

高等学校教材

计算机在分析化学中的应用

— Visual Basic 6.0 程序设计实现

冉国芳 编著



化学工业出版社
教材出版中心

高等院校教材

计算机在分析化学中的应用

——Visual Basic 6.0 程序设计与实例

张德强 主编



化学工业出版社

高等学校教材

计算机在分析化学中的应用

——Visual Basic 6.0 程序设计实现

冉国芳 编著



化学工业出版社
教材出版中心

· 北 京 ·

前 言

作者多年来主讲应用化学及相关专业课程《计算机在分析化学中的应用》，在该课程的教学实践中，作者将分析化学、数学、统计学及计算机科学相结合，不断充实修改讲义，最终完成了本书的编写工作。本书介绍有关计算机在分析化学中的应用知识，采用 Visual Basic 程序设计方法编程处理分析化学中的纷繁数据以获得更多有用的化学信息。

本书的主要内容有：

1. Visual Basic 程序设计上机指导；
2. 分析数据的常规统计分析；
3. 方差分析；
4. 回归分析；
5. 解高次方程及线性方程；
6. 计算机图解分析；
7. 曲线拟合；
8. 模式识别；
9. 优化设计；
10. 计算机辅助分析化学教学。

书中所有程序均由作者编写调试，并经文献示例验证通过，而且许多程序就是从作者所开发的相关软件中挑选出来的源代码，适用性强，可移植性好。有些程序段，请读者在移植或程序设计时加入错误陷阱处理。本书不仅适用于应用化学专业的计算机应用教学，而且可供其它专业如化学、药学、矿物学等相关专业以及学习 Visual Basic 6.0 算法及其程序设计的相关学科工作者参考。为提高化学学科的理论及实验教学水平，以及满足用户处理化学数据的需要，作者不久将出版相关软件。

书中得以出版，徐斌、冯碧元给予了热情的指导和帮助；贾先君负责了书中的图表及文字处理工作；在编写过程中得到了李晓芸、杨晓红、徐贤英以及校院其它同仁的支持；国内外各方面专家的著作和报告也给予了作者有益养分，在此对他们表示真诚感谢。

本书在出版过程中，得到了化学工业出版社的支持与帮助，他们严谨的工作作风和高度的敬业精神保证了本书的质量，谨此表示衷心感谢。

限于编著水平，书中一定存在很多不足，恳请得到各方面专家的批评指正。

冉国芳

E-mail: rangf@163.com

2006-6-30

目 录

第 1 章 Visual Basic 6.0 语言简介	1
1.1 Visual Basic 6.0 的用户环境	1
1.2 Visual Basic 6.0 的常用术语	2
1.2.1 对象	2
1.2.2 属性	2
1.2.3 事件	3
1.2.4 过程	4
1.2.5 方法	4
1.3 变量、运算符及运算规则	5
1.3.1 数据类型	5
1.3.2 常量	5
1.3.3 变量	6
1.3.4 运算规则	8
1.4 函数	9
1.4.1 数字函数	10
1.4.2 字符串函数	10
1.5 输入输出语句	11
1.5.1 窗体设计	11
1.5.2 生成控件	15
1.5.3 命令按钮 (CommandButton)	17
1.5.4 文本框 (TextBox)	18
1.5.5 标签	19
1.5.6 框架	19
1.5.7 列表框	19
1.5.8 组合框	20
1.5.9 消息框	20
1.5.10 InputBox() 函数	20
1.5.11 Print 语句	21

1.5.12 菜单设计	21
1.6 几种常用的控制结构	22
1.6.1 分支结构 (If-Then-Else 结构, 条件语句)	22
1.6.2 选择结构 (也称情况语句)	23
1.6.3 循环结构	24
1.7 过程及调用	26
1.7.1 过程结构	26
1.7.2 过程的调用	26
1.8 自定义函数及调用	27
1.8.1 结构	27
1.8.2 说明	27
1.8.3 调用	27
1.9 Visual Basic 6.0 数据库编程技术	27
1.9.1 ADO 简介	27
1.9.2 设置 ADO Data 控件的 ConnectionString 属性和 RecordSource 属性	28
1.9.3 通过常用的结构化查询语句 (SQL) 操作数据库	28
1.10 Visual Basic 的绘图操作	29
1.10.1 定制坐标 (对窗体, 图片框)	29
1.10.2 颜色	30
1.10.3 像素 (画点) 控制	30
1.10.4 画线	30
1.11 标准 (代码) 模块	30
1.12 程序调试与出错处理	33
1.13 保存用户的工作	34
1.14 创建 Windows 应用程序	34
第 2 章 数据统计分析初步	35
2.1 实验数据的输入	35
2.1.1 通过 InputBox() 函数输入数据	35
2.1.2 通过文本框输入数据	35
2.1.3 通过数据库输入数据	38
2.2 累加法、极大值和极小值、排序	41
2.2.1 累加	41
2.2.2 排序、极大值和极小值	42

2.3 分析化学中的数据处理	44
2.3.1 分析化学中的误差	44
2.3.2 少量数据的统计处理	47
2.3.3 异常值的取舍	55
2.4 方差分析	61
2.4.1 单因素方差分析法	61
2.4.2 双因素方差分析法	66
2.4.3 三因素方差分析法	74
第3章 解方程	84
3.1 解高次方程	84
3.1.1 二分法	84
3.1.2 黄金分割优选法	86
3.1.3 弦割法	87
3.1.4 牛顿迭代法	89
3.1.5 扫描法	90
3.1.6 酸碱溶液 pH 值的计算	94
3.2 解线性方程	108
3.2.1 求解线性方程组的通用模块	108
3.2.2 吸光光度法处理混合组分	111
第4章 数值积分	114
4.1 定步长辛普生积分	114
4.2 变步长辛普生积分	114
4.3 计算随机误差的区间概率	115
第5章 插值	118
5.1 一元二点插值	118
5.2 拉格朗日插值	121
5.3 一元三点插值	122
5.4 三次样条函数插值	125
第6章 回归分析	130
6.1 一元线性回归分析法	130
6.1.1 一元线性回归的数学模拟	130

6.1.2	回归效果的判断	132
6.1.3	标准差和置信界限	133
6.2	非线性回归分析	136
6.3	多元线性回归分析	139
6.3.1	多元线性回归方程的建立	139
6.3.2	回归方程的检验	140
6.4	逐步回归分析法	144
6.4.1	逐步回归分析方程的建立	145
6.4.2	逐步回归分析方程的例解与程序设计	148
第 7 章	模式识别简介	156
7.1	数据的预处理	156
7.2	特征选择	157
7.3	模式识别方法	158
7.3.1	无监督模式识别——聚类分析	158
7.3.2	有监督模式识别	160
7.4	人工智能与专家系统简介	160
第 8 章	优化设计	162
8.1	单纯形序贯优化法	162
8.1.1	定步长单纯形法	163
8.1.2	变步长单纯形法	164
8.2	遗传算法	172
第 9 章	计算机图解分析	173
9.1	多元酸(碱)的分布系数	173
9.1.1	多元酸(碱)的分布系数的数学模型	173
9.1.2	多元酸(碱)的分布系数的程序设计	173
9.2	络合滴定中的酸效应系数及酸效应曲线	179
9.2.1	数学模型	179
9.2.2	程序设计	180
9.3	随机误差正态分布曲线	185
9.3.1	数学模型	185
9.3.2	程序设计	185
9.4	滴定曲线与滴定误差	187

9.4.1	酸碱滴定曲线	187
9.4.2	酸碱滴定误差曲线	188
9.4.3	程序	188
9.5	二次微商法确定电位滴定终点	195
9.5.1	数学模型	195
9.5.2	程序设计	196
9.6	曲线拟合	200
9.6.1	直线拟合	200
9.6.2	最小二乘法拟合曲线	203
9.6.3	三次样条函数拟合曲线	206
9.6.4	数据平滑	209
第 10 章	计算机辅助分析化学教学	215
10.1	概述	215
10.2	计算机辅助分析化学理论教学	215
10.3	计算机辅助分析化学实验教学	220
10.4	实验教学管理软件的开发	222
10.4.1	软件功能模块	222
10.4.2	主要功能介绍	223
10.5	数据库支持及表之间的关系	225
参考文献		226

第 1 章 Visual Basic 6.0 语言简介

1.1 Visual Basic 6.0 的用户环境

Visual Basic 6.0 是可视化的面向对象的开发 Windows 应用软件的工具。它被誉为“惊世骇俗”和“令人震惊的奇迹”。

安装 Visual Basic 6.0 的最低计算机配置为：

- ① Microsoft Windows 95/98/Me/NT 3.51 或更高版本；
- ② Internet Explore 4.01 以上版本；
- ③ 80486 或以上 CPU 处理器；
- ④ 一个 CD-ROM 驱动器；

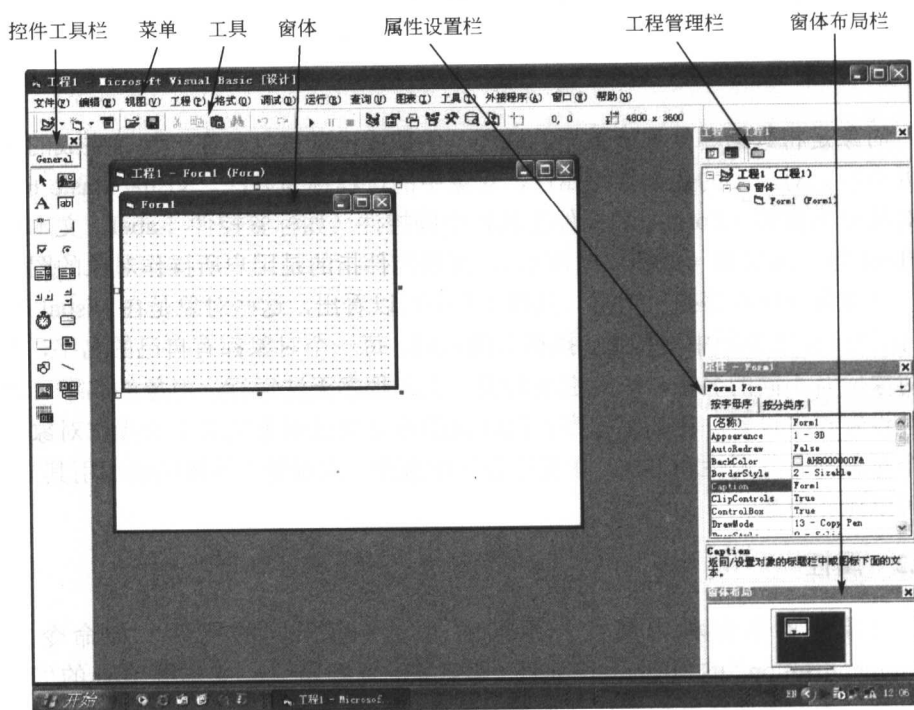


图 1-1 Visual Basic 6.0 的用户开发界面

- ⑤ Windows 支持的 VGA 640×480 或更高分辨率的显示器；
- ⑥ 16M RAM；
- ⑦ 鼠标。

安装 Visual Basic 6.0 之前先安装 Visual Studio 6.0，按安装程序的提示即可完成安装。

安装完毕后运行 Visual Basic 6.0 最简单的办法是点击桌面上左下角的“开始”按钮，在弹出的菜单选项中，选择“程序”命令，出现“程序”的级联菜单，单击级联菜单中“Microsoft Visual S Studio 6.0”命令，出现下级菜单，然后单击下级菜单中的“Microsoft Visual Basic 6.0”，则可运行“Visual Basic 6.0”了（如图 1-1 所示）。

Visual Basic 6.0 的用户开发界面如图 1-1 所示。它允许用户将菜单、文本框、命令按钮、复选框、列表框、滚动条以及文件目录框等增加到窗口中；用户可以使用网格来处理表格数据；可方便地与 Windows 程序通信，以及访问数据库等许多优点。本书将按 Visual Basic 6.0 语言的语法规约定给出程序和说明。

1.2 Visual Basic 6.0 的常用术语

1.2.1 对象

对象是将数据和处理该数据的过程、函数或子过程打包在一起而生成的新的数据类型。对象中的数据称为属性，对象中的过程称为方法。Visual Basic 6.0 中的对象分为窗体（Form）和控件工具栏中的控件（如标签控件 Label、文本框控件 Text 等）两大类（如图 1-1 所示），所谓控件指的是用户所操作对象的图形表示，以便向应用程序提供信息。从图 1-1 中可以看出，这些对象是在 Visual Basic 6.0 的程序开发界面中看得见、摸得着的东西。每一个对象都有自己的名字，不同的对象应有不同的名字，以免造成混淆，引起程序无法运行。对象的名字可以在属性设置栏中设置。在编写程序代码时我们就是通过对象的名字来操作对象的。每个对象都有其属性和方法，在程序设计中改变（或设置）其属性或调用其方法，均可以操作对象。

1.2.2 属性

对象中的数据称为属性，不同的对象具有不同的属性。如命令按钮“CommandButton”的“Caption”属性（即对象的标题属性）、文本框 Text 的“Text”属性等。我们可在属性设置栏（如图 1-1 所示）中通过改变控件的属性来控制控件在程序中的作用，当然也可通过编写程序代码给对象的属性赋值来控制控件在

程序中的作用，其方法是：在对象名的后面紧跟一个英文点号，然后书写其属性名，并把它们放在等号左边，把所要赋予的值放在右边。如给命令按钮“CommandButton”（假设其对象名为“Command1”）的“Caption”属性赋值为“确定”的语句为：

```
Command1.Caption="确定"
```

用户可以根据自己的需要任意通过代码来改变属性设置。

1.2.3 事件

Visual Basic 6.0 是面向对象的、事件驱动的程序设计语言。所谓事件指的是响应对象行为的动作。注意：只有“1.2.1”节中所指出的窗体“Form”和各控件对象才有事件。Visual Basic 6.0 中各对象最常用的事件有诸如鼠标单击事件“Click”、鼠标双击事件“Double_Click”等事件。对象不同，事件各不一样，如时钟控件“Timer”对象的事件只有“Timer”事件。在事件驱动的应用程序中，代码不是按预先编制好的路径运行，而是在响应不同的事件时执行特定的代码片段。每一个对象几乎都有自己的多个事件。在没有事件发生时，整个程序处于停滞状态。

要针对某一个对象编程，可以在如图 1-1 中的用户开发环境中，在“工程管理栏”（见图 1-2），单击“查看对象”按钮可显示当前窗体上的所有的对象；单击“查看代码”按钮可打开代码窗口，如图 1-3 所示，这也是我们书写程序的窗口。在“对象”下拉列表中可查看当前窗体上的所有对象名列表，在“对象事件过程”下拉列表中可查看该对象的所有事件。点击某一事件，在窗口中就会自然弹出关于这一事件过程的程序编写框架，如窗体的加载事件过程（有关事件过程

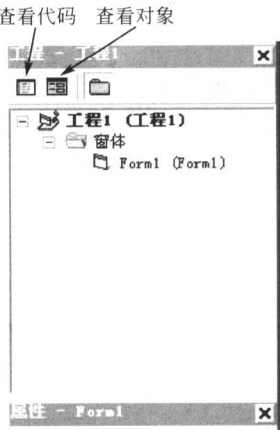


图 1-2 工程管理栏



图 1-3 对象事件代码窗口

和通用过程的语法及过程的调用问题请参见下一节和 1.6 节)：

```
Private Sub Form_Load()
```

```
[书写程序代码]
```

```
End Sub
```

用户就是在以上这样一个程序书写的模板内书写各对象事件过程或通用过程的代码的。

1.2.4 过程

能执行特定任务的代码段叫做过程。每一过程都有表示该过程的名字，即过程名。

过程的表示方法有两种。

(1) 事件过程

事件过程的名称为“对象名_事件”，事件过程的语法如下：

```
Private Sub 对象名_事件 ([参数列表])
```

```
[事件过程的内容]
```

```
End Sub
```

事件过程是 Visual Basic 6.0 编程的核心，事件过程不应过于复杂，为便于调试，最好不要长于一页甚至长于一屏，如果是这样的话就应当考虑用通用过程来做工作，尤其是要将多次调用的程序代码写在一个通用过程中。

(2) 通用过程

其语法为：

```
Sub 过程名([参数列表])
```

```
[事件过程的内容]
```

```
End Sub
```

在 Sub 前可加“Private”或“Public”关键词。“Private”指定该过程只有在声明它的模块中才能调用，“Public”指该过程可以被整个应用程序的其它过程调用。凡过程均应以“Sub 过程名(参数列表)”开头，以 End Sub 结尾。

过程的调用见 1.7 节。

1.2.5 方法

对象的属性实际上是对象的数据。对象本身所包含的对属性操作的函数或过程叫方法(Method)。方法的内容是不可见的，用户只要会使用它就行了。使用“方法”的语法如下：

对象名. 方法

例如，显示或隐藏窗体 Form1（见图 1-3）在程序代码中可写为：

Form1.Show '显示窗体 Form1

Form1.Hide '隐藏窗体 Form1

在上述程序代码中，“'”号是 Visual Basic 6.0 中注释标记符号，其后面的内容在程序运行时不会被编译执行。

在实际编程中，在代码窗口，用户只要将存在的对象名写出来，并在其后加写一英文句点，紧跟着就会自动显示关于这一对象的所有方法或属性列表，用户可以从这一方法列表中选择所需要的方法。

1.3 变量、运算符及运算规则

1.3.1 数据类型

Visual Basic 6.0 中，数据不仅包括一些数值信息，而且包括一些如姓名、年龄、日期、婚否等非数字的信息，Visual Basic 6.0 将数据分为多种不同的类型，如表 1-1 所示。

表 1-1 数据的基本类型

数据类型	类型声明字符	占字节数	有效值
Integer (整型)	%	2	-32768~32767
Long (长整型)	&	4	-217483648~ 217483647
Single (单精度型)	!	4	-3.402823e38~ 3.402823e38
Double (双单精度型)	#	8	-1.7976931348623232d308~ 1.7976931348623232d308
Currency (货币型)	@	8	-922337203685477.5808~ 922337203685477.5807
String (字符型)	\$	一个字符占一个字节	0~65535
Boolean (布尔型)		2	True 或 False
Date (日期型)		8	100.1.1~9999.12.31
Byte		1	0~255
Variant (变体型)		不定	以上任一种
Object		4	

需要指出的是，“Variant”数据类型的运算速度比其它数据类型要慢许多，如果能够确定变量的数据类型，则最好不要用“Variant”数据类型，以使程序运行效果最好。

1.3.2 常量

在程序运行过程中始终固定不变的量就是常量。

用户声明常量的语法：

[Public/Private] Const 常量名 [As 数据类型]=表达式

如:

Const Pi=3.1415926535897932384626433835795028842

[Public/Private]为可选关键词。在后续的程序说明中,我们用“[]”表示在编程中可选也可不选的某一内容。“Public”的作用域为全局,“Private”的作用域为当前模块。关于作用域可参见“变量的作用域”一节。

1.3.3 变量

在程序中可以发生变化的量称为变量,在一个变量中可存放一个数据或一组数据。变量可分为简单变量和数组变量两种(见表1-2)。Visual Basic 6.0的变量掌握了信息(值),无论何时使用一个变量,Visual Basic 6.0都在计算机内存中为它开辟了一个区域来存放它的信息。

表 1-2 变量类型

变 量 类 型	数 值 变 量	字 符 变 量
简单变量	X, Y, pH, Concentration	A, NAME
下标(数组)变量	Ka(6), X(5), Q(3, 10)	INDICATOR (18)

(1) 变量名

PC机变量名由英文字母、阿拉伯数字和下划线组成。变量名的第一个字符必须是字母,Visual Basic 6.0的变量名的长度允许长达255个字符,由相同字符组成的变量名无论是大写还是小写均为同一个变量。保留字(包括关键字、标准函数名、标准子程序名)不能作为变量名。为便于程序的阅读、调试及修改,变量除须合法外,最好应具有提示性,如用Concentration表示浓度,用pH表示溶液的酸碱度等。

(2) 变量类型

可通过Dim关键字将变量声明为数值型变量,如单精度型(Single)、双单精度型(Double)、整型(Integer)、长整型(Long)和字符型变量(String)等(见表1-1)。

(3) 变量作用域

在代码窗口的顶部或在添加的模块中用“Dim”声明的变量是模块级的,对整个模块都有效;在某一过程内用“Dim”声明的变量只在这一过程内有效。

用Public关键字声明的变量是程序级的,在整个程序中起作用,即使声明语句在过程内也一样。

用Static关键字声明的变量是静态变量,即使到过程结束也一直保存着最近一次对该变量所赋的值。Visual Basic在任何时候都不会去初始化静态变量。

(4) 变量的声明

变量在使用之前必须在程序顶端用声明方式为变量指定一种数据类型，即先声明，后使用，以把变量通知给程序。

在代码窗口的顶端“通用”模块中，如果开始便使用了“Option Explicit”关键词，则在该模块的程序编写中，所有的变量都须先声明再使用，不经声明的变量在程序的运行中就会出现错误。采用这一做法，可防范在程序的编写中由于变量的滥用而引起的混乱。

(5) 变量声明的方法

隐性声明：不加说明，以默认方式直接使用。

显性声明：

Dim 变量名 [as 类型]

说明：[as 类型] 子句是可选项，如果省略掉，则该变量被声明为 Variant 类型。用 Public 和 Static 关键字声明全局变量和静态变量。

凡变量在程序运行中未被赋值，则数字变量均为“0”，字符变量均为空“ ”。

(6) 数组

定义：用相同名字保存的一系列数据的集合。

下界：数组中的第一个元素。

上界：数组中的最后一个元素。

声明数组：数组声明与变量声明基本一致，只是增加了一个指定数组大小的参数

Dim 数组名 ([下标] To 上标, {[下标] To 上标,...}) [As 数据类型]

如果是单下标的数组称为一维数组，如果是双下标甚至三下标的数组则称为二维数组和多维数组。默认数组的第一个元素的下标为 0。

动态数组：在程序运行时决定数组的大小。其具有灵活多变的特点。

建立动态数组的步骤：

① 用“Dim”声明空的动态数组，如：Dim x() As Single。

② 用“Redim”语句来配置数组，如：ReDim x(8) As Single。用“Redim”所配置的数组将使字符数组各元素全部赋值为“空”，而使数字数组的各元素全部赋值为“0”。

③ 用“Redim Preserve”语句来配置数组，如：ReDim Preserve x(8) As Single。用“Redim Preserve”所配置的数组将会保留数组已赋的值，但数组下标若小于原数组下标，则会截断原数组使之符合用“Redim Preserve”语句来配置数组的大小。

④ 在程序运行时可用 Ubound(x)和 Lbound(x)函数返回数组的上下界。如将数组元素的个数赋值给变量 n 的语句可写为：