于运行中的 Visual Basic 应用程序，或对于通过 DDE 与该应用程序通讯的进程及对于 Timer 控件的事件，隐藏窗体的控件仍然是可用的。

窗体被隐藏时，用户只有等到被隐藏窗体的事件过程的全部代码执行完后才能够与该应用程序交互。

如果调用 Hide 方法时窗体还没有加载，那么 Hide 方法将加载该窗体但不显示它。

（2）Move 方法

用以移动 MDIForm、Form 或控件。

语法：object . Move left, top, width, height

说明：只有 left 参数是必须的。但是，要指定任何其它的参数，必须先指定出现在语法中该参数前面的全部参数。例如，如果不先指定 left 和 top 参数，则无法指定 width 参数。任何没有指定的尾部的参数则保持不变。

（3）Print 方法

在 Immediate 窗口中显示文本。

语法：object . Print [outputlist]

Print 方法的语法具有下列对象限定符和部分：

部分	描述
Object	必需的。对象表达式，其值为“应用于”列表中的对象。
outputlist	可选的。要打印的表达式或表达式的列表。如果省略，则打印一空白行。

outputlist 参数具有以下语法和部分：

{Spc(n) | Tab(n)} expression charpos

说明：

可以用空白或分号来分隔多个表达式。

对系统指定的国别设置，用小数点分隔符将所有打印到 Immediate 视窗的数据正确格式化。关键字要用适用于主应用程序的语言输出。

对于 Boolean 数据，或者打印 True 或者打印 False。根据主机应用程序的地区设置来翻译 True 和 False 关键字。

使用系统能识别的标准短日期格式书写 Date 数据。当日期或时间部件丢失或为零时，只书写已提供的部件。

如果 outputlist 数据是 Empty，则无内容可写。但是，如果 outputlist 数据是 Null，则输出 Null。在输出 Null 关键字时，要把关键字正确翻译出来。

要把错误数据作为 Error errorcode 输出。在输出 Error 关键字时，要把关键字正确翻译出来。

如果在具有缺省显示空间的模块外使用此方法，则需要 object。例如，如果没有指定对象就在标准模块上调用此方法，则将导致错误发生，但是，如果在窗体模块上进行调用，则会在窗体上显示“outputlist”。

注意： 因为 Print 方法是按照字符比例进行打印，所以字符数与字符所占据的宽度固定的列的数目无关。例如，像 “W” 这样的宽字母占据的宽度超过一固定列宽，而像 "i" 这样的窄字母占据的宽度则较小。考虑到要使用比平均字符更宽的空间，表列一定要留有足够余地。另外，也可以使用固定间距的字体（像 Courier 字体）来确保每一字符均只占一列。

（4）PrintFrom 方法

用以将 Form 对象的图象逐位发送给打印机。

语法：object.PrintForm

说明：PrintForm 将打印 Form 对象的全部可见对象和位图。在绘制图形时，如果 AutoRedraw 属性为 True，则在运行时 PrintForm 将打印 Form 对象或 PictureBox 控件上的图形。

PrintForm 所使用的打印机是由操作系统的控制面板中的设置来决定。

（5） Refresh 方法

强制全部重绘一个窗体或控件。

语法：object.Refresh

说明：在下列情况下使用 Refresh 方法：

在另一个窗体被加载时显示一个窗体的全部。

更新诸如 FileListBox 控件之类的文件系统列表框的内容。

更新 Data 控件的数据结构。

Refresh 方法不能用于 MDI 窗体，但能用于 MDI 子窗体。不能在 Menu 或 Timer 控件上使用 Refresh 方法。

通常，如果没有事件发生，窗体或控件的绘制是自动处理的。但是，有些情况下希望窗体或控件立即更新。例如，如果使用文件列表框、目录列表框或者驱动器列表框显示当前的目录结构状态，当目录结构发生变化时可以使用 Refresh 更新列表。

可以在 Data 控件上使用 Refresh 方法来打开或重新打开数据库（如果 DatabaseName, ReadOnly, Exclusive 或 Connect 属性的设置值发生改变），并能重建控件的 Recordset 属性内的 dynaset。

（6）Show 方法

用以显示 MDIForm 或 Form 对象。

语法：object.Show style, ownerform

说明：

如果调用 Show 方法时指定的窗体没有装载，Visual Basic 将自动装载该窗体。

当 Show 在显示无模式窗体时，随后遇到的代码则要执行。当 Show 在显示模式窗体 (modal form) 时，则随后的代码直到该窗体被隐藏或卸载时才能执行。

当 Show 在显示模式窗体时，除了模式窗体中的对象之外不能进行输入（键盘或鼠标单击）。对其它窗体进行输入前程序必须隐藏或卸载模式窗体（通常是处于响应用户某些操作状态）。MDIForm 不能是形式的。

在模式窗体显示时，虽然应用程序中的其它窗体失效，但其它应用程序不会失效。

应用程序的启动窗体在其 Load 事件调用后会自动出现。

下面的例子说明如何使用 ownerform 参数：

```
Private Sub cmdShowResults_Click()
' 显示模式窗体 frmResults.
frmResults.Show vbModal, Me
End Sub
```

（7）Cls 方法

清除运行时 Form 或 PictureBox 所生成的图形和文本。

语法：object.Cls

说明：

Cls 将清除图形和打印语句在运行时所产生的文本和图形，而设计时在 Form 中使用 Picture 属性设置的背景位图和放置的控件不受 Cls 影响。如果激活 Cls 之前 AutoRedraw 属性设置为 False， 调用时该属性设置为 True，则放置在 Form 或 PictureBox 中的图形和文本也不受影响。 这就是说，通过对正在处理的对象的 AutoRedraw 属性进行操作，可以保持 Form 或 PictureBox 中的图形和文本。

调用 Cls 之后，object 的 CurrentX 和 CurrentY 属性复位为 0。

3.窗体的常用事件

(1) Load 事件

这个事件发生在窗体被装入内存时，且发生在窗体出现在屏幕之前。窗体出现之前，Visual Basic 会看一看 Load 事件里有没有代码，如果有，那么它先执行这些代码，再让窗体出现在屏幕上。

(2) Click 事件，Dblclick 事件

这两个事件在单击或双击窗体时发生。不过单击窗体里的控件时，窗体的 Click 事件并不会发生，Visual Basic 会去看控件的 Click 事件里有没有代码。

(3)Activate (活动事件)与 Deactivate（非活动事件）

显示多个窗体时，可以从一个窗体切换到另一个窗体。每次激活一个窗体时，发生 Activate 事件，而前一个窗体发生 Deactivate 事件。

（4）Resize 事件

在窗体被改变大小时会触发此事件。

4.窗体的控制

(1) 装入或卸出窗体

要装入或卸出窗体，用 Load　或 Unload 语句。

装入窗体： Load formName

卸出窗体： UnLoad formName

FormName 变量是要装入或卸出的窗体名。 Load 语句只是把窗体装入内存，并不显示出来，要显示窗体可以使用窗体的 Show 方法。

(2) 显示或隐藏窗体

要显示或隐藏窗体，用 Show 或 Hide 方法。若尚未装入内存则先装入再显示。

显示窗体： formName.show mode

隐藏窗体： formName.hide

FormName 变量是窗体名,可选变元 mode 为 0（缺省值）时窗体为非模态，为 1 时窗体为模态。模态窗体完全占有应用程序控制权，不允许切换到别的应用程序，除非关闭！而非模态窗体则反之。

（3）END 语句

END 语句的功能是终止应用程序的执行，并从内存卸在所有窗体。

语法是： END

1. 窗体的属性

(1)设置属性的方法

1）在设计态通过属性窗口设置

直接在属性窗口中选择或输入既可。

2）在程序代码中改变属性值

代码中的格式为： 对象名 . 属性 = 属性值

例：Form1.BackColor=RGB（255，0，0）

2. 窗体的常用方法

（1）Hide 方法

用以隐藏 MDIForm 或 Form 对象，但不能使其卸载。

语法：object.Hide

object 所在处代表一个对象表达式，其值为“应用于”列表中的一个对象。如果省略 object，则带有焦点的窗体就认为是该 object。

说明：

隐藏窗体时，它就从屏幕上被删除，并将其 Visible 属性设置为 False。用户将无法访问隐藏窗体上的控件，但是对于运行中的 Visual Basic 应用程序，或对于通过 DDE 与该应用程序通讯的进程及对于 Timer 控件的事件，隐藏窗体的控件仍然是可用的。

窗体被隐藏时，用户只有等到被隐藏窗体的事件过程的全部代码执行完后才能够与该应用程序交互。

如果调用 Hide 方法时窗体还没有加载，那么 Hide 方法将加载该窗体但不显示它。

（2）Move 方法

用以移动 MDIForm、Form 或控件。

语法：object . Move left, top, width, height

Move 方法的语法包含下列部分：

说明：只有 left 参数是必须的。但是，要指定任何其它的参数，必须先指定出现在语法中该参数前面的全部参数。例如，如果不先指定 left 和 top 参数，则无法指定 width 参数。任何没有指定的尾部的参数则保持不变。

（3）Print 方法

在 Immediate 窗口中显示文本。

语法：object . Print [outputlist]

Print 方法的语法具有下列对象限定符和部分：

部分	描述
object	必需的。对象表达式，其值为“应用于”列表中的对象。
outputlist	可选的。要打印的表达式或表达式的列表。如果省略，则打印一空白行。

outputlist 参数具有以下语法和部分：

{Spc(n) | Tab(n)} expression charpos

说明：

可以用空白或分号来分隔多个表达式。

对系统指定的国别设置，用小数点分隔符将所有打印到 Immediate 视窗的数据正确格式化。关键字要用适用于主应用程序的语言输出。

对于 Boolean 数据，或者打印 True 或者打印 False。根据主机应用程序的地区设置来翻译 True 和 False 关键字。

使用系统能识别的标准短日期格式书写 Date 数据。当日期或时间部件丢失或为零时，只书写已提供的部件。

如果 outputlist 数据是 Empty，则无内容可写。但是，如果 outputlist 数据是 Null，则输出 Null。在输出 Null 关键字时，要把关键字正确翻译出来。

要把错误数据作为 Error errorcode 输出。在输出 Error 关键字时，要把关键字正确翻译出来。

如果在具有缺省显示空间的模块外使用此方法，则需要 object。例如，如果没有指定对象就在标准模块上调用此方法，则将导致错误发生，但是，如果在窗体模块上进行调用，则会在窗体上显示“outputlist”。

注意： 因为 Print 方法是按照字符比例进行打印，所以字符数与字符所占据的宽度固定的列的数目无关。例如，像 “W” 这样的宽字母占据的宽度超过一固定列宽，而像 "i" 这样的窄字母占据的宽度则较小。考虑到要使用比平均字符更宽的空间，表列一定要留有足够余地。另外，也可以使用固定间距的字体（像 Courier 字体）来确保每一字符均只占一列。

（4）PrintFrom 方法

用以将 Form 对象的图象逐位发送给打印机。

语法：object.PrintForm

说明：PrintForm 将打印 Form 对象的全部可见对象和位图。在绘制图形时，如果 AutoRedraw 属性为 True，则在运行时 PrintForm 将打印 Form 对象或 PictureBox 控件上的图形。

PrintForm 所使用的打印机是由操作系统的控制面板中的设置来决定。

2.3 控件

1.常用控件介绍

(1)常用控件的属性、方法和事件

公共属性	Name、Caption、Enabled、Fantsize、Height、Width、Index、Left、Top、TabStop
公共方法	Move、Refresh、Setfocus（设置焦点）
公共事件	Click、DblClick、LostFocus（失去焦点）

ComboBox 组合框（将列表框和文本框结合在一起）

Style 属性

外观属性：取 0 时，系统创建一个带下拉式列表框的组合框；为 1 时，系统创建一个由文本框和列表框直接组合在一起的简单组合框，可以从列表框中选择，也可以直接在文本框中输入；为 2 时，系统创建一个没有文本框的下拉式列表框，单击列表框上的按钮才显示文本框，用户不能在文本框中输入，只能在列表框中选择。

Text 属性

其值为用户从列表框中选定的文本或直接输入的文本。

AddItem 方法

添加列表项，使用格式：

[对象名.] AddItem<列表项文本>[，插入位置序号]

若不指定位置，则插入到列表末尾。

Clear 方法

删除列表所有项目

RemoveItem 方法

删除列表项，使用格式：

[对象名 .] RemoveItem 删除项序号

CommandButton

命令按钮

Cancel 属性

取消属性，它为 True 时，按〖ESC〗即等于单击此按钮。

Default 属性

缺省属性，它为 True 时，按回车键即等于单击此按钮。

Timer 计时器

Interval 属性

两次调用 Timer 事件的事件间隔，用于创建动态效果。

Frame 框架

CheckBox 复选框

OptionButton 选项按钮

Alignment 属性

决定它们的对齐方式，0=左对齐，1=右对齐。

Value 属性

决定它们是否被选中的属性

2.常用控件的使用辨析

(1)文本框和标签的区别：

文本框通常用于向计算机输入信息，而标签通常用于输出信息。文本框是一个十分重要的控件，因为由复选框和选项按钮向程序输入的信息毕竟只有少数的几条信息而已。标签和文本框的区别很小，标签可以看成是一个在运行时不能修改正文的文本框，因此标签主要用于输出信息。

(2) Label 的 AutoSize 属性和 WordWrap 属性

为了使标签具有垂直伸展和字换行处理，必须设置它的 AutoSize 属性和 WordWrap 属性同时为 True。

AutoSize 属性为 False，WordWrap 属性为 False 时，若标签不够高而 Caption 太长时，Caption 将被切割掉。

AutoSize 属性为 False，WordWrap 属性为 True 时，情况也如此。

AutoSize 属性为 True，WordWrap 属性为 False 时，表示可以水平伸展，但只显示一行信息。

(3) PictureBox 和 Image 的 Stretch 属性和 AutoSize 属性

Image 只有 Stretch 属性，而 PictureBox 只有 AutoSize 属性。

AutoSize 属性设为 True，则 PictureBox 改变自己的大小来适应其中的图形。

Stretch 属性设为 True，则 Image 中的图形将改变自己的大小来适应外面的边框。

(4)Frame 框架、CheckBox 复选框、OptionButton 选项按钮的区别：

复选框和选项按钮用于向程序输入信息, 框架用来对复选框和选项按钮进行分组。

复选框选中时会在小方框里打一个钩，选项按钮选中时会在小圆圈里点一个点。

2.4 定制菜单

1．菜单概述

Windows 中的菜单一般由菜单条、菜单、菜单项、子菜单、弹出式菜单组成。

2．普通菜单的设计

（1）给菜单命名

菜单标题和菜单命令也有 Caption 和 Name 属性，设置了这两个属性就等于创建了菜单。Name 是一个抽象名称，Caption 是屏幕上可见的，可在 Caption 里加入 “&” 来设置热键。

（2）增加和删除菜单

在 Menu Editor 中部有三个命令钮分别是下一个、插入、删除。插入可用来增加新的菜单。在这三个键下面的 Caption 列表框里选中菜单项（这时它的底色就变成深蓝色），单击插入键，Visual Basic 将上一个增亮菜单下推并增亮一空行，就可以输入新菜单名和标题了。删除键可用来删掉菜单。选中要删掉的菜单，单击 Delete 键就可以删掉它了。

（3）移动菜单标题

有四种情况：向上移动，向下移动，向左缩排，向右缩排，选中某一菜单标题，安上下箭头，则这个菜单将上下移动到你喜欢的位置上，这也决定了它在界面中的位置。如果按左右箭头，情况则有所不同。由于菜单是分级的，所以，如果它没有缩排，则它是一个菜单标题；如果它缩排一次，那么它将变成一个菜单命令；如果缩排两次，那么它将成为一个子菜单命令。VB 里可以总共设计四层子菜单。

（4）设置分离条

分离条是指在菜单中将命令分组的线，VB 将分离条也看成一个菜单项，它也需要 Caption 和 Name 属性，而且也有其它属性，分离条与菜单项的区别是分离条的 Caption 属性必须是连字号即减号。也就是，当设置了一个 Caption 属性为 “-” 的菜单项时，实际上就设置了一个分离条，分离条的名字可以是 barFile1 之类，以表明分离条的位置。

（5） 菜单的各种简单属性

在菜单编辑器里有许多确认框和一些文本框及一个下拉式的列表框，这些决定了菜单的各种属性。

1) Checked 复选属性

这个属性值设置为真，将在菜单命令左边产生一个打勾的确认标志。

2) Enabled 有效属性

各种各样的用户会产生千奇百怪的操作，在许多 Edit 菜单里都会有不同形式的让菜单命令模糊的情况。Enabled 属性为真，则菜单命令是清晰的，Enabled 属性为假，则菜单命令是模糊的，这时用户就不能选中这个菜单项了。

3) Visible 可见属性

对暂时不用的菜单， 如果把 Visible 属性设为假，则菜单根本不会出现在屏幕上。这样做比把 Enabled 属性设为假显得更加干脆！

4) Index 属性

可以生成菜单命令数组，用索引号区分开。例如向 File 菜单中添加一系列最近打开的文件名。添加菜单可用 Load 方法。以上属性可以在运行时设置，形成动态的菜单的情况。

例如：

mnuUndo.Enabled = False

mnuProperty.Visible = False

还可以改变 Caption 等属性。

mnuUndo.Caption = “Redo”

3. 生成弹出式菜单（或浮动菜单）

几乎每个 Windows 应用程序都提供弹出式菜单，用户可以右键单击窗体或控件取得这个菜单。弹出式菜单也属于普通菜单，只是不固定在窗体上，而是可以在任何地方显示。

弹出式菜单用 PopupMenu 方法调用。假设已经用菜单编辑器生成了名为 mnuedit 的菜单，则可以在 MouseUp 事件加入如下代码就可以生成弹出式菜单：

If Button = 2 Then PopupMenu mnuedit

2.5 设计状态条、工具栏、进程条等

1．创建状态条

选中状态条，按 F4 键进入属性窗口，双击（Custom）可以进入主要的设置窗口 SBarCtrl 属性窗口。

（1） 选择面板形状

在 SBarCtrl 属性窗口里选择 General 标签，在 Style 列表框里选择多面板（缺省形式）或单面板简单文本形式。

（2） 添加或删除状态条面板

在 SBarCtrl 属性窗口里选择 Panels 标签，单击 Insert 按钮添加一个面板，或单击 Remove 按钮删除一个面板。

（3） 在单面板里显示文本

在 SBarCtrl 属性窗口里选择 General 标签，在 SimpleText 框里输入想显示在状态条面板里的文本。用代码显示的方式是：

StatusBar1.SimpleText = "New string to appear"

（4） 在多面板里显示文本或图形

1). 在 SBarCtrl 属性窗口里选择 Panels 标签，用 Index 旁的按钮选择面板序号。

2). 在 Text 框里输入想显示在状态条面板里的文本。

3). 如果想加入图形，单击 Browse 按钮打开一个图形选择对话框，选择想加入的图形，然后单击打开按钮。

4). 最后按确定按钮。

5). 用代码显示的方式是：StatusBar1.Panels(x).Text = "New string to appear"

6). 编写代码

如果是一个单面板状态条，当用户单击状态条时，只需用下面的事件过程来响应：

```
Private Sub StatusBar1_Click()

End Sub
```

如果是一个多面板状态条，就需要鉴别用户单击的是哪一个面板，可用下面的事件过程来识别用户所单击的面板：

```
    Private Sub StatusBar1_PanelClick(ByVal Panel As Panel)
        Select Case Panel.Index
            Case 1
                'Code to follow if user clicks the first panel
            Case 2
                'Code to follow if user clicks the second panel
            Case 3
```

```
                    'Code to follow if user clicks the third panel

                End Select

            End Sub
```

2．创建工具栏

（1）建立工具条

1).在工具箱里单击工具条图标，拖到窗体的任何位置，Visual Basic 自动将 ToolBar 移到顶部。

2).按 F4 键打开属性窗口。

3).双击（Custom），打开 ToolBar 属性窗口。

4). 选择 Buttons 标签。

5). 单击 Insert 按钮，Visual Basic 就会在你的工具条上显示一个空按钮，现在为你的每一个按钮重复这一步。

6). 单击确定按钮。

如果想把按钮分组，首先生成一个分隔的按钮，再在刚才提到的对话框里将这个按钮的 Style 属性改为 3-Separator。

（2） 为工具条增加图画

1). 在 Visual Basic 工具箱里单击 ImageList 图标，并将它拖到窗体的任何位置（位置不重要，因为它总是不可见的）。

2). 按 F4 打开属性窗口。

3). 双击（Custom），打开 ImageListCtrl 属性窗口。

4). 选择 Image 标签。

5). 单击 Insert Picture，在现在图形的对话框里选择想使用的位图或图标，然后单击打开按钮。为每个想添加图形的工具条按钮重复此步。

6). 单击确定按钮。

7). 单击工具条，按 F4，双击（Custom），显示 ToolBar 属性对话框。

8). 选择 General 标签，在 ImageList 框里选择刚才添加的 ImageList 控件。

9). 选择 Buttons 标签，单击紧挨 Index 框的向左或向右箭头以选择一个按钮序号。出现在工具条最左边的按钮序号为 1。

10). 在 Image 框里输入一个数，输入为 1，则显示刚才 ImageList 控件里的第一个图形，输入为 2，则显示刚才 ImageList 控件里的第二个图形。

11). 为每个按钮重复第 10 步。

12). 单击确定按钮，现在 Visual Basic 就会在工具条上显示精美的图形了。

2.6 Visual Basic 的输入机制

程序的基本操作就是数据的输入，数据处理和数据的输出。Visual Basic 可中用于输入的控件主要有：文本框 Text Box、复选框 Check Box、选项按钮 Option Button、列表框 List Box、组合框 Combo Box、滚动条 Scroll Bar、通用对话框控件。还有一个函数叫 InputBox 函数，也可用于数据的输入。

1. 通过文本框输入数据

只要取得文本框的 Text 属性就可以对其进行操作了。下面的代码可将 Text1 文本框的正文变成大写输出至 Text2 文本框。

```
Private Sub Form_Click ()
   Text2.Text = UCase(Text1.Text)
End Sub
```

2. 通过复选框和选项按钮输入数据

只要检测到 Check Box 和 Radio Button 的 Value 属性值就可以知道它们是否被选上了。

3. 通过列表框和组合框输入数据

(1) 列表框 List Box 和组合框 Combo Box 的不同

它们在 Windows 的 Open，Save As 对话框里最为常见。组合框又包括三种类型，这三种类型要在它的 Style 属性里设置。列表框仅仅把可以选择的项目列出来，而组合框里有的类型可以允许用户输入数据。

(2) 列表项目的增减

使用 AddItem，RemoveItem 语句可以增减列表项目。通常将项目增减的语句放入 Form_Load 事件里。

```
ListBoxName. AddItem item
ComboBoxName. AddItem item
```

item 是新项目名称，必须是字符串类型，非字符串类型可通过 Str 函数或 Format 函数来转换。转列表框，组合框的每一项目都有一个 Index 值，第一个项目的 Index 值是 0，第二个是 1，依此类推。删除项目可通过删除其 Index 值来实现。

```
ListBoxName. RemoveItem item
ComboBoxName. RemoveItem item
```

如：ListBoxName. RemoveItem 0 则删除了第一项，这时原来的第二项就变成了第一项。

可以在设计阶段设定列表项目，方法是：

在属性窗口选择 List 属性，将出现一个下拉列表，在列表中输入第一项，按 Ctrl+Enter，输入第二项……，最后用 Enter 结束。

(3) 取得列表框/组合框的数据

可以通过取得 Text 属性或通过取得 ListIndex 属性来判断哪一项被选择了。ListIndex 属性的值也同样是第一个为 0，第二个为 1……如：

If List1.Text = “Visual Basic” Then Instructions

If List1.ListIndex = 1 Then Instructions

组合框有时允许用户自己输入数据，这时所输入数据的 ListIndex 值为－1。

4. 通过滚动条 Scroll Bar 取得数据

在 Visual Basic 的工具箱里有两种滚动条：一种是垂直的，一种是水平的，它们的差异无非是摆放的方向不一样。

(1) 滚动条的主要属性及事件

1）Min，Max 属性

Min 属性决定滚动条最左端或最顶端所代表的值。Max 属性决定滚动条最右端或最下端所代表的值。

2）LargeChange,SmallChange 属性

SmallChange 决定在滚动条两端的箭头钮上单击时改变的值。LargeChange 决定在滑块上方或下方区域单击时改变的值。

3）Value 属性

Value 属性代表当前滑块所处位置的值，这个值由滑块的相对位置决定。

4）Change 事件

当滑块位置发生变化时就引发了 Change 事件。

(2) 取得滚动条的数据

滚动条并不是一个数据输入的好控件，因为无法从滚动条上直接看出输入的数据，但它的优点是可以表示一定范围里的相对位置。为了得到数据，只要取得滚动条的 Value 属性就可以了。

5. InputBox 函数

InputBox 函数提供一个简单的对话框供用户输入信息。在把其它版本的 BASIC 程序移植到 Visual Basic 时，InputBox 函数通常用来代替 INPUT 语句。这个对话框的样子基本上是上面的样子。

它的完整语法是：x = InputBox (prompt, title, default, xpos, ypos, helpfile, context)

其中，prompt 是提示的字符串，这个参数是必须的。title 是对话框的标题，是可选的。default 是文本框里的缺省值，也是可选的。xpos，ypos 决定输入框的位置。helpfile，context 用于显示与该框相关的帮助屏幕。返回值 x 将是用户在文本框里输入的数据，x 是一个字符串类型的值。如果用户按了 Cancel 钮，则 x 将为空字符串。

2.7 Visual Basic 的输出机制

作为输出的对象，Visual Basic 提供了标签，文本框，窗体，图片框（Picture Box）等用于输出，而且 Visual Basic 里也有一堆命令、属性涉及信息的输出，较常用的有：MsgBox 函数，Print 方法，Cls 方法，Tab 函数，Format

函数，FontName，FontSize，FontBold，FontItalic，FontStrikethru，FontUnderline 属性，CurrentX，CurrentY 属性等。

1. 通过 MsgBox 函数输出

了解 Windows 应用程序的人都知道当用户操作错误时，应用程序往往打出一个消息框来提示用户的错误，象这样：

```
Dim Action As Integer
Action = MsgBox("单击确定键将引爆此计算机！", vbYesNo + vbCritical + vbDefaultButton2, "警告！")
If Action = 6 Then Explode
```

这个功能在 Visual Basic 里是通过 MsgBox 函数来实现的，这样程序员就不必自己去画一个消息框了。语法是：

```
Action = MsgBox (msg, type, title)
```

其中，msg 是消息的内容，是一个字符串型的变量，type 及 title 参数是可以省略的，type 参数指定显示的按钮是什么及使用什么样的图标（这个参数往往让新程序员思考几分钟），title 参数指定消息框的标题。type 参数的含义是：

例如：Action = MsgBox (“Are you girl?”, vbYesNo + vbQuestio“Question”)。使用这个函数时 Visual Basic 将产生一个标题为 Question，具有问号和 Yes，No 按钮的消息框。作为一个函数，本质上还是要返回值的，MsgBox 的返回值确定了用户的选择，程序可根据返回值做出相应的动作。

2. Print 方法，Cls 方法，Tab 函数

(1) Print 方法

Print 方法的语法是：object. Print expressionlist

object 表示 Print 作用的对象，比如是 Form 或者是 Picture Box，object 是可以省略的，省略时的 Print 往往在窗体上输出。expressionlist 是输出的内容，允许许多项数据的输出，在数据间可以加入“,”或”;”。加入分号将使下一数据项紧挨着上一数据项输出。而加入逗号，则 Visual Basic 将下一数据项在下一打印区输出，其实就是空了几个空格后再输出。两个打印区之间有 14 个字符的宽度，其实这也很难衡量，比如 14 个“W”所占的宽度大概是 14 个“i”所占宽度的三倍多。幸好中国日本之类的方块字所占的宽度基本上是一样的，所以较好处理一些。

(2) Tab 函数

Tab 函数只用在 Print 方法中，它用来移动下一个字符到打印位置，这将省去数空格的功夫。语法是：Tab (column)

Tab 函数将 Visual Basic 自定义的 14 字符宽的打印区扔到一边，用自己的方式定义了新的灵活的打印区宽度。

(3) Cls 方法

Cls 也是一个古老的 BASIC 语句,原来它的作用总是把屏幕变成黑色,然后在左上角或左下角闪烁一个光标，Visual Basic 里它的作用是清除绘图语句和 Print 语句产生的文字和图形。

语法是：object. Cls

object 指清除的对象，可以是 Form 或 Picture Box。如果 object 省略，通常 Visual Basic 都把当前的窗口作为 Cls 操作的对象。

3. Format 函数

Format 函数用于制定字符串或数字的输出格式。语法是：x = Format (expression,fmt)

expression 是所输出的内容。fmt 是指输出的格式，这是一个字符串型的变量，这一项若省略的话，那么 Format 函数将和 Str 函数的功能差不多。

3.1 VB 编码规则

1. 语言元素

VB 的语言基础是 BASIC 语言，VB 程序的语言元素主要由：

关键字（如： Dim、Print、Cls）

函数（如：Sin（）、Cos（）Sqr（））

表达式（如：Abs(-23.5)+45*20/3 ）

语句（如：X=X+5 、 IF……ELSE……END IF）等组成。

2. VB 代码书写规则

（1） 程序中不区分字母的大小写，Ab 与 AB 等效；

（2） 系统对用户程序代码进行自动转换：

1） 对于 VB 中的关键字，首字母被转换成大写，其余转换成小写
2） 若关键字由多个英文单词组成，则将每个单词的首字母转换成大写
3） 对于用户定义的变量、过程名，以第一次定义的为准，以后输入的自动转换成首次定义的形式

3. 语句书写规则

（1）在同一行上可以书写多行语句，语句间用冒号(：)分隔
（2）单行语句可以分多行书写，在本行后加续行符：空格和下划线 _
（3） 一行允许多达 255 个字符

4. 程序的注释方式

（1）整行注释一般以 Rem 开头，也可以用撇号 '；
（2）用撇号 ' 引导的注释，既可以是整行的，也可以直接放在语句的后面，最方便；
（3）可以利用“编辑”工具栏的“设置注释块”、“解除注释块”来将设置多行注释。

5. 保留行号和标号

VB 源程序接受行号与标号，但不是必须的（早期的 BASIC 语言中必须用行号）。
标号是以字母开始以冒号结束的字符串，一般用在 GOTO 语句（现在很少用）中。

3.2 VB 的语言基础

1.数据类型

1）VB 中对没有声明的变量其缺省的数据类型是变体型，可以用来存储各种数据，但所占用的内存比其它类型都多。为提高运行效率(整型效率较高)，或达到一定的运算精确度（浮点型精度较高，但运行较慢），应合理的定义数据类型。

2） 逻辑型数据只有 True 和 False 两个值，转换成整型时，True=-1 ，False=0，将其它类型转换成逻辑型时，非 0 数转换为 True ，0 转换为 False。

3） 字符型可以包括所有的西文字符和汉字，字符必须用双引号括" "起来, 如："abc123"

4）日期型数据按 8 字节的浮点数来存储，日期型数表示方式有两种：可以用号码符#括起来，也可以用数字序列表示（小数点左边的数字代表日期，右边代表时间，0 为午夜，0。5 为中午 12 点，负数表示是 1899 年 12 月 31 日前的日期和时间）。

如：#3/22/2002# #2002-3-22 14:30:20#

```
Dim T As Date
T=-2.5
Print T    ' 打印出来的结果是 1899-12-28 12：00：00
```

5 ）任何数据类型的数组都需要 20 个字节的内存空间，加上每一数组维数占 4 个字节，再加上数据本身所占用的空间。数据所占用的内存空间可以用数据元数目乘上每个元素的大小加以计算。例如，以 4 个 2 字节之 Integer 数据元所组成的一维数组中的数据，占 8 个字节。这 8 个字节加上额外的 24 个字节，使得这个数组所需总内存空间为 32 个字节。

2. 变量与常量

（1）变量或常量的命名规则

1） 必须以字母或汉字开头，由字母、汉字、数字或下划线组成，长度≤255 个字符；

2） 不能使用 VB 中的关键字，并尽量不与 VB 中标准函数名同名；如：Dim 、Sin

3）VB 中不区分变量的大小写，一般变量首字母用大写，其余用小写；常量全部用大写字母表示

4）为了增加程序的可读性，可在变量名前加一个缩写的前缀来表明该变量的数据类型。

（2）变量声明

1）用 Dim 语句进行显式声明

语句形式： Dim 变量名 [As 类型] 如： Dim intX As integer

说明： 如果没有 As 类型，则默认为变体类型。

可在变量名后加类型符来代替 As 类型 如： Dim intX%

一条语句可以同时定义多个变量，但每个变量必须有自己的类型声明，类型声明不能共用；

字符串变量根据其存放的长度是否固定，定义方法不同：

定长字符串： Dim strA As String*10

表示最多存放 10 个字符，如果赋值不足 10 个，则右补空；若多于 10 个，则多余部分截去。

不定长字符串：Dim strA As String ’最多可存放 2MB 个字符

2） 隐式声明

VB 中允许变量不经过声明就直接使用，这种称为隐式声明，所有隐式声明的变量都是变体型的。

隐式声明容易造成错误，为了调试程序方便，一般对使用的变量都进行声明，可以在通用声明段使用 Option Explicit 语句来强制显式声明所有变量。

（3）常量

1）直接常量

指在程序中直接给出值的数据，如：123、123&、123.45、1.234E2 、123D3 分别表示整型、长整型、单精度浮点型（小数形式）、单精度浮点型（指数形式）、双精度浮点型。

八进制常数： 在数值前加&O ，如 &O123

十六进制常数：在数值前加&H ，如&H123

2）用户声明的符号常量

用 Const 来声明： Const 符号常量名 [As 类型] = 表达式

如：Const PI=3.14159 '声明了常量 PI，代表 3.14159，单精度型

Const MAX As Integer=&O144 "声明了常量 MAX，代表八进制数 144，整型

Const COUNT#=45.67 '声明了常量 COUNT，代表 45.67，双精度型

3）系统提供的常量

系统定义的常量位于对象库中，在对象浏览器中的 Visual Basic（VB）和 Visual Basic for Aplication（VBA）等对象库中列出了 VB 的常量。这些常量可以与应用程序的对象、方法、属性一起使用。

如：Form1.WindowsState=vbMaximized 表示将窗口极大化。

3. 运算符

（1）算术运算符

算术运算符两边的操作数应该是数值型，若是数字字符或逻辑型，则自动转换为数值类型后再运算。

（2）字符串运算符

（3） 关系运算符

1）如果两个操作数都是数值型，则按其大小比较

2） 如果两个操作数都是字符型，则按字符的 ASCII 码值从左到右一一比较

3） 汉字字符大于西文字符

4） 关系运算符的优先级相同

5） VB6.0 中 Like 运算符与通配符的使用：

？——表示任何单一字符

*——表示 0 个或多个字符

#——表示任何一个数字（0 - 9）

[字符列表]——表示字符列表中的任何单一字符

[！字符列表]—— 表示不在字符列表中的任何单一字符

（4） 逻辑运算符

1）若有多个条件时，And 必须全部条件为真才为真；
Or 只要有一个条件为真就为真。

2）如果逻辑运算符对数值进行运算，则以数字的二进制值逐位进行逻辑运算。And 运算常用于屏蔽某些位；Or 运算常用于把某些位置 1。

如：12 And 7　表示对 1100 与 0111 进行 And 运算，得到二进制值 100，结果为十进制 4。

3）对一个数连续进行两次 Xor 操作，可恢复原值。在动画设计中，用 Xor 可恢复原来的背景。

4.表达式

（1）表达式的组成

表达式由常量、变量、运算符、函数和圆括号按一定的规则组成，通过运算后有一个结果，运算结果的类型由数据和运算符共同决定。

（2）表达式的书写规则

1）乘号不能省略

2）括号必须成对出现，均使用圆括号，可以嵌套，但必须配对。

3）表达式从左到右在同一基准上书写，无高低、大小之分。

例：sqr((3*x+y)-z)/(x*y)^4

（3）不同数据类型的转换

操作数的数据类型应该符合要求，不同的数据应该转换成同一类型。在算术运算中，如果操作数的数据精度不同，VB 规定运算结果采用精度较高的数据类型。

（4）优先级

同一表达式中，不同运算符的优先级是：算术运算符 > 字符运算符 > 关系运算符 > 逻辑运算符

注意：对于存在多种运算符的表达式，可增加圆括号改变优先级或使表达式更清晰。

4.1 算法概论

1．算法的概念

算法：指用计算机解决某一问题的方法和步骤。

算法分类：　数值算法：用于解决一般数学解析方法难以解决的问题，如：求超越方程的根、求定积分、解微分方程等。

非数值算法：用于对非数值信息进行查找、排序等。

数值算法：用于解决一般数学解析方法难以解决的问题，如：求超越方程的根、求定积分、解微分方程等。

非数值算法：用于对非数值信息进行查找、排序等。

2．算法的特征

（1）确定性：指算法的每个步骤都应确切无误，没有歧义。
（2）可行性：指算法的每个步骤必须是计算机能够有效执行、可以实现的，并可得到确定的结果。
（3）有穷性：指一个算法应该在有限的时间和步骤内可以执行完毕的。
（4）输入性：指一个算法可以有 0 或多个输入数据。
（5）输出性：指一个算法必须有一个或多个输出结果。

3．算法的评价

主要评价指标是：算法是否正确、运行的效率、占用系统资源的多少。

4．算法的描述

一般常用流程框图来描述算法。

5.基本算法结构

“结构化程序设计方法”规定算法有三种基本结构： 顺序结构、选择结构和循环结构

6．算法示例

（1）欧几里德算法——求两个自然数的最大公约数

（2）顺序查找算法—— 在 N 个字符串集合中，查找有无特定的字符串存在

4.2 顺序结构

1. 赋值语句

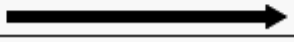
赋值语句是任何程序设计中最基本的语句，赋值语句都时顺序执行的。赋值语句的形式为：

变量名 = 表达式

它的作用是计算右边表达式的值，然后赋给左边的变量，表达式的类型应该与变量名的类型一致。

2. 使用说明

（1）当表达式为数值型而与变量精度不同时，强制转换成左边变量的精度；
（2）当表达式是数字字符串，左边变量是数值类型，自动转换成数值类型再赋值，但当表达式中有非数字字符或空串，则出错。
（3）任何非字符类型赋值给字符类型，自动转换为字符类型；
（4）当逻辑型赋值给数值型时，True 转换为 -1，False 转换为 0；反之，非 0 转换为 True，0 转换为 False；
（5）赋值号左边的变量只能是变量，不能是常量、常数符号、表达式，否则报错；
（6）不能在一句赋值语句中，同时给各变量赋值；
（7）在条件表达式中出现的 = 是等号，系统会根据 = 号的位置，自动判断是否为赋值号；
（8）注意 N=N+1 是累加中常见的赋值语句，表示将 N 变量中的值加 1 后再赋值给 N。

N	执行了 N=N+1 后	N
5		6

4.3　选择结构(或称分支结构)

1. IF 条件语句

(1)If……Then 语句（单分支结构 F）

语句形式：

1)If <表达式> Then

　　语句块

　　End If

2)If　<表达式>　Then　<语句>

说明：表达式一般为关系表达式、逻辑表达式，也可以为算术表达式，非 0 为 True，0 为 False；

语句块可以是一句或多句，若用 2）表示，则只能是一句语句，若多句，语句间需用冒号分隔，而且必须在一行上书写。

例：已知两个数 x 和 y，比较它们的大小，使得 x 大于 y：

方法一：　if　x<y　then

　　t=x

　　x=y

　　y=t

　　end if

方法二：　if　x<y　then　t=x : x=y : y=t

注意：将两个变量中的数进行交换时，必须借助于第三个变量才能实现。

(2)If……Then……Else 语句（双分支结构）

语句形式：

1）If　<表达式>　Then

　　<语句块 1>

　Else

　　<语句块 2>

　End If

2）　If　<表达式>　Then　<语句 1>　Else　<语句 2>

例： x2 -5 x≠0
3x +2 x=0

(3)If……Then……ElseIf 语句（多分支结构）
语句形式：
If <表达式 1> Then
<语句块 1>
ElseIf <表达式 2> Then
<语句块 2>
……
[Else 语句块 n+1]
End If

注意 ：

1）不管有几个分支，程序执行了一个分支后，其余分支不再执行；
2）ElseIf 不能写成 Else If
3）当多分支中有多个表达式同时满足，则只执行第一个与之匹配的语句块

例：已知变量 strC 中存放了一个字符，判断该字符是字母字符、数字字符还是其他字符。

（4）If 语句的嵌套

If 语句的嵌套是指 if 或 else 后面的语句块中又包含 If 语句。语句形式：

If <表达式 1> Then
If <表达式 11> Then
……
End If
……
End If

注意：
1）对于嵌套结构，为了增强程序的可读性，应该采用缩进形式书写；
2）If 语句形式若不在一行上书写，必须与 End If 配对，多个 if 嵌套，End If 与它最接近的 Emd If 配对。
例： 已知 x、y、z 三个数，比较它们的大小并排序，使得 x>y>z.

2. Select Case 语句（情况语句）

Select Case 语句（情况语句）是多分支语句的又一种形式，语句形式：

Select Case 变量或表达式
Case 表达式列表 1
语句块 1
Case 表达式列表 2

语句块 2

……

[Case Else

语句块 n+1]

End Select

说明：1）变量或表达式可以是数值型或字符串表达式

2）表达式列表 I 可以是表达式、一组用逗号分隔的枚举值、表达式 1 to 表达式 2、Is 关系运算符表达式；如：case 1 to 10 、 case "a","w","e","t" 、 case 2,4,6,8,is>10

3）并不是所有的多分支结构都可以用情况语句代替的。

例：已知输入某课程的百分制成绩 mark，要求显示对应五级制的评定，评定条件如下：

优	良	中	及格	不及格
80≤mark< 90	70≤mark< 80	60≤mark< 70	60≤mark< 70	60< mark

3. 条件函数

（1）if 函数

函数形式： Iif（表达式，当条件为 True 时的值，当条件为 False 时的值．例：求 X、Y 中大的数，并放入变量 Tmax 中： Tmax=Iif（X>Y, X, Y）

（2）Choose 函数

函数形式：Choose（整数表达式，选项列表）

如果整数表达式的值是 1，则选择列表中的第 1 项，依次类推；如果小于 1 或大于列表项数时，则返回 NULL。

例： 根据 nub 为 1 -4 的值，换算成不同的运算符： OP= Choose（nub,“＋”，“－”，“×”，“÷”）

4.4 循环结构

循环是在指定的条件下多次重复执行一组语句。VB 中提供了两种类型的循环语句：

计数循环语句和条件型循环语句

1. For 循环语句（知道循环次数的计数型循环）

语句形式： For 循环变量 = 初值 To 终值 [Step 步长]

语句块

[Exit For]

语句块

Next 循环变量

说明: 1） 循环变量必须为数值型
2）步长一般为正，初值小于终值；若为负，初值大于终值；缺省步长为 1；
3）语句块可以是一句或多句语句，称为循环体；
4）Exit For 表示当遇到该语句时，退出循环体；
执行 Next 的下一句；
循环次数=int（（终值 - 初值）/步长+1）
5）退出循环后，循环变量的值保持退出时的值；
6）在循环体内对循环变量可多次引用，但不要对其赋值，否则影响结果。

2. Do……Loop 循环（不知道循环次数的条件型循环）

是用于控制循环次数未知的循环结构，语法形式有两种：

形式 1：Do While …… Loop
Do [While | Until 条件]
语句块
[Exit Do]
语句块
Loop

形式 2：Do …… Loop While
Do
语句块
[Exit Do]
语句块
Loop [While | Until 条件]

说明：
1）形式 1 为先判断后执行，有可能一次也不执行；
2）形式 2 为先执行后判断，至少执行一次；
3）关键字 While 用于指明条件为真时就执行循环体中的语句，Until 刚好相反；
4）当省略了 While|Until 条件字句，即循环结构仅由 Do……Loop 关键字构成时，表示无条件循环，这时循环体内应该有 Exit Do 语句，否则为死循环；
5）Exit Do 语句表示当遇到该语句时，退出循环，执行 Loop 的下一语句。

3. 循环的嵌套

指在循环体内又包含了一个完整的循环结构。循环嵌套对 For 循环和 Do……Loop 循环均适用。

4.5 其它辅助控制语句

1．Go To 语句

语句形式： Go To 标号 | 行号

说明：（1）Go To 语句只能转移到同一过程的标号或行号处；标号是一个字符系列，首字符必须为字母，与大小写无关，任何转移到的标号后面必须有冒号： ；行号是一个数字序列；

（2）以前 BASIC 中常用此语句，可读性差；现在要求尽量少用或不用，改用选择结构或循环结构来代替。

2．Exit 语句

用于退出某控制结构的执行，VB 的 Exit 语句有多种形式，如：

Exit For （退出 For 循环）

Exit Do （退出 Do）循环）

Exit Sub （退出子过程）

Exit Function（退出函数）

3．End 语句

独立的 End 语句用于结束一个程序的执行，可以放在任何事件过程中，形式为：End

VB 的 End 语句还有多种形式，用于结束一个过程或块，如：

End If， End With， End Type， End Select，End Sub， End Function

4．With 语句

它的作用是可以对某个对象执行一系列的语句，而不用重复指出对象的名称。但不能用一个 With 语句设置多个不同的对象。属性前面需要带点号“. ”。

语句形式如下： With 对象名

语句块

End With

例：With form1

. Height=3000

. Width=4000

. BackColor=RGB(255,0,0)

End With

4.6 程序调试

1．VB 的调试工具

（1）设置自动语法检查

打开工具菜单 → 单击选项命令 → 选择 编辑器 页面 →将 自动语法检测 勾上。

（2）利用 VB 调试工具栏

利用该工具栏可以运行程序、中断运行、在程序中设置间断点、监视变量、单步调试、过程跟踪等操作。

2． VB 的三种模式

（1）设计模式

在设计模式下可以进行程序的界面设计、属性设置、代码编写等，标题栏上显示“设计”，在此模式下不能运行程序，也不能使用调试工具。

（2）运行模式

执行“运行”菜单中的“启动”命令或单击工具栏上的启动按钮或按 F5 键，即由设计模式进入运行模式，标题栏显示“运行”，在此阶段可以查看程序代码，但不能修改。若要修改，必须单击工具栏上的“结束”按钮，回到设计模式，也可以选择“中断”按钮，进入中断模式。

（3）中断模式

当程序运行时单击了“中断”按钮，或当程序出现运行错误时，都可以进入中断模式，在此模式下，运行的程序被挂起，可以查看代码、修改代码、检查数据。修改结束，单击“继续”按钮可以继续程序的运行，也可以单击“结束”按钮停止程序的执行。

3. 常见错误

（1） 编辑时错误

当用户在代码窗口编辑代码时，VB 会对程序进行语法检查，当发现语句没有输完、关键字输错等情况时，系统会弹出对话框，提示出错，并在错误处加亮显示，以便用户修改。

（2）编译时错误

是指用户单击了“启动”按钮，VB 开始运行程序前，先编译执行的程序段时产生的错误，此错误是由于用户未定义变量、遗漏关键字等原因而产生的。发现错误时系统会停止编译，提示用户修改。

（3）运行时错误

指 VB 在编译通过后，运行代码时发生的错误，一般是由于指令代码执行了非法操作引起的，如：数据类型不匹配、试图打开一个不存在的文件等。系统会报错并加亮显示、等候处理。

（4）逻辑错误

如果程序运行后得不到所希望的结果，则说明存在逻辑错误。如：运算符使用不正确，语句的次序不对、循环语句的起始、终值不正确。这种错误系统不会报错，需要用户自己分析判断。

4.程序调试方法

（1）进入/退出中断状态

进入中断状态有四种方法：

1） 程序运行时发生错误自动进入中断

2） 程序运行中用户按中断键强制进入中断

3） 用户在程序中预先设置了断点，程序执行到断点处即进入中断状态

4） 在采用单步调试方式，每运行一个可执行代码后，即进入中断状态。

（2） 利用调试窗口

1) 立即窗口

这是调式窗口中使用最方便、最常用的窗口。可以在程序中用 Debug.Print 方法，把输出送到立即窗口，也可以在该窗口中直接使用 Print 语句或？显示变量的值。

2) 本地窗口

该窗口显示当前过程中所有变量的值，当程序的执行从一个过程切换到另一个过程时，该窗口的内容发生改变，它只反映当前过程中可用的变量。

3) 监视窗口

该窗口可显示当前的监视表达式，在此之前必须在设计阶段，利用调试菜单的“添加监视命令”或“快速监视”命令添加监视表达式以及设置的监视类型在运行时显示在监视窗口，根据设置的监视类型进行相应的显示。

（3）插入断点和逐句跟踪

在调试程序时，通常回设置断点来中断程序的运行，然后逐句跟踪检查相关变量、属性和表达式的值是否在预期的范围内。

可在中断模式下或设计模式时设置或删除断点，在代码窗口选择怀疑存在问题的地方作为断点，按下 F9 键，则程序运行到断点处即停下，进入中断模式，在此之前所关心的变量、属性、表达式的值都可以看到。

5.1　数组的概念

1．引例

求 100 个学生的平均成绩及超过平均成绩的人数。

如果用一般变量来表示成绩，需要用 100 个变量，如：mark1、mary2、……mark100。若用数组，可以只用一个来表示 mark（1 To　100）。

2．基本概念

数组：是同类型变量的一个有序的集合。

如：A（1 To 100），表示一个包含 100 个数组元素的名为 A 的数组。

数组元素：　即数组中的变量。用下标表示数组中的各个元素。

表示方法：数组名（P1，P2，……）

其中 P1、P2 表示元素在数组中的排列位置，称为“下标”。

如：A（3，2）代表二维数组 A 中第 3 行第 2 列上的那个元素。

数组维数：　由数组元素中下标的个数决定，一个下标表示一维数组，二个下标表示二维数组。

VB 中有一维数组、二维数组、……最多 60 维数组。

下标：　下标表示顺序号，每个数组有一个唯一的顺序号，下标不能超过数组声明时的上、下界范围。下标可以是整型的常数、变量、表达式，甚至又是一个数组元素。

下标的取值范围是：下界 To 上界 ，缺省下界时，系统默认取 0。

3．数组声明

数组必须先声明后使用。声明数组就是让系统在内存中分配一个连续的区域，用来存储数组元素。

声明内容：数组名、类型、维数、数组大小。

一般情况下，数组中各元素类型必须相同，但若数组为 Variant 时，可包含不同类型的数据。

静态数组：声明时确定了大小的数组。

动态数组： 声明时没有给定数组大小（省略了括号中的下标），使用时需要用 ReDim 语句重新指出其大小。

使用动态数组的优点是根据用户需要，有效地利用存储空间，它是在程序执行到 ReDim 语句时才分配存储单元，而静态数组是在程序编译时分配存储单元。

5.3 动态数组及声明

1. 动态数组的建立与声明

建立动态数组的方法是：利用 Dim、Private、Public 语句声明括号内为空的数组，然后在过程中用 ReDim 语句指明该数组的大小。语法是：

ReDim 数组名（下标 1[，下标 2…]） [As 类型]

其中下标可以是常量，也可以是有了确定值的变量，类型可以省略，若不省略，必须与 Dim 中的声明语句保持一致。

例： Dim D（） As Single
Sub Form_Load（）
……
ReDim D（4，6）
……
End Sub

2．注意事项

（1）在动态数组 ReDim 语句中的下标可以是常量，也可以是有了确定值的变量。

（2）在过程中可以多次使用 ReDim 来改变数组的大小，也可改变数组的维数。

（3）每次使用 ReDim 语句都会使原来数组中的值丢失，可以在 ReDim 语句后加 Preserve 参数来保留数组中的数据，但使用 Preserve 只能改变最后一维的大小，前面几维大小不能改变。

1. 给数组元素赋初值

（1） 利用循环结构

例：Dim iA（1 To 10） As Integer
For i=1 To 10
A(i)=0
Next i

（2） 利用 Array 函数

例：Dim a As Variant, b As Variant, i%
a = Array (1,2,3,4,5)
b = Array ("abc","def","67")
For i=0 To Ubound (a)
Picture1.print a(i);"";
Next i
For i=0 To Ubound (b)
Picture1.print b(i);"";
Next i

2．数组的输入

(1) 通过 InputBox 函数输入适合输入少量数据。

例：Dim sB（3,4） As singer
For i=0 To 3
For j=0 To 4
SB(i,j) =InputBox("输入" & i & j & "的值")
Next j
Next i

(2) 通过文本框控件输入

对大批量的数据输入，采用文本框和函数 split()\join()进行处理，效率更高。

3．数组的赋值

在 VB6.0 中可以直接将一个数组的值赋值给另一个数组：

```
Dim a(3) as integer, b() as integer
A(0)=2: A(1)=5: A(2)=-2: A(3)=2
  b=a
```

在早期的 VB 中，这需要用循环语句才可以实现：

```
ReDim b(UBound(a))
For I=0 to UBound(a)
      b(I)=a(I)
   Next i
```

注意：（1）赋值号两边的数据类型必须一致；

（2）如果赋值号左边的是一个动态数组，则赋值时系统自动将动态数组 ReDim 成右边相同大小的数组；

（3）如果赋值号左边的是一个大小固定的数组，则数组赋值出错。

4．数组的输出

用 For……Next 循环语句输出。

5．求数组中最大元素和所在下标及各元素之和

求数组中最大元素及下标，一般假设第一个元素及下标为最大，然后将该数与数组中的其他元素逐一比较，若有比其大的就替换，同时替换下标。

6．交换数组中各元素

交换的要求是将数组第一个元素与最后一个交换，第二个与倒数第二个交换，依次类推。

5.5 控件数组

1．控件数组的概念

控件数组是由一组相同类型的控件组成的，它们共用一个控件名，具有相同的数组。控件数组适用于若干个控件执行的操作相似的场合，控件组共享同样的事件过程。控件数组通过索引号（属性中的 Index）来标识各控件，第一个下标是 0 。如：Text1(0)、Text1(1)、Text1(2)、Text1(3)……

2．控件数组的建立

（1）在设计时建立

步骤：1）在窗体上画出某控件，并进行属性设置。

2）选中该控件进行“复制”和“粘贴”操作，系统提示“是否建立控件数组”，选择是即可。多次粘贴就可以创建多个控件元素。

3）进行事件过程的编程。

（2）运行时添加控件数组

方法：1）在窗体上画出某控件，设置该控件的 Index 值为 0，表示该控件为数组。

2）在编程时通过 Load 方法添加其余若干个元素，也可以通过 Unload 删除某个添加的元素。

3）每个添加的控件数组通过 Left 和 Top 属性，确定其在窗体上的位置，并将 Visible 设置为 True。

使用示例：建立一个类似国际象棋的棋盘，要求黑白交替，运行时单击某个棋格，会改变颜色并显示其序号。

5.6 自定义数据类型

1. 自定义数据类型的定义

是指由若干标准数据类型组成的一种复合类型，也称为记录类型。

（1）定义方式：

```
Type 自定义类型名
    元素名[（下标）] As 类型名
      ……
    元素名[（下标）] As 类型名
End Type
```

元素名：表示自定义类型中的一个成员

下标（可选）：表示是数组

类型名：为标准类型

例：定义一个学生信息的自定义类型：

```
Type studtype
    No   As  Integer      ’定义学号
    Name  As  String*10   ’定义姓名
    Sex  As  String*2     ’定义性别
    Mark(1 TO 4) As Single  ’定义 4 门课程的成绩
    Total As Single       ’定义总分
End Type
```

（2）注意事项

1）自定义类型一般在标准模块（.bas）中定义，默认是 Public

2）自定义类型中的元素可以是字符串,但应是定长字符串

3) 不可把自定义类型名与该类型的变量名混淆

4) 注意自定义类型变量与数组的差别：它们都由若干元素组成，前者的元素代表不同性质、不同类型的数据，以元素名表示不同的元素；后者存放的是同种性质、同种类型的数据，以下标表示不同元素。

2. 自定义型变量的声明和使用

使用形式： Dim 变量名 As 自定义类型名

例如： Dim student As studtype, mystud As studtype
自定义类型中元素的表示方法是：变量名 . 元素名 如：student.name student.mark(4)
为了简单起见，可以用 With …… End With 语句进行简化。 例：

```
With  student
.no=99001
.name=""
```

```
.sex=""
.total=0
  for I=1 to 4
.mark(I)=int(rnd*101)     ’随机产生 0 - 100 之间的分数
.total=.total+.may(I)
next I
End With
Mystud=student      ’同种自定义类型变量可以直接赋值
```

3.自定义类型数组的使用

自定义类型数组就是数组中的每个元素都是自定义类型。

例如：自定义一个由学生姓名、成绩组成的学生记录类型，用来存放 100 个学生的记录。

6.1VB 的过程设计及子过程（Sub）的定义与调用

1．什么是过程

在程序设计中，为各个相对独立的功能模块所编写的一段程序称之为过程。

2．VB 中的自定义过程分类

（1）以“Sub”保留字开始的子程序过程 （包括事件过程和通用过程），不返回值；

（2）以“Function” 保留字开始的函数过程，返回一个值；

（3）以“Property” 保留字开始的属性过程，可以返回和设置窗体、标准模块以及类模块的属性值，也可以设置对象的值。

3． 事件过程

（1）窗体事件过程

语法：Private Sub Form_事件名（[参数列表]）

[局部变量和常数声明]

语句块

End Sub

注意：

1）窗体事件过程名由 Form_事件名组成，多文档窗体用 MDIForm_事件名；

2）每个窗体事件过程名前都有一个 Private 的前缀，表示该事件过程不能在它自己的窗体模块之外被调用；

3）事件过程有无参数，完全由 VB 提供的具体事件本身决定，用户不可以随意添加。

（2）控件事件过程

语法：Private Sub 控件名_事件名（[参数列表]）

[局部变量和常数声明]

语句块

End Sub

注意：其中的控件名必须与窗体中某控件相匹配，否则 VB 将认为它是一个通用过程。

（3）建立事件过程的方法

1）打开代码编辑器窗口（两种方法：双击对象或从工程管理器中单击“查看代码”按钮）

2）在代码编辑器窗口中，选择所需要的“对象”和“事件过程”

3）在 Private Sub …… End Sub 之间键入代码

4）保存工程和窗体

（4）事件过程的调用

事件过程由一个发生在 VB 中的事件来自动调用或者由同一模块中的其他过程显示调用。

2．通用过程

通用过程是一个必须从另一个过程显示调用的程序段，通用过程有助于将复杂的应用程序分解成多个易于管理的逻辑单元，使应用程序更简洁、更易于维护。

通用过程分为公有（Public）过程和私有（ Private）过程两种，公有过程可以被应用程序中的任一过程调用，而私有过程只能被同一模块中的过程调用。

（1）定义方法：

[Private | Public] [Static] Sub 过程名（[参数列表]）

[局部变量和常数声明] ‘用 Dim 或 Static 声明

语句块

[Exit Sub]

语句块

End Sub

注意：

1）缺省[Private | Public]时，系统默认为 Public ；

2）Static 表示过程中的局部变量为“静态”变量；

3）过程名的命名规则与变量命名规则相同，在同一个模块中，同一符号名不得既用作 Sub 过程名，又用作 Function 过程名。

4）参数列表中的参数称为形式参数，它可以是变量名或数组名，只能是简单变量，不能是常量、数组元素、表达式；若有多个参数时，各参数之间用逗号分隔，形参没有具体的值。VB 的过程可以没有参数，但一对圆括号不可以省略。不含参数的过程称为无参过程。

形参格式为：

[ByVal] 变量名[（）] [As 数据类型]

式中：

变量名[（）]：变量名为合法的 VB 变量名或数组名，无括号表示变量，有括号表示数组。

ByVal ：表明其后的形参是按值传递参数（传值参数 Passed By Value），若缺省或用 ByRef，则表明参数是按地址传递的（传址参数）或称“引用”（Passed By Reference）。

As：数据类型：缺省表明该形参是变体型变量，若形参变量的类型声明为 String，则只能是不定长的。而在调用该过程时，对应的实在参数可以是定长的字符串或字符串数组，若形参是数组则无限制。

5）Sub 过程不能嵌套定义，但可以嵌套调用。

6） End Sub 标志该过程的结束，系统返回并调用该过程语句的下一条语句。

7） 过程中可以用 Exit Sub 提前结束过程，并返回到下调用该过程语句的下一条语句。

（2）建立 Sub 过程的方法

方法一：

1）打开代码编辑器窗口

2）选择“工具”菜单中的“添加过程”

3）从对话框中输入过程名，并选择类型和范围

4）在新创建的过程中输入内容

方法二：

1）在代码编辑器窗口的对象中选择“通用”，在文本编辑区输入 Private Sub 过程名

2）按回车键，即可创建一个 Sub 过程样板

3）在新创建的过程中输入内容

3．Sub 子过程的调用

（1） 用 Call 语句调用 Sub 过程

语法：Call 过程名（实在参数表）

实在参数的个数、类型和顺序，应该与被调用过程的形式参数相匹配，有多个参数时，用逗号分隔。

（2）把过程名作为一个语句来用

语法：过程名 [实参 1[，实参 2……]]

它与（1）的不同点是：去掉了关键字和实参列表的括号

如上例中可以改成：area a,b,c,w

6.3 参数的传递

1．形参与实参的概念

形参：指出现在 Sub 和 Function 过程形参表中的变量名、数组名，过程被调用前，没有分配内存，其作用是说明自变量的类型和形态以及在过程中的角色。形参可以是：

1）除定长字符串变量之外的合法变量名；

2）后面跟（）括号的数组名。

实参：是在调用 Sub 和 Function 过程时，传送给相应过程的变量名、数组名、常数或表达式。在过程调用传递参数时，形参与实参是按位置结合的，形参表和实参表中对应的变量名可以不必相同，但位置必须对应起来。

形参与实参的关系：形参如同公式中的符号，实参就是符号具体的值；调用过程：即实现形参与实参的结合，也就是把值代入公式进行计算。

2． 按值传递参数（定义时加 ByVal）

按值传递参数（Passed By Value）时，是将实参变量的值复制一个到临时存储单元中，如果在调用过程中改变了形参的值，不会影响实参变量本身，即实参变量保持调用前的值不变。

3．按地址传递参数（定义时没有修饰词或带关键字 ByRef）

按地址传递参数时，把实参变量的地址传送给被调用过程，形参和实参共用内存的同一地址。在被调用过程中，形参的值一旦改变，相应实参的值也跟着改变。如果实参是一个常数或表达式，VB 会按“传值”方式来处理。

4．数组参数

VB 允许把数组作为形参出现在形参表中，语法： 形参数组名（） [As 数据类型]

形参数组只能按地址传递参数，对应的实参也必须是数组，且数据类型相同。调用过程时，把要传递的数组名放在实参表中，数组名后面不跟圆括号。在过程中不可以用 Dim 语句对形参数组进行声明，否则会产生“重复声明”的错误。但在使用动态数组时，可以用 ReDim 语句改变形参数组的维界，重新定义数组的大小。

5． 对象参数

VB 中可以向过程传递对象，在形参表中，把形参变量的类型声明为“Control”，可以向过程传递控件；若声明为“Form”，则可向过程传递窗体。对象的传递只能按地址传递。

6.4 变量、过程的作用域

1． VB 应用程序的组成：

2． 过程的作用域

作用范围	模块级		全局级	
	窗体	标准模块	窗体	标准模块
定义方式	过程名前加 Private 例：Private Sub my1（形参表）		过程名前加 Pubilc 或默认 例：[Pubilc] Sub my2（形参表）	
能否被本模块其他过程调用	能	能	能	能
能否被本应用程序其他模块调用	不能	不能	能，但必须在过程名前加窗体名。例：Call 窗体名. My1（实参表）	能，但过程名必须唯一，否则需要加标准模块名。例：Call 标准模块名.My2（实参表）

3． 变量的作用域

作用范围	局部变量	窗体/模块级变量	全局变量	
			窗体	标准模块
声明方式	Dim、Static	Dim、Private	Public	
声明位置	在过程中	窗体/模块的“通用声明”段	窗体/模块的“通用声明”段	
能否被本模块其他过程存取	不能	能	能	
能否被其他模块存取	不能	不能	能，但在变量名前加窗体名	能

4．静态变量

用 Static 声明的静态变量，在每次调用过程时保持原来的值，不重新初始化。而用 Dim 声明的变量，每次调用过程时，重新初始化

例：显示 1 到 5 个数。

5．同名变量

对不同范围内出现的同名变量，可以用模块名加以区别。一般情况下，当变量名相同而作用域不同时，优先访问局限性大的变量。

6.5 递归过程

1．递归的概念

通俗的讲，用自身的结构来描述自身就称为“递归”。如对阶乘运算的定义就是递归的：

n!=n(n-1)!　　(n-1)!=(n-1)(n-2)!

2．递归子过程和递归函数

VB 允许一个自定义子过程或函数过程在过程体的内部调用自己，这样的子过程或函数就叫递归子过程和递归函数。递归过程包含了递推和回归两个过程。构成递归的条件是：

（1）递归结束条件和结束时的值

（2）能用递归形式表示，并且递归向结束条件发展。

例：编制程序求 fac（n）=n！的函数

3．注意事项

（1）递归算法设计简单，但消耗的上机时间和占据的内存空间比非递归大

（2） 设计一个正确的递归过程或函数过程必须具备两点：

1）具备递归条件；

2）具备递归结束条件

（3） 一般而言，递归函数过程对于计算阶乘、级数、指数运算有特殊效果。

7.1 文件系统控件

1．文件系统控件种类

（1） 驱动器列表框（DriveListBox）：用来显示当前机器上的所有盘符

（2） 目录列表框（DirListBox）：用来显示当前盘上的所有文件夹

（3） 文件列表框（FileListBox）：用来显示当前文件夹下的所有文件名

2．重要属性

属性	适用的控件	作用	示例
Drive	驱动器列表框	包含当前选定的驱动器名	Driver1. Drive="C"
Path	目录和文件列表框	包含当前路径	Dir1.Path="C:\WINDOWS"
FileName	文件列表框	包含选定的文件名	MsgBox File1.FileName
Pattern	文件列表框	决定显示的文件类型	File1.Pattern="*.BMP"

3．重要事件

事件	适用的控件	事件发生的时机
Change	目录和驱动器列表框	驱动器列表框的 Change 事件是在选择一个新的驱动器或通过代码改变 Drive 属性的设置时发生。目录列表框的 Change 事件是在双击一个新的目录或通过代码改变 Path 属性的设置时发生
PathChange	文件列表框	当文件列表框的 Path 属性改变时发生
PattenChange	文件列表框	当文件列表框的 Pattern 属性改变时发生
Click	目录和文件列表框	用鼠标单击时发生
DblClick	文件列表框	用鼠标双击时发生

7.2 文件的读写

1. 文件的有关概念

记录：计算机处理数据的基本单位，由若干个相互关联的数据项组成。相当于表格中的一行。

文件：记录的集合，相当于一张表。

文件类型：顺序文件、随机文件、二进制文件。

访问模式：计算机访问文件的方式，VB 中有顺序、随机、二进制三种访问模式。

2. 顺序访问模式

顺序访问模式的规则最简单，指读出或写入时，从第一条记录“顺序”地读到最后一条记录，不可以跳跃式访问。该模式专门用于处理文本文件，每一行文本相当于一条记录，每条记录可长可短，记录与记录之间用“换行符”来分隔。

顺序文件的写入步骤：打开、写入、关闭； 读出步骤：打开、读出、关闭。

（1）打开文件

打开文件的命令是 Open，格式为：

Open “文件名” For 模式 As [#] 文件号 [Len=记录长度]

说明：

1）文件名可以是字符串常量也可以是字符串变量

2）模式可以是下面之一：

OutPut：打开一个文件，将对该文件进行写操作

Input：打开一个文件，将对该文件进行读操作

Append：打开一个文件，将在该文件末尾追加记录

3） 文件号是一个介于 1-511 之间的整数，打开一个文件时需要指定一个文件号，这个文件号就代表该文件，直到文件关闭后这个号才可以被其他文件所使用。可以利用 FreeFile（）函数获得下一个可以利用的文件号。

例： Open "D:\sj\aaa" For Output As #1

意思是：打开 D:\SJ 下 aaa 文件供写入数据，文件号为#1

（2）写操作

将数据写入磁盘文件所用的命令是：Write# 或 Print#。

语法格式：

1） Print #文件号，[输出列表]

例： Open “D:\SJ\TEST.DAT” For Output As #1

Print #1,Text1.Text '把文本框的内容一次性写入文件

Close #1

2）Write #文件号，[输出列表]

其中的输出列表一般指用逗号，分隔的数值或字符串表达式。Write #与 Print #的功能基本相同，区别是 Write #是以紧凑格式存放，在数据间插入逗号，并给字符串加上双引号。

（3） 关闭文件

结束各种读写操作后，必须将文件关闭，否则会造成数据丢失。关闭文件的命令是 Close。

Close [#]文件号[，[#]文件号]……

例： Close #1，#2，#3

（4） 读操作

1） Input #文件号，变量列表

作用：将从文件中读出的数据分别赋给指定的变量。

注意：与 Write #配套才可以准确地读出。

2）Line Input #文件号，字符串变量

用于从文件中读出一行数据，并将读出的数据赋给指定的字符串变量，读出的数据中不包含回车符和换行符，可与 Print #配套用。

3）Input$（读取的字符数，#文件号）

该函数可以读取指定数目的字符。

与读文件有关的两个函数：

LOF（）：返回某文件的字节数

EOF（）：检查指针是否到达文件尾。

例：将一个文本文件读入文本框的三种方法。

3.随机访问模式

该模式要求文件中的每条记录的长度都是相同的，记录与记录之间不需要特殊的分隔符号。只要给出记录号，可以直接访问某一特定记录，其优点是存取速度快，更新容易。

（1） 打开与关闭

打开：Open “文件名” For Random As [#] 文件号 [Len=记录长度]

关闭：Close #文件号

注意：文件以随机方式打开后，可以同时进行写入和读出操作，但需要指明记录的长度，系统默认长度为128 个字节。

（2）读与写

读操作：Get [#]文件号，[记录号]，变量名

说明：Get 命令是从磁盘文件中将一条由记录号指定的记录内容读入记录变量中；记录号是大于 1 的整数，表示对第几条记录进行操作，如果忽略不写，则表示当前记录的下一条记录。

写操作：Put [#]文件号，[记录号]，变量名

说明：Put 命令是将一个记录变量的内容，写入所打开的磁盘文件指定的记录位置；记录号是大于 1 的整数，表示写入的是第几条记录，如果忽略不写，则表示在当前记录后插入一条记录。

4. 二进制访问模式

打开：Open “文件名” For Binary As [#] 文件号 [Len=记录长度]

关闭：Close #文件号

该模式是最原始的文件类型，直接把二进制码存放在文件中，没有什么格式，以字节数来定位数据，允许程序按所需的任何方式组织和访问数据，也允许对文件中各字节数据进行存取和访问。

该模式与随机模式类似，其读写语句也是 Get 和 Put，区别是二进制模式的访问单位是字节，随机模式的访问单位是记录。在此模式中，可以把文件指针移到文件的任何地方，刚开始打开时，文件指针指向第一个字节，以后随文件处理命令的执行而一旦。文件一旦打开，就可以同时进行读写。

7.3 常用的文件操作语句和函数

1．FileCopy 语句 格式：FileCopy 源文件名 目标文件名

功能：复制一个文件

说明：不能复制一个已打开的文件

2．Kill 语句

格式：Kill 文件名

功能：删除文件

说明：文件名中可以使用通配符 *，？

3．Name 语句

格式：Name 旧文件名 新文件名

功能：重新命名一个文件或目录

说明：不能使用通配符；具有移动文件功能；不能对已打开的文件进行重命名操作

4．ChDrive 语句

格式：ChDrive 驱动器

功能： 改变当前驱动器

说明：如果驱动器为空，则不变；如果驱动器中有多个字符，则只会使用首字母

5．MkDir 语句

格式：MkDir 文件夹名

功能：创建一个新的目录

6．ChDir 语句

格式：ChDir 文件夹名

功能：改变当前目录

说明：改变默认目录，但不改变默认驱动器。

7. RmDir 语句

格式：RmDir 文件夹名

功能：删除一个存在的目录

说明：不能删除一个含有文件的目录

8. CurDir（）函数

格式： CurDir[（驱动器）]

功能：可以确定任何一个驱动器的当前目录。

说明：括号中的驱动器表示需要确定当前目录的驱动器，如果为空，返回当前驱动器的当前目录路径。

8.4 进程条（ProgressBar）和滑块（Slider）

1. 进程条 作用：进程条控件用于监视操作完成的进度。

创建：在窗体上画出进程条控件，右击，选择属性，然后进行所需的外观设置。

主要属性：ProgressBar 控件有一个行程和一个当前位置。行程代表该操作的整个持续时间。当前位置则代表应用程序在完成该操作过程时的进度。Max 和 Min 属性设置了行程的界限。Value 属性则指明了在行程范围内的当前位置。

(1) Min 属性代表进程条全空时的值，缺省时为 0。

(2) Max 属性代表进程条全空时的值，缺省时为 100。

(3) Value 属性代表进程条当前的值（但不出现在属性窗口中），它大于 Min 属性，小于 Max 属性。改变 Value 属性的值将改变进程条的进度显　　示。

示例：

```
Private Sub Command1_Click()
If ProgressBar1.Value < ProgressBar1.Max Then
ProgressBar1.Value = ProgressBar1.Value + 5
Else
ProgressBar1.Visible = False ‘当进程条满了的时候让进程条消失
End If
End Sub
```

2．滑块

Slider 控件是包含滑块和可选择性刻度标记的窗口，在窗体上画出滑块控件，右击，选择属性，然后进行下列设置：

（1）选择滑块的外观

1)方向属性决定滑块的方位，可以是垂直的或水平的滑块。

2)滑块样式属性决定滑块标记的样子。

3)滑块频率属性规定了沿着滑块的标记的间隔大小，缺省的状态是 1，表明每个可能值都出现标记，如果把值设置为 3，则每三个可能值出现一个标记。

（2）滑块的主要属性及事件

1）Min，Max 属性

Min 属性决定滑块最左端或最顶端所代表的值。Max 属性决定滑块最右端或最下端所代表的值。

2）LargeChange,SmallChange 属性

SmallChange 决定在滑块两端的箭头钮上单击时改变的值。LargeChange 决定在滑块上方或下方区域单击时改变的值。

3）Value 属性

Value 属性代表当前滑块所处位置的值，这个值由滑块的相对位置决定。

4）Change 事件

当滑块位置发生变化时就引发了 Change 事件。

8.5 排列显示（ListView）控件和分层显示（TreeView）控件

1．ListView 控件 象“资源管理器”的右侧一样，可使用四种不同视图显示项目。通过此控件，可将项目组成带有或不带有列标头的列，并显示伴随的图标和文本。

可使用 ListView 控件将称作 ListItem 对象的列表条目组织成下列四种不同的视图之一：

大（标准）图标 、 小图标 、 列表 、 报表

View 属性决定在列表中控件使用何种视图显示项目。还可用 LabelWrap 属性控制列表中与项目关联的标签是否可换行显示。另外，还可管理列表中项目的排序方法和选定项目的外观。

ListView 控件包括 ListItem 和 ColumnHeader 对象。ListItem 对象定义 ListView 控件中项目的各种特性，如： 项目的简要描述、由 ImageList 控件提供的与项目一起出现的图标、附加的文本片段，称作子项目，它们与显示在报表视图中的 ListItem 对象关联。

可以使用 HideColumnHeaders 属性决定是否在 ListView 控件中显示列标头。列标头可以在设计时添加，也可以在运行时添加。设计时，使用 ListView“控件属性”对话框的“列首”选项卡添加列标头。运行时，使用 Add 方法添加 ColumnHeader 对象到 ColumnHeaders 集合中。

2．TreeView 控件

该控件象“资源管理器”的左侧一样，用于显示结点（ Node） 对象的分层列表，每个 Node 对象均由一个标签和一个可选的位图组成。TreeView 一般用于显示文档标题、索引入口、磁盘上的文件和目录、或能被有效地分层显示的其它种类信息。

创建了 TreeView 控件之后，可以通过设置属性与调用方法对各 Node 对象进行操作，这些操作包括添加、删除、对齐和其它操作。可以编程展开与折回 Node 对象来显示或隐藏所有子节点。

TreeView 控件使用由 ImageList 属性指定的 ImageList 控件，来存储显示于 Node 对象的位图和图标。任何时刻，TreeView 控件只能使用一个 ImageList。这意味着，当 TreeView 控件的 Style 属性被设置成显示图像的样式时，TreeView 控件中每一项的旁边都有一个同样大小的图像。

8.6 多媒体控件

1. 多媒体控件

Multimedia MCI 控件管理媒体控制接口 (MCI) 设备上的多媒体文件的记录与回放。从概念上说，这种控件就是一组按钮，它被用来向诸如声卡、MIDI 序列发生器、CD-ROM 驱动器、视频 CD 播放器和视频磁带记录器及播放器等设备发出 MCI 命令。MCI 控件还支持 Windows (*.avi) 视频文件的回放。

在允许用户从 Multimedia MCI 控件选取按钮之前，应用程序必须先将 MCI 设备打开，并在 Multimedia MCI 控件上启用适当的按钮。在 Visual Basic 中，应将 MCI Open 命令放到 Form_Load 事件中。

2．媒体播放器控件

可以播放各种多媒体文件，主要属性有：

（1） filename（待播放的文件名），可以在属性窗口中设置，也可以用代码实现。

（2） AutoStart（是否自动播放），默认是 True。

（3） AutoRewind（是否自动循环），默认是 False。

（4） PlayCount（文件播放遍数），默认是 1。

8.7 图象列表（ImageList）和图象组合框（ImageCombo）

1．图象列表

ImageList 控件是包含 ListImage 对象的集合，该集合中的每个对象都可以通过其索引或关键字被引用。ImageList 控件不能独立使用，只是作为一个便于向其它控件提供图象的资料中心。

ImageList 控件的作用象图像的储藏室，同时，它需要第二个控件显示所储存的图像。第二个控件可以是任何能显示图像 Picture 对象的控件，也可以是特别设计的、用于绑定 ImageList 控件的 Windows 通用控件之一。这些控件包括 ListView、ToolBar、TabStrip、Header、ImageCombo、和 TreeView 控件。为了与这些控件一同使用 ImageList，必须通过一个适当的属性将特定的 ImageList 控件绑定到第二个控件。对于 ListView 控件，必须设置其 Icons 和 SmallIcons 属性为 ImageList 控件。对于 TreeView、TabStrip、ImageCombo、和 Toolbar 控件，必须设置 ImageList 属性为 ImageList 控件。

一旦 ImageList 与某个 Windows 通用控件相关联，就可以在过程中用 Index 属性或 Key 属性的值来引用 ListImage 对象。

当与 Windows 通用控件一起使用 ImageList 控件时，在将它绑定到第二个控件之前，按照希望的顺序将全部需要的图像插入到 ImageList。一旦 ImageList 被绑定到第二个控件 ，就不能再删除图像了，并且也不能将图像插入到 ListImages 集合中间。但是可以在集合的末尾添加图像。

2．图象组合框

ImageCombo 控件是标准 Windows 组合框的允许绘图版本。控件列表部分中的每一项都可以有一幅指定的图片。它可以显示一个包含图片的项目列表，每一项可以有自己的图片，也可以对多个列表项使用相同的图片。

除了支持图片之外， ImageCombo 还提供了一个对象和基于集合的列表控件。控件列表部分的每一项是一个不同的 ComboItem 对象，而且列表中的所有项组合起来构成 ComboItems 集合。这就使它容易一项一项地指定诸如标记文本、ToolTip 文本、关键字值以及缩进等级等属性。

8.8 通用对话框控件

Windows 应用程序里的 Open 对话框，Save As 对话框在各个应用程序里看起来都是一样的，通用对话框控件就可以提供这些对话框的标准功能。

1．Open 对话框及 Save As 对话框

打开 Open 对话框使用 ShowOpen 方法，打开 Save As 对话框使用 ShowSave 方法。

```
Private Sub mnuOpen_Click ()
On Error GoTo ErrorHandler
CommonDialog1.CancelError = True
CommonDialog1.Filter = "Text Files (*.txt)|*.txt|Batch Files (*.bat)|*.bat|All Files (*.*)|*.*"
CommonDialog1.ShowOpen ' 显示打开对话框
Call OpenFile(CommonDialog1.FileName)
ErrorHandler:
Exit Sub
End Sub
```

其中第三行决定了在文件格式类型栏里出现的文件类型。第五行需要一个自己的打开文件的过程，这个过程需要的参数就是通用对话框返回的文件名。通用对话框的 CancelError 属性设为 True 的话，用户单击 Cancel 按钮将产生一个错误信息程序，凭借这个信息程序可以检测到用户的放弃操作。

2．Color 对话框

下面的过程可用用户选择的颜色作为窗体的底色。

```
Private Sub mnuColor_Click ()
On Error GoTo CancelButton
CommonDialog1.CancelError = True
CommonDialog1.ShowColor
Form1.BackColor = CommonDialog1.Color
CancelButton:
```

```
Exit Sub
End Sub
```

3．Fonts 对话框

下面的过程可用字体对话框改变文本框的字体:

```
Private Sub mnuFonts_Click ()
On Error GoTo CancelButton
CommonDialog1.CancelError = True
CommonDialog1.Flags = cdlCFBoth ' Flags property must be set to cdlCFBoth
CommonDialog1.ShowFont ' Display Font common dialog box.
Text1.FontName = CommonDialog1.FontName
Text1.FontSize = CommonDialog1.FontSize
Text1.FontBold = CommonDialog1.FontBold
Text1.FontItalic = CommonDialog1.FontItalic
Text1.FontUnderline = CommonDialog1.FontUnderline
Text1.FontStrikethru = CommonDialog1.FontStrikethru
Text1.ForeColor = CommonDialog1.Color
CancelButton:
Exit Sub
End Sub
```

代码的第三行出现了通用对话框的 Flags 属性决定了通用对话框的一些可选项，不过即使不赋值给 Flags，代码也一样会按缺省的情况去执行的。

9.1 数据库的基础知识

1．计算机数据管理技术的发展

第一阶段：人工管理阶段，特点是数据不长期保存，没有软件系统对数据进行管理，没有文件的概念，一组数据对应一个程序。

第二阶段：采用文件管理方式，特点是数据不再是程序的组成部分，而是有组织、有结构地构成文件形式，形成数据文件；文件管理系统是应用程序与数据文件的接口。

第三阶段：数据库管理方式，特点是对所有数据实行统一、集中、独立的管理，数据独立于程序存在，并可以提供给各类不同用户使用。

2．数据库的基本概念

（1） 数据库（DataBase DB）

定义：是以一定的组织形式存放在计算机存储介质上的相互关联的数据的集合。

特点：

具有最小的冗余度
具有数据独立性

实现数据共享

安全可靠，保密性能好

（2） 数据库管理系统（DataBase Management System DBMS）

定义：是操纵和管理数据库的系统软件。

功能：维护数据库、接收和完成用户程序或命令提出的访问数据库的各种请求。

数据语言：数据定义语言（DDL）： 用来建立所需的数据库（即设计库结构）
数据操作语言（DML）： 用来对数据库进行查询和维护操作。
数据控制语言（DCL）： 用来控制数据的访问权限及事务管理。

关系型数据库使用的标准语言是结构化查询语言（Structured Query Language， SQL）。

（3） 数据库系统（DataBase System DBS）

定义； 以数据库应用为基础的计算机系统。

组成：一个完整的数据库系统由数据库、数据库管理系统、数据库管理员和应用程序组成。

数据库也可以这样划分其组成： 硬件：计算机硬件设备
软件：数据库管理系统 、操作系统、开发工具、应用程序
用户：应用程序设计员、终端用户、数据库管理员
类 层次型数据库
网状型数据库
网状型数据库

分代
第一代 非关系型数据库系统，60 年代末问世，包括层次型和网状型
第二代 关系型数据库系统（RDBS），70 年代中期问世
第三代 对象-关系数据库系统（ORDBS 、OOBDS），80 年代中期至今

上述三个概念之间的联系：在数据库系统中通过数据库管理系统来建立和使用数据库。

3．数据模型

数据模型：即描述实体模型的数据。

数据模型的分类： 层次模型（采用树型结构）

网络模型（采用无向图型结构）

关系模型（采用二维表结构）

关系模型的性质：　二维表的记录数随数据的增加而改变，但其字段数是相对固定的

二维表中的每一列均有唯一的字段名

二维表中不允许出现完全相同的两行

二维表中行的顺序、列的顺序均可以任意交换

4.　关系型数据库的基本结构

关系型数据库的基本结构是一张二维表，包括以下概念：

（1） 记录（Record）：数据表中的每一行数据

（2） 字段（Field）：数据表中的每一列，表头（第一行）的内容为字段名

（3） 数据表（Table）：相关数据组成的二维表格

（4） 数据库（Database）：相关数据表的集合

（5） 关系（Relation）：相关表之间通过相关联的字段建立的联系。

（6） 索引（Index）：指按表文件中某个关键字段或表达式建立记录的逻辑顺序。它是由一系列记录号组成的一个列表，目的是提供对数据的快速访问。索引不改变表中记录的物理顺序。

索引关键字（索引表达式）：用来建立索引的一个字段或字段表达式。

数据库和数据表可以通过数据库管理系统软件来建立，如：Access，Visual FoxPro，SQL Server 、Sybase、Oracle。在 VB 环境下可以直接建立 Access 数据库。

关系型数据表的特点：

（1）每一个字段不可再分解，也不能有名字相同的字段；

（2）每一列中的数据都有相同的数据类型；

（3）表中没有内容完全相同的行（记录）。

5．查询的概念

查询（query）：从相关数据表中选取符合特定要求的数据。

创建方法：可以通过查询生成器创建一个查询，也可以在 SQL 窗口直接用 SELECT—SQL 命令写出查询。

9.2 数据库管理器

1．数据库管理器介绍

在 VB 中可以通过“外接程序”菜单中的“可视化数据管理器”调出“VisData”数据库管理器窗口。

菜单选项		功能描述
文件	打开数据库	打开指定的数据库
	新建	根据所选类型建立新数据库
	导入/导出	从其他数据库导入数据表，或导出数据表及 SQL 查询结果
	工作空间	显示注册对话框注册新工作空间，用新输入的用户名和密码从新的工作空间重新打开当前数据库。
	压缩 MDB	压缩指定的 Access 数据库，创建一个加密或解密文件。
	修复 MDB	修复指定的 Access 数据库
实用程序	查询生成器	建立、查看、执行和存储 SQL 查询
	数据窗口设计器	创建数据窗体并将其添加到 VB 工程中
	全局替换	创建 SQL 表达式并更新所选数据表中满足条件的记录
	附加	显示当前 Access 数据库中所有附加数据表及连接条件
	用户组/用户	查看和修改用户组、用户、权限等设置
	System.mda	创建 System.mda 文件，以便为每个文件设置安全机制
	性能选项	设置超时值

2．建立数据库

数据库的基本操作：
创建数据表： 设计表名和表结构
输入记录
建立索引
数据维护： 增加记录
修改记录
删除记录

创建数据表的主要步骤： 设计表结构 → 输入记录 → 建立索引 → 数据维护

（1）设计表结构

启动可视化数据管理器 → 在 database 窗口中右击，从菜单中选择“新表” → 在表结构窗口中输入表名→ 添加字段 → 确定字段名称和属性、有效性规则 → 建立索引 →生成表结构。

“添加字段”对话框各选项说明

选项名	描述
名称	即每个字段的名字（对所用字符没有什么限制）
类型	指该字段的数据特征，包括：Boolean、Byte、Integer、Long、Currency、Single、Double、Date/Time、Text、Binary（二进制型，存放图片）、Memo（备注型，存放长文本）
大小	字段宽度，指该字段所能容纳数据的的最大字节数
固定字段	字段宽度固定不变
可变字段	字段宽度可变
允许零长度	表示空字符串可作为有效的字段值
必要的	表示该字段值不可缺少
顺序位置	字段在表中的顺序位置
验证文本	当向表中输入无效值时系统显示的提示信息
验证规则	验证输入字段值的简单规则，目的是使所输数据符合设定的条件
默认值	在输入时设置的字段初始值，以减少输入重复性数据时的工作量

（2）输入记录

在 database 窗口中选中表名并右击，从菜单中选择“打开”→在数据表窗口输入记录（注意窗口样式）→ 单击“新增”→ 在窗口中输入记录并“更新”→重复……

（3）维护记录

在 database 窗口中选中表名并右击，从菜单中选择“打开”→单击“编辑”/“删除”/“新增”，即可完成对记录的修改、删除、添加操作。

3．建立查询

建立查询就是在数据表中找到符合特定条件的记录并组成一张新表。

在 database 窗口中选中表名并右击，从菜单中选择“新查询”→在查询生成器中构造查询条件→单击“运行”→单击“保存”，给出查询结果文件名→“关闭”。

查询生成器说明

各选项	描述
查询表达式	设置查询应该满足的基本条件，可以用 and /or 来设置应满足的多个条件
表/字段名	设置查询结果中将显示的字段名，单击一个加亮便选中了
前百分之几条	只显示查询结果中的前若干条记录
前 N 条记录	只显示查询结果中的前 N 条记录
“运行”按钮	执行查询命令，并显示出查询结果
“显示”按钮	用消息框显示 SQL 命令
“复制”按钮	将 SQL 命令复制到 SQL 窗口
“保存”按钮	将查询结果取个名并保存到数据库中
“清除”按钮	清除条件列表框中的查询条件
分组条件	将查询结果分组，一般每组只有一个记录
排序条件	将查询结果按顺序显示出来，ASC（升序）、DESC（降序）
连接条件	设置相关表之间的连接字段

9.3 数据控件

1. 数据控件的建立
从 VB 工具箱中单击 Data 控件，在窗体上画出数据控件即可。

2. 数据控件的属性

属性名称	作用
Connect	指定数据控件所要连接的数据库类型，VB 默认的是 Access 的 MDB 数据库，也可以连接 DBF、XLS、ODBC 等数据库
DatabaseName	指定具体使用的数据库文件名，包括路径名
RecordSource	指定具体可访问的数据，这些数据构成记录集对象 Recordset 对象，可以是数据库中的单个表名、一个存储查询，也可以是 SQL 查询命令
RecordsetType	确定记录集类型，有三种： 0——Table（表）； 1——Dynaset（动态，默认的） ；2——Snapshot（快照）
BofAction	当记录指针指向记录集的开始时，确定数据控件该采取的操作： 0——控件重定位到第一个记录 1——移过记录集开始位，定位到一个无效记录，触发数据控件对第一个记录的无效事件 Validate
EofAction	当记录指针指向记录集的结束时，确定数据控件该采取的操作： 0——控件重定位到最后一个记录 1——移过记录集结束位，定位到一个无效记录，触发数据控件对最后一个记录的无效事件 Validate

绑定控件、数据控件、数据库之间的关系：

绑定控件的属性：

DataSource——通过一个有效的数据控件连接到一个数据库上。

DataField——将数据库中的有效字段连接到绑定控件上。

除了常规控件外，可以与 Data 数据控件绑定的控件有：

名称	部件名称	常用属性
MSFlexGrid	Microsoft FlexGrid Control 6.0（OLE DB）	DataSourse
DBCombo	Microsoft Data Bound List Controls 6.0	DataField、DataSource、ListField、RowSource、BoundColumn
DBList		
DBGrid	Microsoft Data Bound Grid Control 5.0（SP3）	DataSourse

3. 数据控件的事件

事件名称	触发时间
Reposition	发生在一条记录成为当前记录后。只要将记录指针从一条记录移动到另一条记录就会触发。
Validate	在一条不同的记录成为当前记录之前，Update 方法之前（用 UpdateRecord 方法保存数据时除外）；以及 Delete、Unload 或 Close 操作之前会发生该事件。它检查被数据控件绑定的控件内的数据是否发生变化。 语法：Private Sub Data_Validate(Action As Integer, Save As Integer) …… End Sub

Validate 事迹的 Action 参数

Action 值	描述	Action 值	描述
0	取消对数据控件的操作	6	Update 操作
1	MoveFirst 方法	7	Delete 方法
2	MovePrevious 方法	8	Find 方法
3	MoveNext 方法	9	设置 Bookmark 属性
4	MoveLast 方法	10	Close 的方法
5	AddNew 方法	11	卸载窗体

4. 数据控件的常用方法

方法名称	作用	示例
Refresh	激活数据控件，使各用户对数据库的操作有效。	Data1.Refresh
UpdateControls	将数据从数据库中重新读到数据控件绑定的控件内，通过它可以终止用户对绑定控件内数据的修改。	放弃修改按钮代码： Data1. UpdateControls
UpdateRecord	强制数据控件将绑定控件内的数据写入到数据库中，不再触发 Validate 事件	确认修改按钮代码： Data1. UpdateRecord

5. 记录集的属性与方法

名称	作用	名称
属性	AbsolutePostion	返回当前指针值，如果是第一条记录，其值为 0；是只读属性
	Bof / Eof	Bof 判断记录指针是否在首记录之前，若是则为 True；Eof 判断记录指针是否在末记录之后，若是则为 True
	Bookmark	用于设置或返回当前指针的标签，可以用在在程序中重定位记录集的指针，其值采用字符串类型。
	NoMatch	在记录集中进行查找时，如果找到相匹配的记录，则为 False，找不到则为 True。
	RecordCount	对 Recordset 对象中的记录记数，为了准确起见，在记数前用 MoveLast 方法将记录指针移到最后一条记录上；是只读属性。
方法	Move	用于移动记录指针，共有 5 种方法： MoveFirst——将指针移到第 1 条记录 MoveLast——将指针移到最后一条记录 MoveNext——将指针移到下一条记录 MovePrevious——将指针移到上一条记录 Move n——将指针向前或向后移动 n 条记录
	Find	在指定的 Dynaset 或 Snapshop 类型的 Recordset 对象中查找与指定条件相符的一条记录，并使之成为当前记录，共有 4 种方法： FindFirst——从记录集的开始查找满足条件的第 1 条记录 FindLast——从记录集的尾部向前查找满足条件的第 1 条记录 FindNext——从当前记录开始查找满足条件的下一条记录 FindPrevious——从当前记录开始查找满足条件的上一条记录 语法格式举例： Data1.Recordset.FindFirst "课程名='计算机基础' " Find 方法支持通配符，默认情况下忽略大小写，可以添加说明改变默认设置： Option Compare Text （与大小写无关） Option Compare Binary （与大小写有关） 如果找不到相匹配的记录，当前记录保持在查找的始发处；如果找到了，则指针定位到该记录。
	Seek	使用该方法必需打开表的索引，它在 Table 表中查找与指定索引规则相符的第一条记录，并使其成为当前记录。语法格式举例： Data1.Recordset.Index= "课程名" Data1.Recordset.Seek "=", "计算机基础"
	Move	用于移动记录指针，共有 5 种方法： MoveFirst——将指针移到第 1 条记录 MoveLast——将指针移到最后一条记录 MoveNext——将指针移到下一条记录 MovePrevious——将指针移到上一条记录 Move n——将指针向前或向后移动 n 条记录

6. 利用数据控件对数据库进行增、删、改操作

操作项目	操作方法	注意事项
增加记录	1） 调用 AddNew 方法：Data1.Recordset. AddNew 2） 给各字段赋值：Recordset.Fields("字段名")=值或在绑定控件中直接输入内容 3） 调用 Update 方法，将缓冲区内的数据写入数据库：Data1.Recordset. Update 4） 调用 MoveLast 方法显示新记录：Data1.Recordset. MoveLast	如果缺少第 3）步而将指针移动到其他记录或关闭了记录，则所做的输入全部丢失；若没有第 4）步。虽然加入了新记录，但记录指针自动返回到添加新记录前的位置上，并不显示新记录。
删除记录	1） 定位被删除记录使之成为当前记录（用 Move 或 Find 方法） 2） 调用 Delete 方法：Data1.Recordset. Delete 3） 调用 MoveNext 方法移动记录指针	使用 Delete 方法时，当前记录立即删除，但被数据库约束的绑定控件仍旧显示该记录的内容，故必须用第 3）步刷新绑定控件。
修改记录	1） 调用 Edit 方法：Data1.Recordset. Edit 2） 给各字段赋值：在绑定控件中直接修改 3）调用 Update 方法，确定所做的修改 Data1.Recordset. Update	如果要放弃对数据的所有修改，可在第 3）步之前用 Refresh 方法，重读数据库，刷新记录。

9.4ADO 数据控件

1． 什么是 ADO?

ADO（ActiveX Data Object）数据访问接口是微软处理数据库信息的最新技术，它是一种 ActiveX 对象，采用了 OLE DB（动态连接与嵌入数据库）的数据访问模式，是数据访问对象 DAO、远程数据对象 RDO 和开放式数据库互连 ODBC 三种方式的扩展。

要使用 ADO 对象必需先为当前工程引用 ADO 对象库，方法是： 执行“工程”菜单中的“引用”命令，在对话框中选中“Microsoft ActiveX Data Object 2.0 Library”。

ADO 对象描述

对象名	描述
Connection	连接数据来源
Command	从数据源获取所需数据的命令信息
Recordset	所获取的一组记录组成的记录集
Error	在访问数据库时，由数据源所返回的错误信息
Parameter	与命令对象相关的参数
Field	包含了记录集中某个字段的信息

2．使用 ADO 数据控件

（1） 添加 ADO 数据控件

从“工程”菜单中选择“部件”命令，在对话框中选中“Microsoft ADO Data ControlS 6.0（OLE DB）”，将其添加到工具箱，并在窗体上拖划出 ADO 数据控件。

（2） ADO 数据控件的基本属性

属性名	作用
ConnectionString	用来与数据库建立连接，它包括 4 个参数： Provide——指定数据源的名称 FileName——指定数据源所对应的文件名 RemoteProvide——在远程数据服务器打开一个客户端时所用的数据源名称 RemoteServer——在远程数据服务器打开一个主机端时所用的数据源名称
RecordSource	确定具体可访问的数据，可以是数据库中的单个表名、一个存储查询或一个 SQL 查询字符串
ConnectionTimeout	设置数据连接的超时时间，若在指定时间内连接不成功则显示超时信息
MaxRecords	确定从一个查询中最多能返回的记录数

（3） ADO 数据控件的属性设置

1）先在窗体上放置一个 ADO 数据控件

2）在 ADO 属性窗口中单击 ConnectionString 属性右边的…按钮，从对话框中选择连接数据源的方式：

使用连接字符串——单击“生成”按钮，通过选项设置系统自动产生连接字符串

使用 Data Link 文件——通过一个连接文件来完成

使用 ODBC 数据资源名称——在下拉列表中选择某个创建好的数据源名称作为数据来源对远程数据库进行控制。

3）在 ADO 属性窗口中单击 RecordSource 属性右边的…按钮，在“命令类型”中选择 2——adCmdTable，在“表或存储过程名称”中选择所需要的表。

以上 2）、3）可以合并成一步：在 ADO 控件上单击右键，从快捷菜单中选择 ADODC 属性，直接在属性页对话框中进行所有设置。

（4） ADO 数据控件的方法和事件

与 Data 数据控件完全相同。

（5） 在 ADO 上新增绑定控件

可以从“工程”的“部件”中添加如下绑定控件：

控件名称	部件名称	常用属性
DataGrid	Microsoft DataGrid Control 6.0（OLE DB）	DataSource
DataCombo	Microsoft DataList Controls 6.0（OLE DB）	DataField、DataSource、ListField、RowSource、BoundColumn
DataList		
MSChart	Microsoft Chart Control 6.0（OLE DB）	DataSource

3. 使用数据窗体向导

从“外接程序”菜单中选择“外接程序管理器”，在对话框中选中 VB6 数据窗体向导，“加载”并“确定”，再从“外接程序”菜单中选择“数据窗体向导”，然后根据系统提示逐步操作即可创建所需要的数据窗体，系统自动把所创建的窗体加到工程中。

9.5 结构化查询语言

1．SQL 的基本组成

SQL 语言由命令、子句、运算、函数等组成：

（1） SQL 命令

命令	功能
CREATE	用于建立新的数据表结构
DROP	用于删除数据库中的数据表及其索引
ALTER	用于修改数据表结构
SELECT	用于查找符合特定条件的某些记录
INSERT	用于向数据表中加入数据
UPDATE	用于更新特定记录或字段的数据
DELETE	用于删除记录

（2） SQL 子句

子句	功能
FROM	用于指定数据所在的数据表
WHERE	用于指定数据需要满足的条件
GROUP BY	将选定的记录分组
HAVING	用于说明每个群组需要满足的条件
ORDER BY	用于确定排序依据
INTO	查询结果去向

（3） SQL 运算符

逻辑运算符	And（与）、 Or（或）、 Not（非）
比较运算符	< <= > >= = <>

（4） SQL 函数

AVG	COUNT	SUM	MAX	MIN
求平均值	计数	求和	求最大值	求最小值

2. SQL 语句的应用

语句功能	语法格式
建立数据表	CREATE TABLE 数据表名（字段名 1 数据类型（长度）， 字段名 2 数据类型（长度），……） 举例：create table student（xh text（9）， xm text（8）， cj single（4），nl integer（2）） 建立含有 xh、xm、cj、nl 4 个字段的 student 表
添加字段	ALTER TABLE 数据表名 ADD COLUMN 字段名 数据类型（长度） 举例：alter table student add column xb text（2） '在学生表中添加性别字段
删除字段	ALTER TABLE 数据表名 DROP COLUMN 字段名 举例： alter table student drop column nl '将学生表中的年龄字段删除
数据查询	SELECT 字段名表 FROM 子句 WHERE 子句 GROUP BY 子句 HAVING 子句 ORDER BY 子句 INTO 子句 举例：select xh, xm from student where xb="男" order by xh 从学生表中查询性别为男的学生，显示其学号和姓名并使结果按学号升序排列。
添加记录	INSERT INTO 数据表名（字段名 1，字段名 2……） VALUES（数据 1，数据 2……） 举例：insert into student (xh, xm, xb) values("015200101", "王小二", "男")
删除记录	DELETE FROM 数据表名 WHERE 条件表达式 举例：delete from student where xb="男"
更新记录	UPDATE 数据表名 SET 新数据值 WHERE 条件表达式 举例：update student set cj=cj+5 where xb="女"

9.6 报表制作

1.报表的概念

利用报表可以把数据表中的数据按一定的格式输出到屏幕上或打印到纸上。

2.制作报表的方法

在 VB6.0 中可以利用报表设计器来制作报表，从“工程”中选择“添加 data report”，将报表设计器加入到当前工程中，报表由 5 部分组成：

报表标头——每份报表只有一个，可以用标签建立报表名。

页标头——每页有一个，即每页的表头，如字段名。

细节——需要输出的具体数据，一行一条记录。

页脚注——每页有一个，如页码。

报表脚注——每份报表只有一个，可以用标签建立对本报表的注释、说明。

使用报表设计器处理的数据需要利用数据环境设计器创建与数据库的连接， 从“工程”菜单中选择 “添加 Data Enviroment”，在连接中选择指定的数据库文件，完成与数据库的连接，然后产生 Command 对象连接数据库内的表。

制作报表的步骤：

（1） 新建工程，在窗体上放置两个命令按钮；

（2） 从“工程”菜单中“添加 Data Enviroment”，右击 Connection1，在属性中选择“Microsoft Jet 4 OLE DB Provider”，在“连接”中指定数据库；

（3） 再次右击 Connection1，选则“添加命令”，创建 Command1 对象，右击 Command1，在属性中设置该对象连接的数据源为需要打印的数据表；

（4） 在从“工程”菜单中“添加 Data Report”，在属性窗口中设置 DataSource 为数据环境 DataEnviroment1 对象，DataMember 为 Command1 对象，即指定数据报表设计器 DataReport1 的数据来源；

（5） 将数据环境设计器中 Command1 对象内的字段拖到数据报表设计器的细节区；

（6） 利用标签控件在报表标头区插入报表名，在页标头区设置报表每一页顶部的标题；

（7） 利用线条控件在报表内加入直线，利用图形控件和形状控件加入图案或图形；

（8） 利用 DataReport1 对象的 Show 方法显示报表，在窗体 Click 事件加代码：DataReport1.Show；

（9） 利用预览窗口按打印按钮可以打印报表；

（10）利用预览窗口工具栏上的导出按钮可以将报表内容输出成文本文件或 Html 文件；也可以利用 DataReport1 对象的 ExportReport 方法将报表内容输出成文本文件或 Html 文件。

制作报表的简单方法是从“外接程序”中选择报表向导来设计报表。