

普通高中课程标准实验教科书

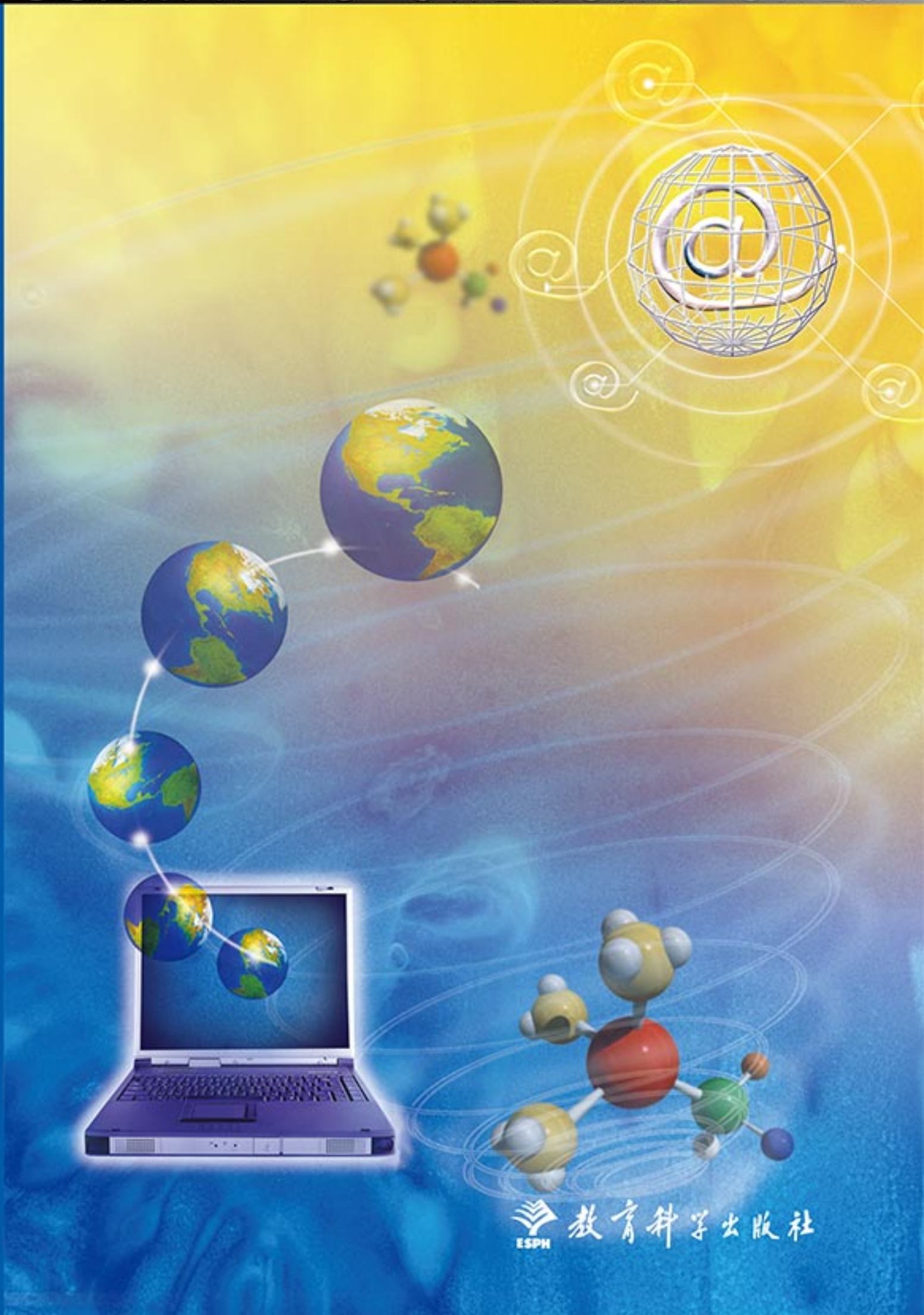


算法与程序设计

(选修)

SUANFA YU CHENGXU SHEJI

SUANFA YU CHENGXU SHEJI



教育科学出版社

经全国中小学教材审定委员会 2004 年初审通过
普通高中课程标准实验教科书

算法与程序设计

SUANFA YU CHENGXU SHEJI

(选修)

教育科学出版社

· 北 京 ·

The background features a light blue and green gradient with a pattern of binary code (0s and 1s) in various colors (green, blue, yellow) scattered across the top and right. On the right side, there is a small, stylized illustration of a computer component, possibly a hard drive or a power supply, with a yellow and orange casing.

前 言

同学们，也许你对程序设计有浓厚的兴趣，想了解一下程序设计到底是怎么回事；也许你梦想将来能成为一名程序设计大师。不管怎样，都祝贺你选修了“算法与程序设计”，因为你已经开始向更高层次的信息技术迈进了！

当你学习信息技术并试图用计算机解决问题的时候，是否感叹过计算机的无穷魅力？是否想过为什么计算机能按照我们的意图一步一步完成复杂的任务呢？“算法与程序设计”这门课将帮助我们揭开计算机神秘的面纱。通过学习，我们不仅能知道平时使用的软件是如何开发出来的，还能从计算机语言层面上了解计算机能做什么，是如何工作的，知道软件背后的设计过程，甚至我们自己也能设计出优秀的计算机软件。同时，学完这门课后，对我们分析问题、解决问题的能力也会有很大的帮助。

如何才能学好“算法与程序设计”呢？“兴趣是最好的老师，爱好是成功之母。”这门课学起来虽然有一定的困难，但也非常有意思，学好了我们就可以“指挥”计算机了！只要你保持对程序设计的兴趣，多读多练，在学习的过程中多和同学们交流，就一定能学好程序设计！

本书选用 Visual Basic 6.0(以下简称 VB 6.0)作为程序设计语言，是因为它简单易学，基本能反映当今程序设计的思想和方法。本书要求的学习环境为 VB 6.0 学习版，但为了扩展你的知识面，书中也涉及了少量专业版功能，对于超出学习版功能的内容，我们可以根据学校的条件选择学习。

在高中阶段，学习程序设计毕竟不同于专业训练，并且程序设计语言的变化也非常快，因此，我们尽可能从程序设计的基本知识与一般规律出发组织编写本教科书。另外，为了兼顾同学们的兴趣和爱好，我们对教科书内容

进行了适当地扩展，这就是你在教科书中看到的带粉色背景颜色的内容与页边栏目内容。我们精心选取了能贴近大家的生活与经验、能激发同学们学习兴趣的案例，同时也兼顾了案例的典型性、连贯性、算法的多样性、地区差异性原则。

在教科书中，你将遇到两种不同类型的实践。一类是贯穿于正文当中的实践，目的是帮助你加深对正文的理解和获得解决问题的能力；另一类是位于每章最后的综合实践，可以帮助你综合地掌握本章内容。在学习正文时，你要特别注意与实践相配合，在做中学，在学中做。有些实践必须完成，有些可以在教师的指导下选做。对于部分难度较大的实践，光盘中提供了程序源代码，同学们在学习时可以直接调用，也可以自行分析，重新编写。

第五章讲述了一个小软件的完整开发过程，同学们可根据具体情况或者自主开发，或者将配套光盘中的模块重新进行组装，从中体验软件开发的过程；也可以自选主题进行分析设计。第六章是从VB 6.0的完整性考虑而设置的，作了“*”标记，属于选修内容，同学们可根据自己的能力与兴趣自主探究，或者在教师的帮助下有选择地进行学习。

“世上无难事，只怕有心人！”我们殷切希望你能通过这门课的学习奠定程序设计的良好基础，在不久的将来为加速我国软件产业的发展作出自己的贡献！



目 录

第一章 如何用计算机解决问题	1
1.1 计算机解决问题的过程	2
1.2 算法描述与设计	5
1.3 初试身手，体验编程	9
第二章 程序设计 ABC	13
2.1 可视化程序开发工具	14
2.2 数据的表示与处理	20
2.3 程序的基本结构	32
第三章 算法的程序实现	41
3.1 用解析法解决问题	42
3.2 用穷举法解决问题	45
3.3 在数组中查找数据	49
3.4 对数据进行排序	57
3.5 用递归法解决问题	65

第四章 程序设计思想和方法 71

4.1 结构化程序设计 72
4.2 面向对象程序设计 75

第五章 软件是如何开发的 84

5.1 任务分析与系统设计 85
5.2 代码编写与调试 90
5.3 软件测试与维护 100




* 第六章 综合实践 104

6.1 开发数据库软件 105
6.2 开发多媒体软件 112
6.3 开发网络应用软件 116

附录 124

附录一 常用字符与 ASCII 代码对照表 124
附录二 VB 常用函数 125
附录三 VB 常用控件 127
附录四 中英文术语对照表 133

第一章 如何用计算机解决问题

-  计算机解决问题的过程
-  算法描述与设计
-  初试身手，体验编程

本章我们将带你探究计算机解决问题的过程，在你已有的数学知识基础上，进一步了解算法，学会用不同的方法描述算法；然后，通过使用 Visual Basic 编程工具编写一段简单的程序，让你小试身手，初步体验程序设计的过程，感受一下什么是源程序、计算机如何执行源程序等。



1.1 计算机解决问题的过程

本节将给出一个有趣的问题，请同学们试着找出不同的求解方法。在求解的过程中，请留意你是如何思考问题，又是如何解决问题的。通过本节的学习，你可以：

- 知道人是如何分析问题、解决问题的
- 了解计算机求解问题的过程
- 知道人解决问题和计算机解决问题的异同

运行光盘中的VB程序“桌面小时钟问题”，如图1-1所示，在桌面的右上角会显示数字形式的小时钟。单击显示区域，小时钟停止变化；再次单击显示区域，小时钟又开始变化；双击显示区域，可结束程序。

时间显示



图 1-1 桌面小时钟问题

该程序具有以下几个特点。

- (1) 界面设计。运行该程序时，会在桌面右上角显示不断变化的数字，但它不像我们平常使用的其他程序那样具有窗口。
- (2) 数据同步。小时钟数字的变化始终和计算机的系统时间一致。
- (3) 状态切换。运行该程序时，单击显示区域，小时钟停止变化；再次单击显示区域，小时钟又开始变化。
- (4) 程序控制。双击显示区域，可结束程序。

人类在解决问题的时候，一般按照下列思路进行：首先观察、分析问题，收集必要的信息，然后根据已有的知识、经验进行判断、推理，尝试按一定的方法和步骤去解决问题。

对于同一个问题,我们可能会有多种求解方法和步骤。不同的方法有优劣之分,有的方法需要的步骤较少,而有的则需要的步骤较多。当然,并不是说步骤较多的方法一定不如步骤较少的方法好,要结合具体情况评价一种方法的优劣。

那么计算机又是如何解决问题的呢?

要让计算机解决问题,首先我们需要通过分析找出求解问题的方法和步骤,然后将待处理的信息用计算机能识别的方式表示,使之成为计算机能处理的数据。最后,用程序设计语言设计出问题的求解过程,计算机就会按照我们给定的指令一条一条地进行处理,其流程如图 1-2 所示。

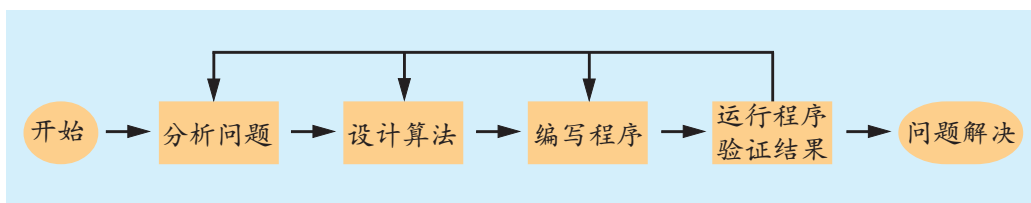


图 1-2 用计算机解决问题的流程图

虽然人类解决问题的过程和计算机解决问题的过程有某种相似性,但它们之间还是有区别的。例如,人类会创造性地解决问题,而计算机只会按照人们设计的步骤一步步地完成;人类的思维有时具有跳跃性、抽象性、模糊性等特点,而计算机则不行,你必须明确告诉它解决问题的具体对象和确切步骤。因此,在设计算法时务必按照计算机解决问题的方式,具体、明确地设计每一步的处理对象和处理过程,否则结果的正确性就难以保证。

信息论的奠基人克劳德·艾尔伍德·香农(Claude Elwood Shannon)在1950年就曾经说过,“不论是在允许的操作(移动棋子)还是在最终目标(将死对方)上,这一问题都有明确的界定。对于获得满意的解决方案来说,它既不是简单得微不足道,也不是高不可攀”。自1950年以来,数百位计算机工程师、程序员和数学家的目标一直是设计出足以战胜世界级棋类大师的具有创造力与想象力的计算机和软件。

这方面的典型案例有两个。一是1997年“深蓝”以3.5:2.5的微弱优势战胜国际象棋大师卡斯帕罗夫。二是人工智能程序阿尔法围棋(AlphaGo, 阿尔法狗)首次战胜人类围棋世界冠军。它通过训练形成一个策略网络(Policy Network),将棋盘上的局势作为输入信息,并对所有可行的落子位置生成一个概率分布,然后训练出一个价值网络(Value Network)对自我对弈进行预测,以-1(对手的绝对胜利)到1(阿尔法围棋的绝对胜利)为标准,预测所有可行落子位置的结果。这两个网络自身都十分强大,而阿尔法围棋将这两种网络整合进基于概率的蒙特卡罗树搜索(MCTS)中,从而极具优势。新版的阿尔法围棋会产生大量自我对弈棋局,为下一代版本提供训练数据,此过程可循环往复。

计算机程序(Computer Program)就是指示计算机如何去解决问题或完成任务的一组可执行的指令。

程序设计(Program Design)就是寻求解决问题的方法,并将其实现步骤编写成计算机可以执行的程序的过程。

程序设计语言(Programming Language)泛指一切用于书写计算机程序的语言。



西方数学家将“韩信点兵问题”称做“中国剩余定理”，它在世界数学史上占有重要地位，由此可见，我国古代文化在当时是相当发达的。

(1) 当不需要精确计算时，我们会使用估算的方法。例如，考虑班级外出旅游费用时，如果每人需要21元，则全班48人的费用大约为1000元。计算机会这样解决问题吗？

(2) 相传我国汉代有位大将军叫韩信，他在点兵时让士兵分别从1~3、1~5、1~7报数，然后报告各次报数的余数，这样，他就能推算出总共有多少士兵。这个问题即著名的“韩信点兵问题”，又称“鬼谷算”、“秦王暗点兵”。我国南北朝时的一部重要数学著作《孙子算经》中对此有如下记载：

“今有物不知其数，三三数之余二，五五数之余三，七七数之余二，问物几何？”

请同学们开动脑筋，算算该“物”最少有几个？看看能不能找出多种解决方法，然后将你的解题过程填在表1-1中，并就不同解决方法的步骤、效率、合理性等进行比较，相互讨论后进行评价。

表1-1 “韩信点兵问题”的求解

策略提示

逐个去试，看看能不能找到解决方案。

运用最小公倍数知识看看能不能解决。

到网上查阅资料，或者查阅相关参考书，看看有没有更巧妙的解决方法。

“韩信点兵问题”的求解

“韩信点兵问题”的求解		
求解问题的条件	已知条件	
	隐含条件	
求解方法一	具体步骤 ① ② ③ ……	评价
求解方法二	具体步骤 ① ② ③ ……	评价
求解方法三	具体步骤 ① ② ③ ……	评价

1.2 算法描述与设计

算法对于程序设计至关重要,当找到算法后还要考虑如何准确、具体地描述算法。通过本节的学习,你可以:

- 进一步理解什么是算法,知道算法的多样性
- 能够对设计的算法作出简单的评价
- 学会用自然语言、流程图和伪代码来描述算法

1.2.1 算法是“灵魂”



请同学们回忆数学课中学过的算法,谈谈你对算法是如何理解的。

简单地说,算法(Algorithm)就是解决问题的方法和步骤。“桌面小时钟问题”的求解过程就代表着一种算法。其实,我们日常生活中也经常使用算法,只是没意识到罢了。例如,我们到商店购物,首先确定要购买的东西,然后进行挑选、比较,最后到收银台付款,这一系列活动实际上就包含着算法。解决问题的过程,就是实现算法的过程。

算法是程序设计的“灵魂”,世界著名计算机科学家尼克劳斯·沃思(N. Wirth)指出:算法+数据结构(Data Structure)=程序,可见,算法在程序设计中具有多么重要的地位。算法独立于任何具体的程序设计语言,一个算法可以用多种程序设计语言来实现。例如,“桌面小时钟问题”的算法可以用BASIC语言来实现,也可以用C语言来实现。

世界上最早的算法

目前所知的世界上最早的算法是写在考古学家发掘出来的粘土板上的,这些粘土板的制作年代大约是在公元前3000年~公元前1500年,也就是大约3500~5000年以前。考古学家是在靠近古代城市巴比伦的地方发现这些粘土板的,那里离现在的巴格达不远。巴比伦人发明了六十进制系统,我们现在关于时、分、秒的记法和关于角度的记法就是从他们那里学来的。

为了做数学用表,巴比伦人需要解代数方程,他们的做法是写个求解的“算法”。在算法中,基本上都是对实际数目的计算。在算法的最后还写上一句短语,这个短语可以粗略地翻译为“这是一个过程”,这也是最早出现的关于程序设计语言的记号。

算法的特征

有穷性: 执行有限步之后结束,且每一步的执行时间也都是有限的;

确定性: 每一步都有确切的含义;

输入: 有零个或多个输入;

输出: 至少产生一个输出;

可行性: 原则上能精确运行,用纸和笔做有限次运算后即可完成。

事实上，计算机能有如此广泛而神奇的应用，除了半导体集成电路芯片制造工艺的提高外，主要是靠软件，而软件的核心是算法。无论是制造芯片的软件，还是“深蓝”战胜卡斯帕罗夫，都是靠巧妙的算法。算法是人类智慧的结晶，计算机科学中的知识创新主要就是算法的创新，创建一种新算法的意义不亚于建造一种新机型。



(1) 高斯是德国数学家，他和阿基米德、牛顿、欧拉齐名，有“数学王子”之称。高斯最出名的故事就是他十岁时，小学老师出了一道算术难题：“计算 $1+2+3+\dots+100$ 的和”。这可难为了初学算术的学生，但是高斯却很快将答案写了出来。原来他用的是凑数法： $1+49=50$ ， $2+48=50$ ，…你是否有更简便的算法？请同学们对不同的算法进行比较，看看哪种算法的效率更高，从中可以得出什么结论，体会一下算法的多样性与复杂性。

(2) “水仙花数问题”：水仙花数是指一个三位数，它的各位数的立方和正好等于该数本身。如： $153=1^3+5^3+3^3$ 。请设计算法求解“水仙花数问题”。

1.2.2 如何描述算法

当我们找到算法之后，不能只是自己心知肚明，还需要准确、具体地将它描述出来，这样才能编写成程序供计算机执行。描述算法可以有多种表达方法，一般用自然语言、流程图和伪代码进行描述。

1. 用自然语言描述算法

自然语言是指人们日常生活中使用的语言，如汉语、英语、法语等都是自然语言。用自然语言描述算法符合我们的表达习惯，并且容易理解。

要让计算机解决问题，必须明确地告诉它要处理的具体对象和每一步准确的处理过程，否则计算机就无法工作，因此，算法的描述要求尽可能精确、详尽。

对于“桌面小时钟问题”，我们可以用算法语言描述其中的关键步骤。

(1) 算法分析。

先检查小时钟数字的显示状态（处于变化还是停止状态），如果是变化状态，就让它停止，如果是停止状态，就让它开始变化。

(2) 算法描述。

Step1: 检查小时钟数字的显示状态。

Step2: 如果是变化状态, 就让它停止。

Step3: 如果是停止状态, 就让它开始变化。



凯撒密码的原理是将“明文”中的每个字母用另外一个字母替换, 这样就形成“密文”。已知凯撒密码的计算公式为 $F(a)=(a+k) \text{ Mod } n$, 其中 $k=3$, $n=26$, 如果将英文字母进行加密, 其对应关系如下所示。

明文: A B C D E F …… X Y Z

密文: D E F G H I …… A B C

现给出待加密字符串为“PROGRAM”, 请同学们设计算法, 然后用自然语言将它描述出来, 填入表 1-2 中, 并相互评价所采用的算法。

表 1-2 凯撒密码的算法描述

用自然语言描述凯撒密码的算法	
算法分析	
算法描述	
相互评价	你认为上面的算法描述 <input type="checkbox"/> 较好 <input type="checkbox"/> 一般 <input type="checkbox"/> 不好

Mod 为取模运算, 即求余数。
如:

7 Mod 3 结果为 1

10 Mod 5 结果为 0

11 Mod 6 结果为 5

4 Mod 5 结果为 4

用自然语言描述算法的优点是通俗易懂, 但缺乏直观性和简洁性, 并且容易产生歧义。例如, “这个人谁都不认识”。首先, 这句话隐含了一个条件: “在某个特定场合下”; 其次, 它到底表示在这个特定场合下没人认识他, 还是他不认识任何人呢? 可见, 用自然语言描述算法有时容易产生误解。

2. 用流程图描述算法

流程图(Flow Chart)也称为程序框图, 它是算法的一种图形化表示方法。与用自然语言描述算法相比, 用流程图描述算法形象、直观, 更容易理解。对于一些复杂的算法, 直接编写代码很难保证程序的正确性, 程序设计人员往往先用流程图描述算法, 然后根据流程图就可以方便地写出程序代码了。

目前常用的流程图由美国国家标准化协会(American National Standard Institute, 简称 ANSI)制定的一系列流程图符号组成, 如图 1-3 所示。

用流程图描述“桌面小时钟问题”的算法的结果, 如图 1-4 所示。

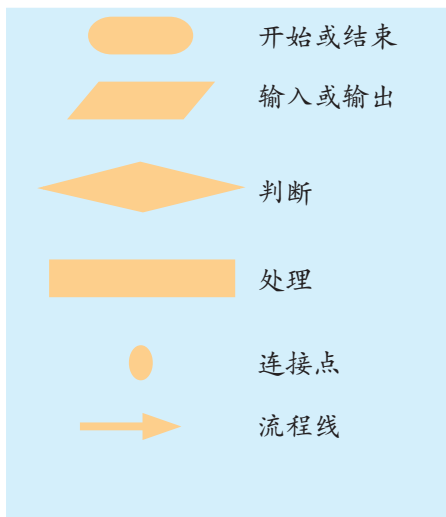


图 1-3 流程图图例

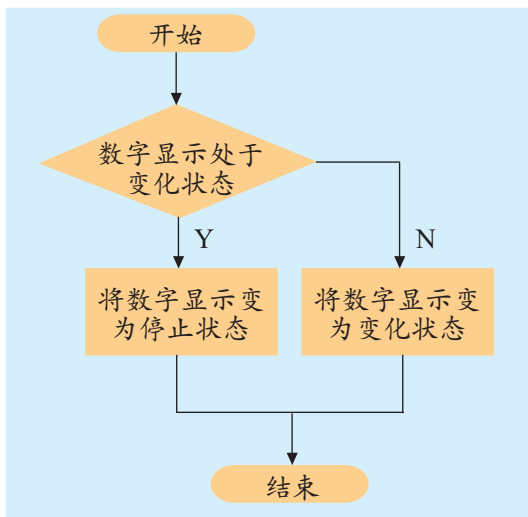


图 1-4 “桌面小时钟问题”的流程图



参照表 1-3，试比较“桌面小时钟问题”的算法用自然语言描述和用流程图描述各有什么特点。

表 1-3 比较“桌面小时钟问题”的算法描述

自然语言描述与流程图描述的比较		
	自然语言描述	流程图描述
直观清晰		
确定性		
烦琐程度		
是否容易修改		
通俗易懂		

3. 用伪代码描述算法

伪代码表示举例

(1)表示条件判断:

```
If 条件 Then
    执行语句一
Else
    执行语句二
End If
```

(2)表示循环处理:

```
Do While 条件
    循环处理语句
Loop
```

伪代码(Pseudocode)是介于自然语言和计算机程序语言之间的一种算法描述。它也是专业软件开发人员描述算法的一种常用方法。

使用伪代码描述算法没有严格的语法限制,书写格式也比较自由,只要把意思表达清楚就可以了,它更侧重于对算法本身的描述。

在伪代码描述中,表示关键词的语句一般用英文单词,其他语句可以用英文语句,也可以用汉语语句。

例如,给定一个四位数的年份,判断它是否为闰年。如果用伪代码来描述它的算法,可以表示如下。

(1) 算法分析。

如果2月份是28天,则这一年是平年;如果是29天,则这一年是闰年。判断闰年的条件是:如果该年份能被4整除但不能被100整除,或者能被400整除,则该年为闰年。

(2) 算法描述。

```

输入年份→y
If y 能被4整除 Then
    If y 不能被100整除 Then
        输出“是闰年”
    Else
        If y 能被400整除 Then
            输出“是闰年”
        Else
            输出“不是闰年”
        End If
    End If
End If
Else
    输出“不是闰年”
End If

```

由此可以看出，用伪代码描述的算法简洁、易懂，修改起来也比较容易，并且很容易转化为程序语言代码。但用伪代码描述的算法也有缺点，例如，它没有流程图描述的算法直观，一旦出现逻辑错误也不便排查。



比较自然语言、流程图和伪代码三种算法描述方法，谈谈你自己的看法，你更喜欢哪种描述方法？

1.3 初试身手，体验编程

1.2节我们学习了用自然语言、流程图和伪代码三种方法描述算法，但计算机是无法直接识别这些描述的。对于用计算机解决问题而言，设计算法、描述算法不是最终目的，最终目的是编写程序代码，实现算法设计。下面就让我们用Visual Basic程序开发工具尝试编写一段小程序，实现“桌面小时钟问题”的算法，然后请同学们通过自主学习，认识一下有关程序设计方面的知识。通过本节的学习，你可以：

- 初步体验编程的乐趣，知道什么是程序源代码
- 进一步领会算法和程序设计在解决问题中的地位
- 理解算法、程序设计、程序设计语言之间的关系
- 了解程序设计语言的发展及种类
- 了解如何编辑程序、编译程序和连接程序

1.3.1 程序编写尝试

Visual Basic 6.0(简称 VB 6.0)是一种基于 BASIC 语言的可视化程序开发工具。下面就让我们用 VB 6.0 一步一步尝试编写程序,实现“桌面小时钟问题”的算法,体会程序设计的魅力。

Step1: 通过“开始”菜单启动 Visual Basic 6.0,在“新建工程”对话框中选择“标准 EXE”,然后单击“打开”按钮,建立标准 EXE 工程,如图 1-5 所示。



图 1-5 新建 VB 工程

Step2: 单击工具箱中的 Timer 控件,在默认窗体“Form1”的适当位置上拖动鼠标添加“Timer1”控件,在属性栏中将该控件的名称改为“tmrClock”。用同样的方法添加

标签控件“lblShow”。将默认窗体“Form1”的“BorderStyle”属性改为“0-None”,去掉窗体的标题栏,如图 1-6 所示。



图 1-6 属性设置

Step3: 为了使窗体背景与桌面背景颜色相同,修改窗体的“BackColor”属性为“系统-桌面”,如图 1-7 所示。将“lblShow”



图 1-7 颜色设置

的“Caption”属性设为“12345”,“AutoSize”属性设为“True”。将“tmrClock”的“Interval”属性改为 1000 (即每秒更新一次)。

Step4: 双击窗体,切换到“代码窗口”,在“代码窗口”中填写如下代码。

```
Private Sub Form_Activate()
    Me.BackColor = RGB(0, 78, 152)           ' 使窗体的背景与桌面的颜色一致
    Me.Move Screen.Width - Me.Width, 0     ' 将窗体移动到桌面右上角
End Sub
Private Sub lblShow_Click()
```

```

    tmrClock.Enabled = Not tmrClock.Enabled ' 切换时钟状态
End Sub
Private Sub lblShow_DblClick()
    End ' 结束程序的语句
End Sub
Private Sub tmrClock_Timer()
    lblShow.Caption = Time() ' Time 是函数，调用它可以得
    ' 到计算机的系统时间
End Sub

```

Step5: 执行“运行”菜单中的“启动”命令（或按 F5 键）运行程序。程序运行结果如图 1-1 所示。



试着改变小时钟显示的位置，将它“挪”到桌面的左上角，如何实现呢？注意，我们并没有讲解移动窗体的办法，但是程序中用到了。请发挥你的聪明才智，完成上述任务。

1.3.2 认识程序设计语言



阅读下面的资料，并上网查找相关资料，填写表 1-4。

资料一

程序设计语言的发展经历了机器语言、汇编语言到高级语言的发展过程。

对于计算机来说，它能直接识别的语言是机器语言，机器语言是一串由“0”和“1”构成的二进制代码。由于理解和记忆机器语言非常困难，并且容易出错，工作效率低，于是科学家发明了汇编语言、高级语言。

汇编语言实际是一种符号化的机器语言，它采用英文助记符代替机器指令，比机器语言容易识别和记忆，从而提高了程序的可读性。但是汇编语言仍然是面向机器的语言，是为特定的计算机系统设计的，依赖于特定的指令集，它要求程序设计人员对相应的机器硬件非常熟悉，因而汇编语言属于低级语言。汇编语言毕竟还不是机器语言，因而，用汇编语言编制的源程序不能被计算机执行，必须经过处理程序转换为目标程序。

高级语言是更接近于自然语言（英语）和数学语言的编程语言，它并不特指某一种语言，也不依赖于特定的计算机系统，因而更容易掌握和使用，通用性也更好。比较流行的高级语言有 BASIC、C/C++、FORTRAN、PASCAL、COBOL、JAVA 等。用高级语言编制的源程序可读性更强，也便于修改、维护。

就像汇编语言一样，用高级语言编制的源程序计算机也无法直接识别，必须经过转换才能被计算机执行。按转换方式不同，可以将高级语言分为编译型语言和解释型语言。

对于编译型语言,编写的源程序需要用编译程序先翻译成机器语言的目标程序,然后再由连接装配程序进行连接装配,生成可执行程序,这样才能被计算机执行。像C/C++、PASCAL、COBOL等语言都属于编译型语言,Visual Basic也已发展为编译型语言。

对于解释型语言,源程序输入计算机后,运行源程序,相应的解释程序会逐条分析源程序中的语句,每解释一句由计算机执行一句,QBASIC、LISP等属于解释型语言。它和编译型语言的差别在于不产生目标程序,而是直接执行源程序,每次执行都要逐条进行解释。

随着软件业的发展,软件开发工具也越来越丰富,例如,基于BASIC语言的Visual Basic、Visual Basic.Net;基于PASCAL语言的Delphi以及基于C++语言的VC++、C++ Builder等,这些开发工具使软件的开发效率更高、功能更强、使用也更方便。

资料二

Python最初是由吉多·范罗苏姆(Guido van Rossum)在1989年圣诞节期间,为了打发无聊的圣诞节而编写的一个编程语言。Python是一种相当高级的程序设计语言,用它编写程序非常简便。有人说,完成同一个任务,使用汇编语言需要1000行代码,使用C语言需要500行,使用Java只需要100行,而使用Python可能只要20行就可以了。Python程序简单易懂,初学者容易入门。若能持续深入学习,用户也可以用它编写出非常复杂的程序。之所以如此方便易用,是因为Python有非常完善的基础代码库。这些代码库覆盖了网络、文件、图形用户界面、数据库、文本等大量内容,所以也被形象地称为“内置电池(Batteries Included)”。用Python进行项目开发,许多功能的实现不必从零开始编写代码,可以从基础代码库中选择现成的。除了内置的库外,Python还有大量的第三方库,也就是别人开发的、可供直接使用的代码。当然,你开发的代码通过封装,也可以作为第三方库提供给别人使用。

表1-4 程序设计语言对比

认识程序设计语言	
程序设计语言发展经历的阶段	
程序设计语言举例	低级语言:
	高级语言:
程序设计语言的分类	
解释型语言、编译型语言举例	解释型语言:
	编译型语言:
编辑程序、编译程序的区别	
可执行程序的生成步骤	

第二章 程序设计 ABC

- 可视化程序开发工具
- 数据的表示与处理
- 程序的基本结构

本章我们将通过编制一个有趣的打字测试程序,熟悉 VB 的集成开发环境,知道数据在计算机中是如何表示与处理的,初步掌握程序执行的三种基本结构:顺序结构、选择结构和循环结构。

2.1 可视化程序开发工具

VB 是一种被广泛使用的程序开发工具，VB1.0 版本推出的时候，微软总裁比尔·盖茨说 VB 是“用 BASIC 语言开发 Windows 应用程序最强有力的工具”，是“令人震惊的新奇迹”。此后，微软公司相继推出 VB 2.0、VB 3.0、VB 4.0、VB 5.0、VB 6.0、VB.NET 等版本。

大家初学打字的时候都使用过打字软件，那么，我们能不能自己设计程序实现打字测试功能呢？本节我们将用 VB 开发工具设计一个简易的打字测试程序。通过本节的学习，你可以：

- 熟悉 VB 集成开发环境
- 了解可视化编程的一般步骤
- 理解控件、对象、属性等基本概念
- 学会使用 VB 中的标签、文本框、按钮等基本控件

2.1.1 认识我们的“创作室”

Visual Basic 是美国微软公司于 1991 年推出的基于 Basic 语言 (Beginners All-purpose Symbolic Instruction Code, 初学者通用指令码) 的可视化程序设计语言。所谓可视化 (Visual) 程序设计是一种全新的程序设计方法，它主要是让程序设计人员利用软件本身所提供的各种控件，像搭积木式地构造应用程序的各种界面，然后再编写少量的代码就可以构建应用程序。

启动 VB 6.0，在“新建工程”对话框中选择“标准 EXE”后单击“打开”按钮就会看到我们的“创作室”：VB 集成开发环境，如图 2-1 所示。

如果你看不到图 2-1 中的某些窗口，可以通过“视图”菜单中命令或工具栏中的按钮调出。

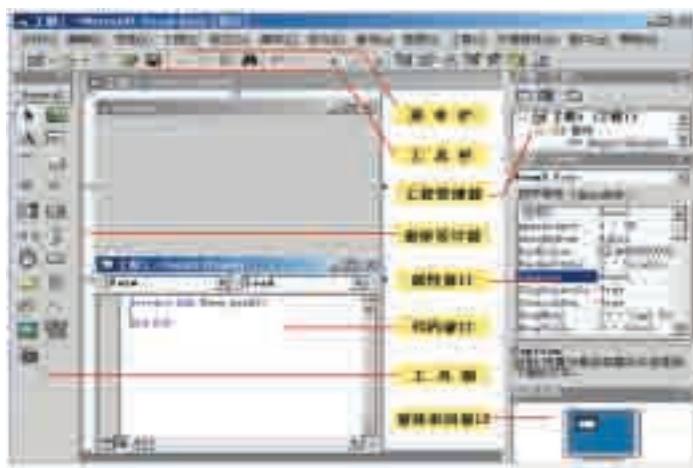


图 2-1 VB 集成开发环境

创建新工程有两种方法：

(1) 启动 VB 后，在“新建工程”对话框中选择“标准 EXE”，然后单击“打开”按钮。

下面设计一个程序，当用鼠标单击窗体时，窗体中显示“欢迎进入 VB 世界！”字样。操作步骤如下。

Step1: 新建“标准 EXE”工程，进入 VB 集成开发环境。系统自动创建一个窗体“Form1”，如图 2-1 所示。

Step2: 双击“Form1”窗体，系统弹出代码窗口，如图 2-2 所示。

Step3: 在如图 2-3 所示的对象(Object)列表框中选择对象“Form”，在过程列表框中选择“Click”事件(Event)过程。

Step4: 在两行代码之间插入如下代码，如图 2-3 所示。

```
Print "欢迎进入 VB 世界!"
```

(2) 在 VB 主窗口中选择“文件”菜单中的“新建工程”命令。



图 2-2 代码窗口


图 2-3 编写代码

Step5: 保存文件。

一个工程 (*.vbp) 中往往包含多个不同类型的文件，如窗体文件 (*.frm) 等，需要分别保存。

(1) 单击“文件”菜单中的“保存工程”命令，弹出“文件另存为”对话框，设置保存位置(如“我的文档”)及文件名(如 vb01.frm)，然后单击“保存”按钮即可保存窗体文件，如图 2-4 所示。

(2) 接着系统会提示保存工程文件，设置工程文件名(如 vb01.vbp)，然后单击“保存”按钮即可。

Step6: 单击工具栏中的“启动”按钮 ，或按 F5 键运行程序。


程序运行后显示一个窗体，当你用鼠标单击窗体时，它会显示“欢迎进入 VB 世界!”字样，如图 2-5 所示。

保存工程文件时，安装了 SourceSafe 组件的 VB 系统会弹出“Source Code Control”对话框，询问是否将工程文件添加到 SourceSafe 中。SourceSafe 对研发小组共同开发的代码提供版本安全管理。



图 2-4 保存文件对话框

图 2-5 程序运行结果

要结束程序的运行，可以单击程序窗体的“关闭”按钮，或单击 VB 工具栏中的“结束”按钮 。



(1) 将鼠标指针指向工具箱中的某个控件并稍作停留，VB 就会显示该控件相应的名称。请大家利用此功能认识一下 VB 工具箱中的控件。

(2) 比较 VB 和 Word、Excel 等常用软件，看看它们的菜单栏、工具栏有何异同。

2.1.2 可视化编程的一般步骤

VB 应用面向对象的程序设计方法,把程序和数据“封装”起来成为对象。开发人员只需按设计要求的屏幕布局,用系统提供的工具,直接在窗体中“画”出命令按钮、文本框等不同对象就可以设计图形用户界面。

你现在了解我们的“创作室”了吗?下面让我们一起尝试用它设计一个简易打字测试程序。

简易打字测试程序的界面如图 2-6 所示,单击“开始”按钮,计时开始,输入字符;单击“结束”按钮,程序输出打字速度。

Step1: 启动 VB,新建一个 VB 标准 EXE 工程。单击工具箱中的文本框控件 (TextBox),将鼠标指针移到窗体上,拖动十字线在窗体上画出大小适当的控件,如图 2-7 所示。



图 2-6 打字测试程序的界面



图 2-7 界面设计窗口

控件是应用程序的基本元素,与窗体共同构成应用程序的界面。控件以图标的方式放在“工具箱”中。

Step2: 如图 2-8 所示,用同样的方法在窗体中添加两个命令按钮控件 (CommandButton)、两个标签控件 (Label) 和一个文本框控件。

接下来设置对象的属性 (Property)。



图 2-8 打字测试程序界面设计

标签控件主要用来显示不希望被用户修改的文本。

文本框控件是一个文本编辑区域,用户可以在该区域中输入、编辑和显示文本内容。

在日常生活中,我们经常接触“对象”。比如,操作计算机,对象是计算机;小孩玩气球,对象是气球。在可视化程序开发环境中,文本框、标签、命令按钮等都是对象。对象的特征称为对象的属性,例如,文本框具有名称、字体等属性。修改对象的属性值,就会改变对象的特征。

以小孩玩的气球来说,充气前和充气后的属性变化如图 2-9 所示。

充气前和充气后气球的形状改变如图 2-10 所示。



图 2-9 气球属性的变化



图 2-10 气球形状的变化

Step3: 通过属性窗口可以更改对象的原有属性, 例如, 单击“Form1”窗体, 然后在属性窗口中将其“名称”属性改为“frmTest”, 如图 2-11 所示。

用同样的方法, 按表 2-1 所示改变对象的属性。



图 2-11 更改对象属性

表 2-1 对象的属性设置

对象	属性	设置值
Form1	名称	frmTest
	Caption	打字测试软件
Text1	名称	txtInput
	Text	空
	MultiLine	True
Text2	名称	txtSpeed
	Text	空
Command1	名称	cmdStart
	Caption	开始
Command2	名称	cmdEnd
	Caption	结束
Label1	名称	lbls
	Caption	速度
Label2	名称	lbld
	Caption	字 / 分钟

每个对象都有自己的名称, 有了名称才能在程序中引用该对象。在建立对象时, VB会自动提供一个诸如Command1、Command2、Label1、Label2之类的默认名。为了使程序便于阅读、理解, 最好给对象设置一个有意义的名称, 如cmdStart。一般约定对象名由3个小写字母组成的前缀(指明对象的类型)和表示该对象作用的字符组成, 表 2-2 是一些控件名的前缀约定和命名举例。

表 2-2 部分控件名的前缀约定和命名举例

对象的类型	意义	前缀	命名举例
Form	窗体	frm	frmTest
Command	命令按钮	cmd	cmdStart
Label	标签	lbl	lblDisplay
TextBox	文本框	txt	txtInput

Step4: 保存文件。

.vbp 文件: 与 VB 工程有关的文件和对象清单, 是一个文本文件。

.frm 文件: 包含窗体及控件的属性设置、相关程序代码等。

.ocx 文件: ActiveX 控件文件, 为 VB 提供扩展控件。

.bas 文件: 标准模块文件。



- (1) 请查阅英汉字典或参考相关资料, 判断表 2-3 中的属性在 VB 中的含义。
- (2) 表 2-3 中的 Caption、Visible、BackColor 都是 VB 中常用对象的属性, 从中你得到什么启发?
- (3) 请尝试修改“打字测试软件”窗体的 BackColor 和 Font 属性, 改变窗体的背景色和字体, 设计一个自己喜欢的界面。

表 2-3 对象的属性在 VB 中的含义

序号	对象的属性	在 VB 中的含义
①	Caption	
②	Font	
③	Visible	
④	Width	
⑤	Top	
⑥	Enabled	
⑦	BackColor	
⑧	ForeColor	
⑨	BorderStyle	

- (4) 请按图 2-12 所示为奥运会设计一个奖牌总数统计程序界面。

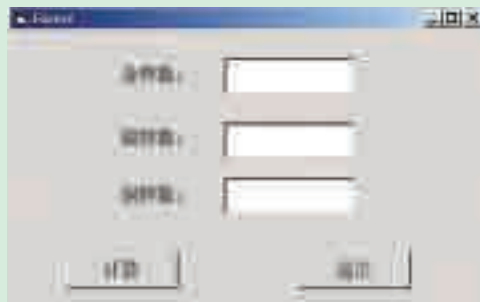


图 2-12 奖牌总数统计程序界面

Step5: 编写代码。

双击“frmTest”窗体中的“开始”按钮, 打开对应的代码窗口。在对象列表框中选择“通用”, 输入图 2-13 所示的代码。



图 2-13 “通用”声明

在对象列表框中选择“cmdStart”，在过程列表框中选择“Click”事件，输入如图 2-14 所示的代码。



图 2-14 “开始”按钮的代码

在对象列表框中选择“cmdEnd”，在过程列表框中选择“Click”事件，输入如图 2-15 所示的代码。

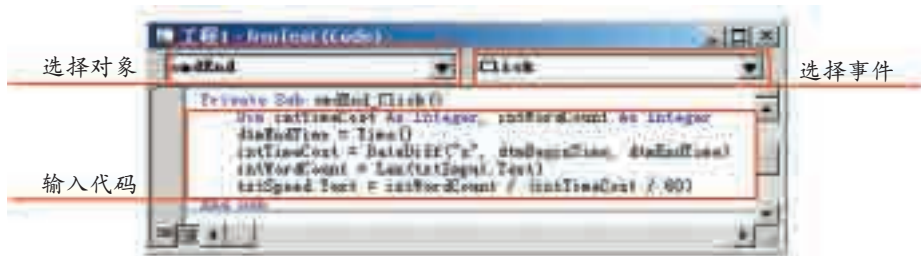


图 2-15 “结束”按钮的代码

Step6: 保存工程，运行程序。

程序运行后，单击“开始”按钮开始测试，在文本框中输入字符；单击“结束”按钮停止测试，程序显示打字测试的速度。

如需多次测试打字速度，需要手工将前面输入的字符删除后重新开始测试。

打字测试程序终于制作完毕了，回顾制作过程，我们可以归纳出可视化编程的一般步骤，如图 2-16 所示。

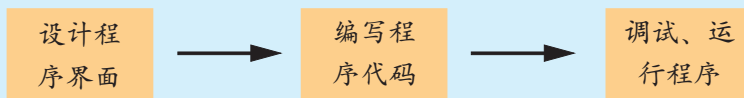


图 2-16 可视化编程的一般步骤

面向对象的程序设计(Object-Oriented Programming, 简称 OOP)、方法(Method)、事件、事件驱动(Event Driven)的含义是什么?

面向对象的程序设计以对象为中心，以事件为过程执行的起点。

在上例中，单击“开始”和“结束”按钮都是事件，VB 提供的可编程事件有单击(Click)、双击(DblClick)、加载(Load)、按键(KeyPress)等。

方法是对象能够执行的操作。如：Form1.Cls 的作用是清除窗体 Form1 中的内容。

为了突出程序的层次结构，代码采用缩进格式，一般情况下，可按下 Tab 键使每个层次缩进一个制表位。

若程序运行过程中出错，系统显示出错信息，自动进入“中断”模式，回到代码窗口提示修改代码，修改好后再次运行程序，直到正确为止。

单击“开始”按钮，程序就执行“cmdStart”的Click事件过程，记下当时的时间；单击“结束”按钮，程序就执行“cmdEnd”的Click事件过程，计算出打字的速度，并将打字速度显示在“速度：”右边的文本框中。当事件作用于对象时，程序就被告知执行相应的操作，我们把这种机制叫做事件驱动。

对象的属性、事件、方法等知识，我们将在第四章中进一步学习。



- (1) 谈谈你对于“Visual Basic”的“Visual”的体会。
- (2) 标签和文本框控件都可以显示文本，讨论它们之间有什么区别。
- (3) 结合使用Windows的经验，对照VB工具箱，认识VB中标准控件的用途。
- (4) 利用因特网搜索有关VB的教学网站，并记录下网址或添加到收藏夹中，供以后学习使用。

2.2 数据的表示与处理

在不同的程序设计语言中，数据表示与处理方法不尽相同。在VB中数据到底是如何表示的？如何赋值、计算、输出数据呢？通过本节的学习，你可以：

- 知道VB的常用数据类型、常量、变量、运算符及表达式的含义
- 学会定义变量，学会使用常用语句及标准函数

2.2.1 数据类型

在程序设计中，往往要处理各种各样的数据，不同的程序设计语言有不同的数据类型(Data Type)；在2.1节的“打字测试软件”程序中，涉及到的数据就有开始时间、结束时间、输入字符所花时间、输入字符数、打字速度等。

数据类型实际上是一种约定。我们知道，计算机中是0和1的世界，它的数据是以0和1的形式存储的。如二进制数“01000001”，如果约定为数值，它表示整数“65”；如果约定为字符，则表示字母“A”。不同的约定，计算机分配的存储空间也是不同的。在VB中，为了便于数据的表示与处理，VB提供了数值型、字符串型、布尔型、日期型等基本数据类型，表2-4所示的是我们常用的数据类型。

表 2-4 VB 的数据类型

数据类型	关键字	所占内存	取值范围
整型	Integer	2 字节	-32768 ~ 32767
长整型	Long	4 字节	-2147483648 ~ 2147483647
单精度型	Single	4 字节	负数: -3.402823E38 ~ 1.401298E-45 正数: 1.401298E-45 ~ 3.402823E38
双精度型	Double	8 字节	负数: -1.797693134E308 ~ -4.940656458E-324 正数: 4.940656458E-324 ~ 1.797693134E308
字符串型	String	10 字节 + 字符串长度	定长字符串: 0 ~ 65535 个字符 不定长字符串: 0 ~ 20 亿个字符
布尔型	Boolean	2 字节	True 或 False
日期型	Date	8 字节	100/1/1 ~ 9999/12/31

在表 2-4 中, 整型、长整型、单精度型和双精度型都属于数值型, 它们表示数值的范围是不同的。其中, 单精度型和双精度型又合称为浮点型。

字符串型数据是指用 " " 引起来的一串字符。如 " 欢迎进入 VB 世界! "、" Visual Basic "、" 1234 " 都是字符串类型。

布尔型又称逻辑型, 常用于表示逻辑判断的结果, 如“真/假”、“是/非”等, 其值为 True 或 False, 分别表示逻辑真和逻辑假。VB 中也常用数值 0 表示假值, -1 表示真值。

日期型数据专门用来处理日期和时间。VB 采用一对“#”把日期和时间的值括起来, 如 #2010/08/20# 表示 2010 年 8 月 20 日。



- (1) 区分 VB 中的“数据”和数学课中的“数据”之间的差异。
- (2) 分析在 2.1 节编写的“打字测试软件”程序中用到了哪些数据类型。

2.2.2 常量与变量

1. 常量

在“打字测试软件”程序中, 计算打字速度时, 我们需要将打字时间除以 60, 把秒转化为分钟; 在程序运行期间“60”始终保持不变, 我们把它叫做常量(Constant)。VB 中的常量有数值常量、字符串常量等。

[As 类型] 中,“[]”表示其中的内容是可选的,也就是说,定义符号常量PI时,可以写成: Const PI = 3.1415926,也可以写成: Const PI As Single = 3.1415926。

(1) 数值常量类似于数学中的常数,如前面的“60”。

(2) 用一对双引号引起来的字符串称为字符串常量。例如,“China”、“Visual Basic”、“18”、“3.14”等。字符串常量指的是双引号中的字符,不包括双引号本身。

如果程序中多次用到同一个常量,往往用一个有意义的符号表示,这种常量称为符号常量;代表常量的符号称为“常量名”,其定义形式如下:

Const 常量名 [As 类型]=表达式

如: Const PI = 3.1415926



(1) 比较下面两段计算圆周长和面积的程序,你认为哪段程序更好,为什么?

```
Private Sub Form_Click()
    Dim r As Single ' 定义 r 为单精度型
    Dim c As Single ' 定义 c 为单精度型
    Dim s As Single ' 定义 s 为单精度型
    r = InputBox(" 输入 r: ") ' 输入半径 r
    c = 2*3.14*r ' 计算周长
    s = 3.14*r*r ' 计算面积
    Print c ' 输出周长
    Print s ' 输出面积
End Sub
```

```
Private Sub Form_Click()
    Const PI=3.14 ' 定义符号常量
    Dim r As Single ' 定义 r 为单精度型
    Dim c As Single ' 定义 c 为单精度型
    Dim s As Single ' 定义 s 为单精度型
    r = InputBox(" 输入 r: ") ' 输入半径 r
    c = 2*PI*r ' 计算周长
    s = PI*r*r ' 计算面积
    Print c ' 输出周长
    Print s ' 输出面积
End Sub
```

(2) 如果把 π 值改为3.14159265,哪段程序修改起来更方便?为什么?

InputBox
(提示信息): 用于产生一个能接受用户输入的对话框。

2. 变量

变量(Variable)原来是数学里的概念,程序设计借用过来,赋予它新的含义。

我们知道计算机在处理数据时，首先要将数据调入内存单元，然后才能进行各种操作。而内存单元类似旅馆里的房间，只有编号没有具体名称，编程时如何操纵这些单元呢？高级语言是借助变量实现的。

程序设计中的变量和家庭使用的信箱类似：家庭信箱名称对应一个具体的收信地点；变量有一个名字，它对应着相应的内存单元。

(1) 变量命名。

变量名可以简单地用 i、j、k 等单个字母表示，也可以包含比较复杂的信息描述，如“打字测试软件”程序中使用的 intTimeCost 和 intWordCount，分别表示打字所花时间和输入的字符数。尽管变量命名可以有很大范围的自由度，但是还必须遵守以下几个约定：

- ① 必须以字母或汉字开头，而不能以数字或其他字符开头。
- ② 只能由字母、汉字、数字和下划线组成，不能含有小数点、空格等字符。
- ③ 字符个数不超过 255 个。
- ④ 变量名不能使用 VB 中的保留字。
- ⑤ VB 不区分变量名中字母的大小写。如 HELLO 和 Hello 是同一个变量。
- ⑥ 为增加程序的可读性，变量名最好取有意义的名称，建议前面加上数据类型的缩写。

保留字也称为关键字，由 VB 专用，具有特殊含义。例如，Print、Sub、End 等都是 VB 中的保留字。

VB 编码约定：变量命名时，往往用前缀表明数据类型，这种命名规则是由匈牙利规则演变而来的。表 2-5 列出了一些常用的变量类型的前缀。

表 2-5 常用的变量类型的前缀

变量类型	字符型	整型	长整型	单精度型	双精度型	布尔型	日期型
前缀	str	int	lng	sng	dbl	bln	dtm

(2) 变量声明。

变量应该先声明后使用。VB 中声明变量的方法是：

```
Dim 变量名 As 数据类型
```

例如：

```
Dim N As Integer
```

还可以在一行中定义多个变量，例如：

```
Dim dtmBeginTime As Date, dtmEndTime As Date
Dim intTimeCost As Integer, intWordCount As Integer
```




(1) 请用英语表示 a, b, c 为整型数据, 再用 VB 语言定义 a, b, c 为整型变量, 试比较用自然语言描述和程序设计语言描述有何异同。

(2) 在窗体的 Click 事件中编写一段程序求 5!, 程序代码如下。

```
Private Sub Form_Click()
    Dim i As Integer, x As Integer
    x = 1
    For i = 1 To 5
        x = x*i
    Next i
    Print "x = ";x
End Sub
```

请留意求 5! 的程序代码中变量的定义语句, 完善以下两段程序, 并上机运行。

① 计算 10!。

```
Private Sub Form_Click()
    Dim i As _____, x As _____
    x = 1
    For i = 1 To 10
        x = x*i
    Next i
    Print "x = ";x
End Sub
```

② 计算 100!。

```
Private Sub Form_Click()
    Dim i As _____, x As _____
    x = 1
    For i = 1 To 100
        x = x*i
    Next i
    Print "x = ";x
End Sub
```

完善程序后, 请上机运行。

(3) 比较第 (2) 题中的三段程序, 说出定义不同数据类型变量的理由。

2.2.3 运算符与表达式

在“打字测试软件”程序中, 我们是这样计算打字速度的:

$$\text{打字速度} = \text{打字数} \div (\text{打字所用秒数} \div 60)$$

在程序设计语言中, 我们如何表示“打字数 \div (打字所用秒数 \div 60)”呢? VB 是用“表达式”来完成这个任务的。

表达式(Expression)是由变量、常量、运算符、函数和圆括号等按一定的规则组合而成的。VB中有算术表达式、字符串表达式、日期表达式、关系表达式和逻辑表达式等。

例如,在“打字测试软件”程序中计算打字速度的表达式如下:

```
intWordCount/(intTimeCost /60)
```

与数学运算一样,VB也可以使用括号控制运算顺序。但是,与数学中不同的是,VB中没有“[]”和“{}”,而是用“()”的嵌套来表示,VB括号里的表达式优先运算。

1. 算术运算符与算术表达式

VB中有很多算术运算符(Operator),如“+”、“-”、“*”、“/”等,用它们可以构造出各种各样的算术表达式。算术运算符要求参与运算的量是数值型,算术表达式运算的结果也是数值型。算术运算符的运算规则及优先级如表2-6所示。

表2-6 算术运算符的运算规则及优先级

算术运算符	含义	运算优先级	实例	结果
^	指数	1	2^10	1024
-	取负	2	-5+2	-3
*	乘法	3	5*4	20
/	除		5/2	2.5
\	整除	4	5\2	2
Mod	求余	5	5 Mod 2	1
+	加	6	20+5	25
-	减		20-5	15

2. 字符串运算符与字符串表达式

VB中字符串运算符有: + 和 &。它们的作用都是将两个字符串连接起来。

"计算机" & "软件" 的运算结果是 "计算机软件"。

"123" + "45" 的运算结果是 "12345"。

需要注意的是: 当一个数字字符串表达式(如 "123")与一个数值型表达式用“+”运算时,结果将是数值型数据。如 "123" + 45 的结果是 168; 而用“&”运算的结果仍是字符串类型的数据 "12345"。

3. 关系运算符与关系表达式

关系运算符有： $=$ 、 $<>$ 、 $<$ 、 $>$ 、 $>=$ 和 $<=$ 。关系表达式的运算结果为布尔值 True 或 False，如 $3 > 2$ 的结果为 True。所有关系运算符优先级相同。

4. 逻辑运算符与逻辑表达式

常用的逻辑运算符有：Not(取反)、And(与)、Or(或)等，运算结果为布尔值 True 或 False。逻辑运算符的优先级为：Not>And>Or。如“韩信点兵问题”程序代码中的语句：

```
If (N Mod 3 = 2) And (N Mod 5 = 3) And (N Mod 7 = 2) Then
```

其中就包含了逻辑表达式。

5. 日期运算符与日期表达式

日期表达式是用日期运算符“+”或“-”将算术表达式、日期型常量、日期型变量、日期函数等连接起来的式子，如表达式“#2010/09/01#+30”的运算结果为“2010-10-1”；表达式“#2010/10/20#-#2010/10/10#”的运算结果为整数 10。

6. 运算符的执行顺序

当一个表达式中同时出现多种运算符时，如何决定运算的次序呢？在 VB 中，运算次序由运算符的优先级决定，优先级高的运算符先运算，优先级相同的从左向右进行运算。不同性质的运算符优先级如下：

算术运算符 > 字符串连接运算符 > 关系运算符 > 逻辑运算符



(1) 你能写出下列表达式的值吗？

① $123 + "321" \& "ABCD"$

② $\text{Not}(2 * 5 <> 11)$

(2) 把表 2-7 中的代数式改写成 VB 表达式。

表 2-7 改写代数式

代数式	VB 表达式
$\frac{x+y}{x-y}$	
$\frac{1}{2}gt^2$	
$b^2 - 4ac$	

2.2.4 常用语句与函数

在实际编程中,许多处理过程并不需要我们亲自编写,VB 提供了大量现成的语句与函数供我们使用。

1. 赋值语句

VB 中的赋值语句用“=”来实现,如:

```
intBeginTime = Time() ' 将系统当前时间赋给变量 intBeginTime
N = 1 ' 将 1 赋给 N
```

赋值语句中的“=”左边只能是变量,右边可以是常量、变量或表达式,它表示将“=”右边的值赋给左边的变量。如:

```
N = N+1
```

这里的意义是将 N 的值加 1 后重新赋给 N,结果 N 的值比原值增加了 1。

2. Print 语句

语法格式: [对象名.]Print [表达式列表]

功能: 在指定的对象上输出文本。

对象可以是窗体、图片框等。如果不指定对象名,默认在当前窗体中输出。如在窗体的 Click 事件过程中添加如下代码:

```
Private Sub Form_Click()
    Print "*****"
    Print "  新北京  新奥运  "
    Print "*****"
End Sub
```

程序运行后,单击窗体,窗体中将显示如下字符:

```
*****
  新北京  新奥运
*****
```

变量被赋值前的值常称为“初始值”。

VB 中字符串变量的初始值为空串;数值型变量的初始值为 0;逻辑型变量的初始值为 False。

Print 语句行末若没有分隔符“,”或“;”,则 Print 语句执行完后自动换行。

表达式列表可以是一个或多个表达式，各表达式之间用“;”或“,”隔开。



(1) 请使用 Print 语句输出“打字测试软件”程序中的打字速度。

(2) 请上机编写如下代码：

```
Private Sub Form_Click()
    Print "北京", "奥运"
    Print "北京"; "奥运"
End Sub
```

观察表达式用“,”和“;”分隔，输出结果有什么不同？

3. 注释语句

为了提高程序的可读性，通常需要在程序的适当位置加上必要的注释。添加注释是编程的一个好习惯。

在VB中添加注释语句的方法为在字符“'”后加注释内容或采用“Rem 注释内容”的形式。

如：

```
r = InputBox (" 输入 r: ") ' 输入半径 r
```

如果用 Rem 语句注释，应将 Rem 语句独立成行书写，不可像“'”一样直接放在其他语句后面，否则 VB 将提示错误。注释如下：

```
Rem 计算圆的周长
c = 2*PI*r
```

通常一条语句占一行。要在一行写多条语句，需要在每条语句之间用冒号进行分隔。

如果一条语句一行写不下，可用续行符(空格后面跟“_”)将语句分成多行。

各关键字之间，关键字与变量名、常量名之间要用空格。

4. 结束语句

语法格式：End

功能：强制程序结束运行。



(1) 编写一个程序，输入正方形的边长，输出正方形的面积和周长。

(2) 请运用已学的VB知识，编写一个奥运会奖牌总数统计程序。

(3) 编写一个袖珍计算器程序，程序参考界面如图 2-17 所示。



图 2-17 袖珍计算器

在“打字测试软件”程序中，我们是如何计算输入字符数的？如何得到打字所用时间的？

我们是利用VB提供的内部函数来解决这两个问题的。VB中函数的概念与数学中函数的概念没有本质区别。内部函数也称为标准函数，它们是VB系统为实现常用功能而提供的内部程序。在编写“打字测试软件”程序的代码时，我们输入了如下两行代码：

```
intTimeCost = DateDiff("s", dtmBeginTime, dtmEndTime)
intWordCount = Len(txtInput.Text)
```

“DateDiff("s", dtmBeginTime, dtmEndTime)”用来计算dtmBeginTime与dtmEndTime之间相隔的秒数，“Len(txtInput.Text)”则用来计算文本框txtInput中的字符数。

由此可见，使用内部函数可以减轻编程的工作量。VB中常用的函数有以下几种。

5. 与用户交互函数

VB提供InputBox()函数和MsgBox()函数用于人机交互。

(1) InputBox()函数。

InputBox()函数在前面已经用过，其语法格式为：

InputBox(提示[,标题][,默认值][,x坐标][,y坐标])

InputBox()函数的返回值是字符型数据，在对其进行数学运算前需要进行强制类型转换。

例如：`r = InputBox("请输入r", "输入圆半径")`

执行该语句，将显示如图2-18所示的对话框。



图2-18 InputBox对话框

(2) MsgBox()函数。

MsgBox()函数是VB程序设计中常用的交互函数，其功能非常强大，可以显示多种样式的消息对话框。

基本语法格式：`MsgBox(提示信息[,按钮][,标题])`

该函数的返回结果为一个数值，其中，“按钮”参数一般用VB的内部符号常量，这些符号常量有vbOkOnly、vbOKCancel、vbYesNo、vbYesNoCancel等，望文生义，看到常量的名称很容易理解它们的含义。

例如：`j = MsgBox("你输入的数据格式不正确，是否继续?", vbYesNo, "错误提示")`

该语句的执行结果如图2-19所示。

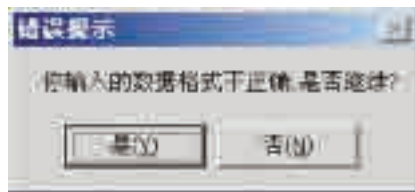


图2-19 MsgBox对话框

内部函数是由VB系统提供的，每个内部函数完成某个特定功能。在程序中使用函数称为调用函数，函数被调用后，一般都有一个确定的函数值，即返回值。

6. 数学函数

数学函数用于数学运算，包括三角函数、求平方根函数、求绝对值函数、对数函数、指数函数等，它们与相应的数学定义相同，如表 2-8 所示。

表 2-8 常用数学函数

函数	功能	实例	结果
Abs(x)	求 x 的绝对值	Abs(-4.6)	4.6
Sqr(x)	求 x 的平方根	Sqr(9)	3
Sin(x)	求 x 的正弦值	Sin(30*3.14/180)	0.499770102643102
Cos(x)	求 x 的余弦值	Cos(30*3.14/180)	0.866158094405463
Tan(x)	求 x 的正切值	Tan(60*3.14/180)	1.72992922008979
Atn(x)	求 x 的反正切值	4*Atn(1)	3.14159265358979
Exp(x)	求指数函数 e^x	Exp(2.0)	7.38905609893065
Log(x)	求 x 的自然对数值	Log(10)	2.30258509299405
Int(x)	取整数函数	Int(99.8)	99
Rnd()	产生 0~1 之间 (不包括 1) 的随机数	Rnd()	返回一个随机数

Randomize
()和 Rnd()往往结合使用,前者将随机数初始化,避免 Rnd()产生相同的随机数。

使用数学函数时需要注意以下几点。

(1) 三角函数中的参数应是弧度数,如数学中 $\sin 30^\circ$ 应写为:

$$\text{Sin}(30*3.14159/180)$$

(2) Rnd 函数返回 [0,1) (即包括 0,但不包括 1) 之间的一个随机数,它往往和 Randomize 结合使用。如果一个班有 54 名同学开联欢会,要编程抽取一名幸运同学,可以用以下代码求得:

```
Private Sub cmdOk_Click()
    Randomize      ' 初始化随机数生成器
    Label1.Caption = " 幸运同学的学号为: " & Int((Rnd * 54) + 1)
End Sub
```

7. 字符串函数

VB 系统提供了丰富的字符串处理函数,给编程带来了极大的方便,如表 2-9 所示。

表 2-9 常用字符串函数

函 数	功 能	实 例	结 果
Len(字符串)	求字符串的长度	Len("2008 奥运 ")	6
Left(字符串,截取长度)	截取字符串左边部分	Left("2008 奥运 ",4)	2008
Mid(字符串,起始截取位置[,截取长度])	从中间指定位置截取字符串	Mid("2008 奥运 ",3,2)	08
Right(字符串,长度)	截取字符串右边部分	Right("2008 奥运 ",2)	奥运
String(数值,字符)	生成指定长度与指定字符的字符串	String(3,"9")	999

在“打字测试软件”程序中，计算输入字符数时就使用了 Len 函数。

```
intWordCount = Len(txtInput.Text)
```

其含义是求出文本框中的字符数并赋给变量 intWordCount。

8. 日期函数

Date()函数返回系统的当前日期。

Time()函数返回系统的当前时间，如：dtmBeginTime = Time()，将系统当前时间赋给变量 dtmBeginTime。

以此类推，Day()、Month()、Year()函数分别返回指定日期的日数、月份数、年份数。



(1) 编程实现如图 2-20 所示计算器的功能，输入一个数，可以计算该数的平方数、绝对值、开方数。

(2) 家里办了个奶牛场，每月存入银行一部分钱，你能设计一个程序，计算指定月后，银行账户上有多少钱吗？假设计算公式为：

$$F = D(1 + I)[(1 + I)^M - 1] / I$$

其中：F—金额总数，D—每月存入数，I—银行利率，M—存的月数。
程序界面参考图 2-21。



图 2-20 计算器功能图



图 2-21 计算存款数额程序界面

9. 常用转换函数

在程序设计过程中,我们经常要将不同类型的数据进行转换,这时就需要使用转换函数。VB 中常用的转换函数如表 2-10 所示。

表 2-10 常用转换函数

函 数	功 能	实 例	结 果
Val(字符串)	将字符串转换为数值	2+Val("12")	14
Str(数值)	将数值转为字符串	Str(5)	"5"
Chr(字符代码)	求字符代码对应的字符	Chr(65)	"A"
Asc(字符)	求字符对应的字符代码	Asc("A")	65
CInt(表达式)	将表达式的结果转换为整数, 小数部分四舍五入	CInt(3.14259*2)	6



- (1) 通过“帮助”菜单启动 MSDN, 获取更多的常用函数的使用方法。
- (2) 在 VB 中输入以下代码:

```
Private Sub Form_Click()
    Dim X As String, Y As Integer
    Dim A As String, B As String, C As Integer
    X="1"
    Y=123
    A=Chr(Asc(X)+5)
    B=Str(Val(X)+5)
    C=Val(Str(Y)+"5")
    Print A,B,C
End Sub
```

运行程序, 指出这段程序中用了哪几类函数, 并说出程序输出这样结果的原因。

MSDN(Microsoft Developer Network)是开发人员的重要参考资料, 也是初学者的学习资源之一。它包含了许多编程技术信息, 其中有示例代码、文档、技术手册等。

2.3 程序的基本结构

任何正规的程序都可以分解为三种基本结构, 即顺序结构、选择结构和循环结构。牢固掌握这三种基本结构, 是学习程序设计的基本要求, 是编写出结构清晰、易读易懂的程序的前提。通过本节的学习, 你可以:

- 掌握顺序、选择、循环三种基本结构
- 能够使用三种基本结构编写简单的程序解决具体问题

2.3.1 顺序结构

如图 2-22 所示，有两个锥形瓶，1 号瓶里面装的是溴水，2 号瓶里面装的是水，如果我们要交换两个瓶子里的液体，该怎么办？

按照生活常识，我们可以用一个空瓶子作为“中转”来解决这个问题。同理，在程序设计中也可以使用这种方法来交换两个变量的值。

程序代码如下：

```
Private Sub cmdChang_Click()
    Dim a As Single, b As Single, c As Single
    a = Val(InputBox("输入 a"))
    b = Val(InputBox("输入 b"))
    Print "交换前：a = "; a, "b = "; b
    c = a
    a = b
    b = c
    MsgBox "交换后 a = " & a & "b = " & b, vbOKOnly, "交换结果"
End Sub
```

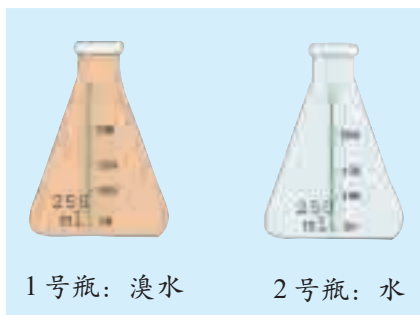


图 2-22 两瓶待交换的液体

参考算法：
输入 a 的值
输入 b 的值
 $c \leftarrow a$
 $a \leftarrow b$
 $b \leftarrow c$
输出 a 的值
输出 b 的值

运行程序，不难发现代码是按照由上到下的顺序一行一行地执行。程序执行过程中没有分支、没有重复，我们把这种结构称为顺序结构。顺序结构是一种最基本的程序结构。



(1) 分析上面的程序，完善图 2-23 中的流程图。

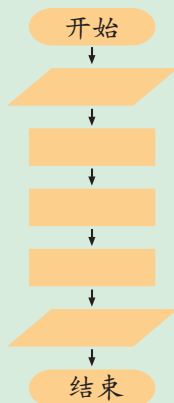


图 2-23 流程图

(2) 你知道自己已经出生了多少天、多少小时吗？请编写一段程序，输入自己的出生日期，输出答案。

2.3.2 选择结构

我国首位航天员杨利伟乘坐的“神舟”五号载人飞船，于北京时间2003年10月15日9时，在酒泉卫星发射中心成功升空，这标志着我国载人航天技术走向一个新的里程，如图2-24所示。由物理学知识可知：只有当飞船的速度达到第一宇宙速度(7.91千米/秒)时，才可以在地球引力作用下，绕地球近似做匀速圆周运动。

下面编写程序，实现以下功能：输入飞船速度，判断飞船是否能升空。流程图如图2-25所示。

像这种根据不同的条件判断来决定程序执行走向的结构，我们把它称为选择结构。那么，如何在VB中实现以上程序呢？



图 2-24 杨利伟出舱瞬间

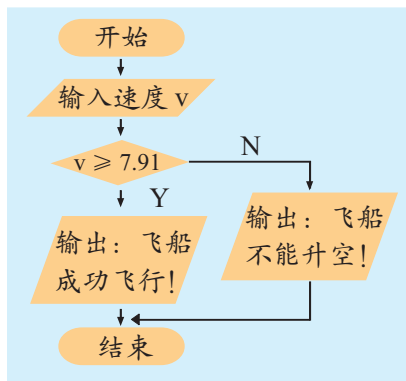


图 2-25 飞船运行状态流程图

1. 设计程序界面

Step1：在窗体中添加一个标签控件(将其名称改为 lblShow)、一个图像框控件(Image1)和一个命令按钮控件(将其名称改为 cmdOk)。

Step2：设置各控件的属性值，选择图像框控件 Image1，在属性窗口单击“Pic-ture”属性，选择“神舟号”图片。程序参考界面如图2-26所示。



图 2-26 设计程序界面

2. 编写程序代码

在VB中，选择结构可用If语句来实现，If语句有如下三种。

(1) 单分支选择结构，形式为：

```

If 条件 Then
    语句或语句组 '语句组是指有多条语句的情况
End If
  
```

(2) 双分支选择结构，形式为：

```

If 条件 Then
    语句或语句组 1
  
```

```
Else
    语句或语句组 2
End If
```

(3) 多分支结构，形式为：

```
If 条件 1 Then
    语句或语句组 1
ElseIf 条件 2 Then
    语句或语句组 2
.....
ElseIf 条件 n Then
    语句或语句组 n
Else
    语句或语句组 n+1
End If
```

本例中需使用双分支选择结构。完整的程序代码参考如下：

```
Private Sub cmdOk_Click()
    Dim v As Single
    v = Val(InputBox("输入速度(km/s)"))
    If v >= 7.91 Then
        lblShow.Caption = "飞船成功飞行！"
    Else
        lblShow.Caption = "飞船不能升空！"
    End If
End Sub
```

在书写 If 语句时往往利用 Tab 键把执行的语句组向右缩进，这样不仅美观，而且可读性强，便于调试程序。

3. 调试并运行程序

按 F5 键运行程序。

4. 保存程序文件

编写程序的过程中一定要注意及时保存文件。程序调试完毕后，一定要记住将所有相关文件保存在先前建立好的文件夹中。



- (1) 讨论顺序结构与选择结构的不同之处。
- (2) 编写程序，使计算机产生一个 1 到 100 之间的随机整数，然后判断这个数是奇数还是偶数，并输出该数及判断结果。
- (3) 在第一章中我们曾经分析过判断闰年的算法，请编写程序，输入一个公元年号，输出该年是否是闰年。
- (4) 在“神舟号”程序中，我们只判断了飞船成功飞行的条件。当飞船速度继续加大时，飞船将达到第二宇宙速度、第三宇宙速度，速度与运行状态之间的关系如表 2-11 所示。
试编写程序，输入不同的飞船速度，判断它的各种飞行状况。

策略提示:

可以用条件语句的嵌套,也可以用多分支选择语句,或多个条件的复合判断来实现。

“表达式列表”通常是一组常量值,每一个值确定一个分支;如果多个值共用一个分支,还可以用以下三种方法:

① 一组用逗号隔开的值,如 Case 1,3,5 表示 1, 3,5 均满足指定的条件。

② 表达式 1 To 表达式 2,如 Case 10 To 20 表示 10~20 都满足指定的条件。

③ Is 关系式,如程序中所示。

表 2-11 飞船的飞行状况

飞船速度(v) (单位: km/s)	飞行状况
$7.91 \leq v < 11.19$	飞船绕地球近似做匀速圆周运动
$11.19 \leq v \leq 16.67$	飞船离开地球的控制, 围绕太阳转
$v > 16.67$	飞船挣脱太阳引力飞出太阳系

当程序出现多分支选择时,用 If 语句实现比较烦琐,而使用多分支选择结构 Select Case 语句就方便多了。Select Case 的语法格式为:

```

Select Case 表达式
    Case 表达式列表 1
        语句组 1
    [Case 表达式列表 2
        语句组 2]
    .....
    [Case Else
        语句组 n]
End Select
  
```

例如,可用 Select Case 语句编写程序判断“神舟号”各种飞行状况,程序如下:

```

Private Sub cmdOk_Click()
    Dim v As Single
    v = Val(InputBox("输入速度 v(km/s)"))
    Select Case v
        Case Is > 16.67
            lblShow.Caption = "飞出太阳系"
        Case Is >= 11.19
            lblShow.Caption = "绕太阳运行"
        Case Is >= 7.91
            lblShow.Caption = "绕地球运行"
        Case Else
            lblShow.Caption = "数据输入错误!"
    End Select
End Sub
  
```



(1) 我们分别使用了 If 语句与 Select Case 语句完成了“神舟号”程序,请比较两段程序。思考在什么情况下适合用 If 语句,什么情况下适合用 Select Case 语句,

并与其他同学进行交流。

(2) 我们在数学课中学过的哪些算法适合使用 Select Case 语句实现, 选择其中的一个, 编程实现。

2.3.3 循环结构

假设你从今年开始, 第一个月为“希望工程”存入1元钱, 第二个月存入2元钱, 第三个月存入3元钱……依次类推, 两年时间你将为“希望工程”存入多少钱?

我们假设 n 为月数, $intSum$ 是一个累加变量, 且初始值为 0, 那么存入钱的累计数 $intSum$ 的计算方法是:

第 1 个月: $n=1, n \leq 24$ 条件成立, $intSum=intSum+n$, 即 $intSum=0+1$;

第 2 个月: $n=2, n \leq 24$ 条件成立, $intSum=intSum+n$, 即 $intSum=1+2$;

第 3 个月: $n=3, n \leq 24$ 条件成立, $intSum=intSum+n$, 即 $intSum=3+3$;

……

第 24 个月: $n=24, n \leq 24$ 条件成立, $intSum=intSum+n$, 即 $intSum=276+24$;

第 25 个月: $n=25, n \leq 24$ 条件不成立, 不再累计。

由此, 我们发现每次都重复执行 $intSum=intSum+n$ 这一语句, 像这种需要重复执行某一段代码的程序, 需要使用循环结构来实现。VB 循环结构常用的语句是 For...Next 语句和 Do While...Loop 语句等。那么如何编写程序解决这个问题呢?

“希望工程”是中国青少年发展基金会于 1989 年开始实施的一项社会公益项目, 是一项旨在集社会之力, 捐资助学, 保障贫困地区失学孩子受教育的基本权利的伟大工程。

1. 设计程序界面

新建一个标准 EXE 工程, 在 Form1 中添加一个标签控件 (将其名称改为 lblSum) 和一个命令按钮控件 (将其名称改为 cmdOk)。

程序界面可参考图 2-27 所示进行设计。

该程序的流程图如图 2-28 所示。



图 2-27 设计程序界面

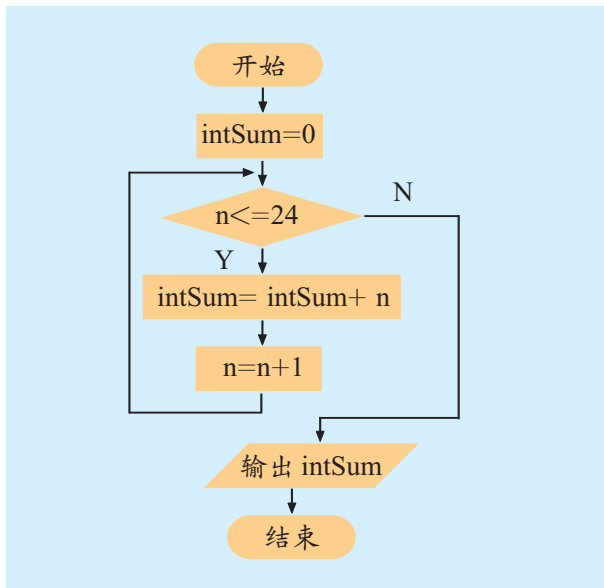


图 2-28 “希望工程储蓄”程序流程图

2. 编写程序代码

不要在 For 循环内部重新设置循环变量的值，否则会导致循环出错，甚至出现死循环。想一想，为什么？

双击 cmdOk 按钮，为 cmdOk 的 Click 事件过程编写如下代码：

```

Private Sub cmdOk_Click()
    Dim n As Integer, intSum As Integer
    intSum = 0
    For n = 1 To 24
        intSum = intSum + n
    Next n
    lblSum.Caption = "两年共存入：" & intSum & "元"
End Sub
  
```

上段代码中使用 For…Next 语句实现循环，变量 n 的初始值为 1，每执行一次循环，n 的值加 1(即循环步长为 1)，直到 n>24，退出循环。如果循环步长为 2，那么，For 语句为：For n = 1 To 24 Step 2。

3. 调试并运行程序

按 F5 键运行程序。

4. 保存程序文件

建立“希望工程储蓄”文件夹，保存该工程。

如果要计算多少月之后，“希望工程”储蓄才能多于 500 元，该怎么办呢？由于循环次数未知，使用 Do While…Loop 语句实现更合适。

将 lblSum 的 Caption 属性改为“多少月才能为希望工程存款超过 500 元？”，cmdOk 的 Click 事件过程的代码需要改写如下：


```

Private Sub Command1_Click()
    Dim n As Integer, intSum As Integer
    n = 0
    intSum = 0
    Do While intSum <= 500
        n = n + 1
        intSum = intSum + n
    Loop
    lblSum.Caption = n & "个月才能为希望工程存款超过500元。"
End Sub

```

从这段程序可以看出，Do While 语句先判断是否满足 While 后面的条件表达式，如果满足，程序反复执行 Do While 和 Loop 之间的语句，直到条件不再满足为止。

For...Next 语句和 Do While...Loop 语句在许多情况下是可以通用的，但是，For...Next 语句更适合用于循环次数容易确定的情况，而 Do While...Loop 语句的使用则更灵活些。



(1) 循环语句除了 Do While...Loop 形式外，还有 Do...Loop While 等多种形式。请你查阅有关资料，总结 VB 循环语句的所有形式，形成报告，与同学相互交流。

(2) ① 编程计算 $s=1+2+3+\dots+100$ 的值。

② 编程计算 1 到 100 之间所有偶数的和，要求 Step 值为 2。

③ 请用不同的循环语句实现：在程序运行过程中任意输入一个自然数 n ，求 1 到 n 之间所有偶数的和。

(3) 输出 100~200 之间不能被 3 整除的数。

单击窗体时在窗体中打印输出如图 2-29 所示的三角形图形，需要用到多重循环。我们采用两重循环来实现。外循环控制输出的行数，内循环控制每行输出的字符数。在进入内循环前，使用 Print Tab() 语句对 “*” 的输出位置进行定位并控制换行，完整的程序代码如下：

```

Private Sub Form_Click()
    For i = 1 To 7
        Print Tab(10 - i);
        For j = 1 To 2 * i - 1
            Print "*";
        Next j
    Next i
End Sub

```



图 2-29 三角形示意图

循环体内部还有循环语句的循环叫做多重循环，又叫多层循环或嵌套循环。



(1) 编写程序，在窗体中输出如图 2-30 所示的菱形图形。

Tab(n): 对输出进行精确的绝对列定位。

```

      *
     ***
    *****
   *********
  ***********
 *****
  *****
   *****
    *****
     *****
      *****
       *
    
```

图 2-30 菱形示意图

(2) 与同学讨论数学中哪些问题可以使用哪种循环语句解决，填写表 2-12，并尝试编程实现。

表 2-12 数学问题与循环语句

	For...Next	Do While...Loop	循环嵌套
求 1~100 之内的素数			
求两个整数的最大公约数、最小公倍数			

定时器 Timer 控件可以用来计时，它有一个“Timer”事件，如果启动了定时器，每隔一段固定时间，Timer 事件会被触发。定时器有两个重要属性。

① Interval: 设置 Timer 控件被触发的时间间隔，单位是毫秒，例如 Interval 设置为 1000，Timer 控件每隔 1 秒钟被触发一次。






② Enabled: 它的设置值可以是 True 或 False。只有在它的值是 True 时定时器才会工作。



请你为学校的春季运动会设计一个倒计时程序，要求如下：

- ① 程序显示“现在距我校春季运动会开幕还有：××××天××小时××分钟”；
- ② 程序显示当前时间，格式为“现在是：××××年××月××日上午(下午)××时××分”；
- ③ 程序显示内容能每分钟自动刷新一次(提示：可以使用 Timer 控件)；
- ④ 可以通过改变窗体的背景颜色、添加图片等手段设计出符合自己个性的界面。

第三章 算法的程序实现

-  用解析法解决问题
-  用穷举法解决问题
-  在数组中查找数据
-  对数据进行排序
-  用递归法解决问题

算法是程序设计的灵魂,在掌握程序设计的基本知识后,本章侧重于运用算法解决实际问题。你将学会如何运用已有知识,设计合理的算法并编程实现。在学习的过程中,你需要进一步理解程序设计的基本知识,能够做到独立编程,解决比较复杂的问题。

3.1 用解析法解决问题

大自然中包含了丰富多彩的图形,计算机也能模拟出这些漂亮的图形。通过本节的学习,我们将用解析法绘制一颗“钻石”。通过本节的学习,你可以:

- 了解解析法,学会用解析法分析问题、解决问题
- 学会编写程序实现解析法

3.1.1 解析法分析

1. 什么是解析法

高级程序设计语言提供的内部函数的内部实现大多也是用解析法。

“解析”这个词对我们并不陌生,我们经常使用“解析表达式”、“解析几何”、“解析法”等。所谓“解析”就是剖析、深入分析的意思。

解析法就是在分析具体问题的基础上,抽取出一个数学模型,这个数学模型能用若干个解析表达式表示出来,解决了这些表达式,问题也就得以解决。解析法是程序设计中常用的算法之一。

在日常生活与学习中,我们经常使用解析法解决问题。比如,如果乘火车以100千米/小时的速度从北京去上海(全程约1400千米),那么,所需要的时间是: $1400/100=14$ 小时。这种方法就是解析法。凡能用解析法求解的问题都可以通过定量分析,并能用解析表达式进行描述。数学、物理等学科中的许多问题也都是用解析法解决的。

下面我们就用解析法编写程序绘制“钻石”。



图 3-1 钻石图案

2. 问题分析

这颗“钻石”是由点和线构成的,如图3-1所示。图形四周的点位于一个圆周上,点与点之间都有一条线段相连。因此,只要求出各点的位置,绘制出各点间的线段就可以绘出这个图形了。

如何求各点的位置呢?

首先,建立如图3-2所示的坐标系,坐标原点位于图形的中心点上。假设我们在圆

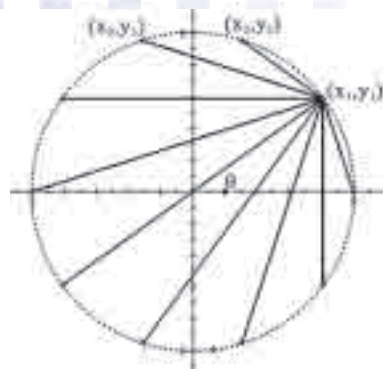


图 3-2 数学分析

上平均取N个点，将圆等分为N份，利用数学知识，不难写出各点的坐标。

第1个顶点的坐标 (x_1, y_1) 可表示为：

$$x_1 = r \cos(\theta), y_1 = r \sin(\theta)$$

其中， $\theta = 360^\circ / N$ ；r为圆的半径。

第2个顶点的坐标 (x_2, y_2) 可表示为：

$$x_2 = r \cos(2\theta), y_2 = r \sin(2\theta)$$

以此类推，可以计算出所有顶点的坐标，这样，就可以用两层循环语句，画出从每个点出发到其他各点的线段。

算法的伪代码表示如下：

```

For i = 1 To 总的点数
    求出第 i 个点的坐标(rcos(iθ), rsin(iθ))
    For j = 1 To 总的点数
        If i 点和 j 点不是同一个点 Then
            求 j 点的坐标(rcos(jθ), rsin(jθ))
            画出 i 点到 j 点的线段
        End If
    Next j
Next i
    
```

从上面的分析过程可以看出，用解析法解决问题的关键就是要找出描述求解问题的解析表达式。



(1) 某超市规定，顾客购买同种商品10件以上(含10件)可享受批发价。请设计一个收款机程序，输入顾客所购商品的零售价、批发价、购买数量及付款数，程序能计算出顾客的应付款及需要找回的零钱。请设计解析法，并用流程图或伪代码描述。

(2) 某县政府为了开发山区农业，需要研究本区的山地气候。已有的研究成果表明，海拔每升高100米，年平均气温降低 0.5°C 。试根据这一规律编写程序，输入山下地表温度及相对海拔高度，输出相应高地的气温。请用解析法设计算法。

(3) 俗话说“失之毫厘，谬之千里！”，果真如此吗？二元一次方程组

$$\begin{cases} 35.26x + 14.96y = 28.35 \\ 187.2x + 79.43y = 83.29 \end{cases} \text{ 和 } \begin{cases} 35.26x + 14.96y = 28.35 \\ 187.3x + 79.43y = 83.29 \end{cases}$$

虽然只是其中的一个系数相差了0.1，其结果却大相径庭！试设计并描述求解二元一次方程组的算法。

3.1.2 编程实现

我们需要使用VB提供的三角函数和画线命令绘制“钻石”。首先，新建一个标准EXE工程，对象属性设置参见表3-1。

表 3-1 对象属性设置

对象	属性	设置值
工程1	名称	计算机绘图
Form1	名称	frmDraw
	Caption	钻石图形
	StartPosition	2-屏幕中心

下面我们在窗体的Click事件过程中编写代码,当程序运行后,单击窗体,程序开始执行。下面是根据前面的伪代码编写的参考程序。

VB的屏幕坐标系和数学中的坐标系不同,它的原点不在窗体中心,而在窗体工作区的左上角,但可以通过窗体的Scale方法更改:

窗体.Scale
(左上角坐标)–
(右下角坐标)

用VB绘制
线段:

窗体.Line
(起点坐标)–
(终点坐标),颜色。

也可以直接
通过画线命令
实现:

Line(起点
坐标)–(终点坐
标),颜色。

```
Private Sub Form_Click()
    Const PI = 3.14159265 ' 定义常量PI
    ' 以下定义变量
    Dim i As Integer, j As Integer
    Dim sngX1 As Single, sngY1 As Single
    Dim sngX2 As Single, sngY2 As Single
    Dim sngAngle As Single ' 表示等分的单位弧度
    Dim sngRadius As Single ' 表示点所在圆周的半径
    Dim intNodes As Integer ' 表示总点数
    frmDraw.Scale (-1.5, -1.5)-(1.5, 1.5) ' 设置屏幕坐标系
    Cls ' 清除窗体原有显示的内容
    ' 下面对变量进行初始化
    sngRadius = 1 ' 设半径为1个单位
    intNodes = 15 ' 设绘图的点数
    sngAngle = 2 * PI / intNodes ' 等分的单位弧度
    ' 以下程序画出从一个点出发到其他各点间的线段
    For i = 1 To intNodes
        ' 求出点i的坐标
        sngX1 = sngRadius * Cos(i * sngAngle)
        sngY1 = sngRadius * Sin(i * sngAngle)
        For j = 1 To intNodes
            ' 如果i不等于j表示不是同一个点
            If i <> j Then
                ' 求线段的另一个点j的坐标
                sngX2 = sngRadius * Cos(j * sngAngle)
                sngY2 = sngRadius * Sin(j * sngAngle)
                ' 画出从i点到j点的线段,颜色为蓝色
                frmDraw.Line (sngX1, sngY1)-(sngX2, sngY2), vbBlue
            End If
        Next
    Next
Next
End Sub
```

程序通过窗体的 Scale 方法重新设置坐标系,使坐标原点位于窗体的中心,则窗体左上角的坐标变为(-1.5, -1.5),右下角的坐标变为(1.5, 1.5);然后用 Cls 方法清屏;接着对变量进行初始化,设总点数为 15,各点距坐标原点的直线距离为 1 个单位,这些点平分圆周,相邻点之间的夹角为 $2\pi/15$ (弧度);最后,在外层循环语句中求出一个点 i 的坐标(sngX1, sngY1),在内层循环语句中求出其他对应各点 j 的坐标(sngX2, sngY2),并用窗体的画线方法画出点 i 到点 j 间的线段。



- (1) 选择 3.1.1 节中的一个实践,根据自己的算法设计,编写出 VB 程序。
- (2) 科技馆的门票价格为每人 10 元,但对团体票 (8 人以上)实行 8 折优惠;持学生证可以半价购票;军人免费。请为科技馆设计一个程序实现上述功能。
- (3) 仔细比较本节所用的伪代码和源程序代码,讨论如何根据伪代码编写程序代码。

3.2 用穷举法解决问题

用解析法解决问题具有高效、快捷的特点,但是,解析法也有“束手无策”的时候,有些问题即便可以用解析法,但求解过程和步骤很复杂。为了解决这类问题,人们往往利用计算机高速运算的特性,采取一一尝试的办法,最终得出求解结果。本节我们将以“明日之星英语演讲大赛”活动为例,通过参赛选手小组划分方案的设计,引领同学们设计算法,编写程序。通过本节的学习,你可以:

- 了解穷举法的特点,掌握用穷举法设计算法的基本要求
- 学会编写程序实现穷举法

3.2.1 穷举法分析

1. 什么是穷举法

穷举法也叫枚举法、列举法,它将求解对象一一列举出来,然后逐一加以分析、处理,并验证结果是否满足给定的条件,穷举完所有对象,问题将最终得以解决。

请看下面的问题：

“明日之星英语演讲大赛”再过几天就要开始了，全校共有 M (328) 名中学生参赛。比赛规则规定先进行小组比赛，然后取小组前三名进行决赛。因此，组委会需要对参赛选手进行分组。

请按下面的分组规则设计算法，求解可能的分组方案供组委会参考。

规则一：每组最少 N_1 (10)人，最多 N_2 (30)人；

规则二：如果不能平均分组，则各小组间人数之差不得多于1人。

2. 问题分析

用VB打开光盘中的“大赛分组.vbp”文件，然后运行程序，观察程序的分组情况。

用解析法来解决这个分组问题，不易找到解析表达式，而用穷举法就容易多了。

首先，要确定穷举对象的范围。通常，我们可以按小组人数穷举，也可以按小组数来穷举。如果按小组人数穷举，由于各组最后分得的人数可能不同，这样设计算法比较麻烦；如果按小组数穷举则相对比较方便。

如何确定小组数的范围呢？假定每组以 N_1 (10)人试分配，可得最多组数 Max ；若以 N_2 (30)人试分配，可得最少组数 Min 。

最大组数 Max 的确定：以每组 N_1 试分配 M 人，若没有余数，则每组都是 N_1 人；若有余数，将余数分散到其他组，有小组实际分得的人数将为 N_1+1 。因此，无论有无余数，分配的结果都满足指定的两条规则，由此可以确定 $Max=M \setminus N_1$ 。

最小组数 Min 的确定：每组以 N_2 人试分配，若没有余数，则各组都是 N_2 人，满足指定的两条规则；若有余数，将余数分散后将会有小组最后分得 N_2+1 人，至少有一个小组人数将超出规则一中规定的小组人数的最大值(N_2)，因此，这种分配方案无效。所以，每组应以 N_2-1 人再进行试分配，这样就能满足指定的两条规则。由此可以确定 Min 的值：

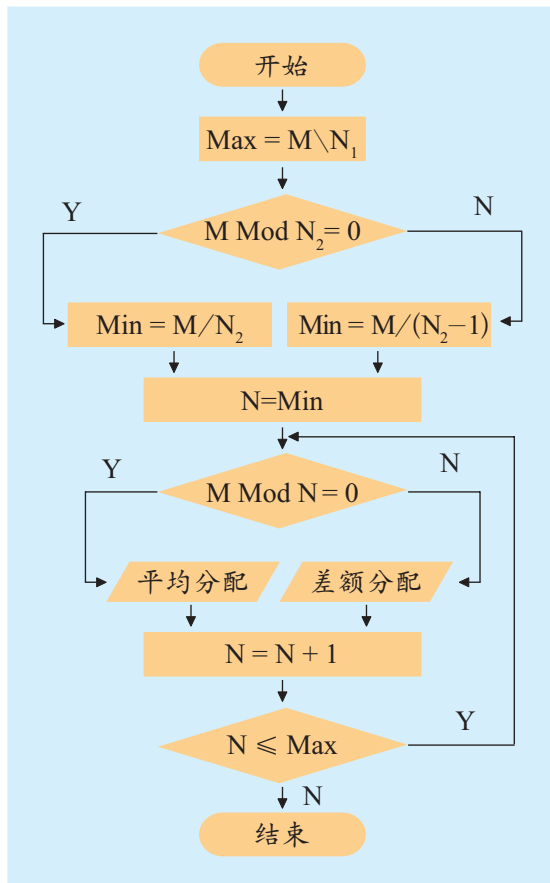


图 3-3 大赛分组的程序流程图

若 M 除以 N_2 没有余数, 则 $\text{Min} = M/N_2$

若 M 除以 N_2 有余数, 则 $\text{Min} = M/(N_2 - 1)$

小组穷举范围确定后, 接下来按组数(N)穷举从 Min 到 Max 之间所有可能的分配方案。

进行分组时, 先平均试分配人数, 如果余数(R)为 0, 说明能平均分配, 直接输出分配结果; 否则, 将余数分散到其他组中, 这样, 将会有 $N - R$ 组 M/N 人, R 组 $M/N + 1$ 人。其分配算法的流程图如图 3-3 所示。

从上面的分析可以看出, 设计穷举法的关键是确定穷举对象的范围。穷举对象不能有遗漏, 否则穷举不全面; 确定范围后, 只要将所有对象逐一列举, 分别处理, 就能最终解决问题。

适用穷举法解决问题的例子很多, 例如, 1.2.1 节中的“水仙花数问题”, 以及我们熟知的“鸡兔同笼问题”、“百鸡百钱问题”等。这类问题的共同点是: 求解对象应该是有限的, 而且可以按某种规则列举对象。



(1) 根据上面的分析, 结合图 3-3 所示的程序流程图写出“明日之星英语演讲大赛”分组算法的伪代码, 讨论穷举法的特点。

(2) 碱基排列问题: DNA 是非常重要的遗传物质, DNA 中的碱基有 A、G、C、T 共四种, 分别是 A:腺嘌呤; G:鸟嘌呤; C:胞嘧啶; T:胸腺嘧啶。科学研究发现, 这四个“字母”的不同排列, 构成了不同的遗传密码。试用穷举法列举出所有可能的排列, 并用流程图或伪代码描述算法。



DNA 模型

3.2.2 编程实现

在 VB 中新建一个标准 EXE 工程, 对象属性设置参见表 3-2。

表 3-2 对象属性设置

对象	属性	设置值
工程 1	名称	大赛分组
Form1	名称	frmGroup
	Caption	预赛分组
	StartPosition	2-屏幕中心

以下是程序的参考源代码:

```

Private Sub Form_Click()
    Dim intTotal As Integer
    Dim intGroup As Integer
    Dim intRemainder As Integer
    Dim intMember As Integer
    Dim intMin As Integer, intMax As Integer
    Cls
    intTotal = 328
    ' 确定可能的分数组
    If intTotal Mod 30 <> 0 Then
        intMin = intTotal \ (30 - 1)
    Else
        intMin = intTotal \ 30
    End If
    intMax = intTotal \ 10
    Print "总组数", "分组情况"           ' 打印列表头
    ' 以下穷举从 intMin 到 intMax 的分组情况
    For intGroup = intMin To intMax
        intMember = intTotal \ intGroup
        intRemainder = intTotal Mod intGroup
        If intRemainder = 0 Then
            ' 正好能平均分
            Print intGroup, "平均分"; intMember; "人"
        Else
            ' 如果有余数, 将余下的分散到其他组中
            Print intGroup, "其中:"; intGroup - intRemainder; _
                "组"; intMember; "人,";
            Print intRemainder; "组"; intMember + 1; "人"
        End If
    Next
End Sub

```

程序首先确定可能的分组范围 `intMin` 和 `intMax`, 然后穷举从 `intMin` 到 `intMax` 范围的分组情况。

在分组时, 先以总人数 `intTotal` 除以小组数 `intGroup` 求得试分人数 (`intMember`), 然后求出试分后的余数 `intRemainder`。如果余数为 0, 表示能平均分, 直接输出平均数; 如果余数不为 0, 则将余下的人数分散到其他组中; 这样, 将会有 `intGroup - intRemainder` 组为 `intMember` 人, 有 `intRemainder` 组多分得一人, 为 `intMember + 1` 人, 最后, 输出分组结果。

程序运行结果如图 3-4 所示。



图 3-4 大赛分组程序运行结果



(1) 用穷举法编写程序，找出所有的“水仙花数”。

(2) 古希腊数学家丢番图的墓志铭上有如下记载:下面的数字可以告诉你他的一生有多长:他生命的六分之一是愉快的童年;再过了生命的十二分之一,面颊上长了细细的胡须;又过了一生的七分之一,他结了婚。婚后五年,他有了第一个孩子,感到很幸福。但命运给这孩子的光辉灿烂的生命只有他父亲的一半。儿子死后,他在深切的悲痛中活了四年就告别了尘世。请用穷举法编写程序求出丢番图的年龄。

(3) 百鸡百钱问题:相传我国南北朝时,京城有个卖鸡的张姓老汉,他有一个儿子非常聪明,尤其擅长算术,到十二三岁时已是远近闻名的“小神童”了。当朝宰相听说后想试探个究竟,于是派仆人到张老汉的店里打听鸡的价钱,张老汉告知“公鸡五文钱一只,母鸡三文一只,小鸡一文三只”。于是,仆人给他一百文钱,要求公鸡、母鸡、小鸡都要,数量不多不少正好一百只,命他次日送到府上。这可难为了张老汉,他怎么凑也凑不够这个数,只好问儿子。“小神童”不慌不忙,掐指一算就给出了答案,第二天照数送到宰相府。宰相见难不倒“小神童”,又让仆人给张老汉一百文钱,要求再买一百只鸡,搭配方法不能和上次一样。结果“小神童”又很快给出了答案,宰相暗暗称奇,想最后再试一次,谁知还是没有难倒“小神童”。

这个故事就是我国古代数学名著《张邱建算经》里的百鸡百钱问题。请用穷举法编写程序求解所有的组合方法。

3.3 在数组中查找数据

在日常生活中,我们经常需要查找一些信息,如查询列车时刻、查询电话号码等;在程序设计中,我们也经常需要进行数据查找。通过本节的学习,你可以:

- 理解数组的概念,学会用简单的数组处理数据
- 掌握顺序查找算法,学会用顺序查找法编写简单的查找程序
- 了解二分法查找算法,学会编写二分法查找程序

让我们先在VB中体验一下数据的查找:在“视图”菜单中单击“立即窗口”命令,输入语句:

```
?InStr("算法与程序设计","程序")
```

按下回车键后观察运行结果,然后指出函数返回值与两个字符串的关系。如果让你设计这段程序,你能做到吗?

设计查找算法时,往往要考虑算法实现起来是否方便、快捷,因此,查找算法中经常会用到数组。

InStr函数的作用:指定一个字符串在另一个字符串中最先出现的位置。例如,InStr("算法与程序设计","程序")的返回值是4。

3.3.1 什么是数组

如果要在程序中表示某位同学的语文、数学、英语、物理、化学五门课的成绩，我们很容易想到定义 5 个变量存放这 5 门功课的成绩：

```
sngChinese, sngMath, sngEnglish, sngPhysics, sngChemistry
```

当我们处理的不是一位同学的成绩，而是 50 名或更多人的成绩时，如果采用这种单个变量的处理方法将非常烦琐，假如借助数组(Array)处理就方便多了。

在 VB 中，数组起始编号默认为 0，但可以通过如下形式改变：

```
Dim 数组 (下界 To 上界) As 数据类型
```

数组不是一种基本数据类型，而是一组相同类型的变量集合，集合中的每个元素都是独立的变量，它们能够通过数组序号被引用。变量与数组的关系如图 3-5 所示，变量类似于一个个独立的柜子，而数组则类似于一组柜子，它们共同构成一个整体，可以用统一的名称进行管理。

在 VB 中定义数组的格式为：

```
Dim 数组名(常量) As 数据类型
```

例如，用数组 aryChinese 来存储 50 名同学的语文成绩，可以定义为：

```
Dim aryChinese(49) As Single
```

这个数组共有 50 个元素，分别为 aryChinese(0), aryChinese(1), ..., aryChinese(49)，数据类型为单精度。

从上面的定义可以发现，数组中的数据在逻辑上应该是相互关联的。

数组的赋值和一般变量的赋值类似，下面这段代码可以将数组中各元素重新置 0：

```
For i = 0 To 49
    aryChinese(i) = 0
Next
```

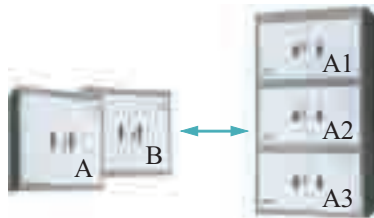


图 3-5 单柜与组合柜

程序设计中，常常通过循环语句来处理数组中的数据。



- (1) 讨论数组的特点，比较简单变量与数组的差异。
- (2) 通过使用联机帮助或上网搜索等，探索 VB 中有关数组的更多知识。

3.3.2 顺序查找

顺序查找是查找算法中简单、易行的算法。进行查找时，一般从数据的第一个元素开始，按照数据的顺序查找指定的关键值；如果被查数据和该关键值相匹配，则表示找到；如果所有数据与关键值都不匹配，则表示

被查数据中不存在该关键值，查找失败。下面我们用“简单替换密码”程序说明如何设计顺序查找算法，将所给的一段英文字符进行加密、解密处理。

1. 问题描述

简单替换密码就是把字符串中的每个字符对应地替换为另一个字符，重新进行编码，从而形成密文；解密时再按对应关系将加密字符还原。例如，字符串“Book”加密后显示为“Γ κ κ ζ”。

2. 问题分析

为了处理方便，我们分别用两个数组来存储源码字符表和密码字符表。

加密时，从待加密字符串中取出一个字符，顺序查找它在源码字符表中的位置，找到后用密码字符表中对应位置的字符代替；若找不到，则仍用原来的字符。如此循环，将整个字符串进行替换处理。

解密过程是加密的逆过程，其处理方法和加密方法相同。

那么，如何构建字符表呢？如果不处理标点符号的话，英文字母为26个大写字母和26个小写字母，加上空格，共53个，相应的密码表中也需要53个字符，这里不妨以希腊字母代替。希腊字母大、小写加起来只有44个，为了能和英文字母相对应，还应再加9个其他字符。为了简便起见，我们从希腊字母“A”开始顺次取53个字符，到“φ”结束，其中包含了部分不可见字符，其对应关系如图3-6所示。当然，你也可以用其他字符构建密码表，如汉字、俄文字母等。

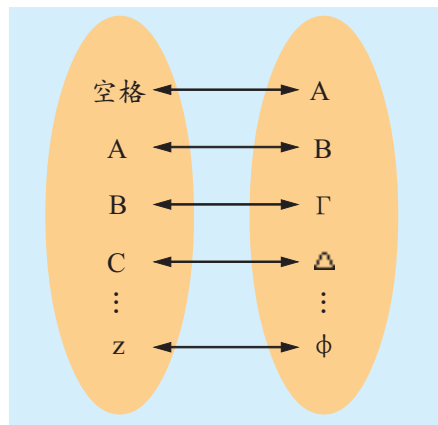


图 3-6 源码与密码

3. 编程实现

在VB中新建一个标准EXE工程，添加一个标签控件、两个命令按钮控件和三个文本框控件，按照表3-3所示内容设置各控件的属性，效果如图3-7所示。



图 3-7 程序设计界面

表 3-3 设置控件属性

对象	属性	设置值
工程 1	(名称)	字符串加解密
Form1	(名称)	frmEncrypt
	Caption	英文字母加解密
	StartPosition	2-屏幕中心
Label1	Caption	待加密字符串
	BackStyle	0-Transparent
	AutoSize	True
CommandButton1	(名称)	cmdEncrypt
	Caption	加密
CommandButton2	(名称)	cmdDecrypt
	Caption	解密
Text1	(名称)	txtText
	Appearance	0-Flat
	Text	空
Text2	(名称)	txtEncrypt
	Appearance	0-Flat
	Text	空
Text3	(名称)	txtDecrypt
	Appearance	0-Flat
	Text	空

VB 中的变量分全局变量和局部变量两大类。局部变量又分为模块级变量和过程级变量。每种变量的作用范围不同。

代码编写分四个部分：常量和数组定义、数组初始化、字符串加密处理和字符串解密处理。

(1) 常量和数组定义：定义常量 CHARNUM 用来约束两个字符表的字符总数；定义数组 aryCode()和 aryEncode()分别用于存储两个字符表。为了能在其他过程中访问它们，需在窗体代码的通用部分声明，参考代码如下：

```
' 定义窗体模块级常量和变量
Const CHARNUM = 52
' 定义源码字符表和密码字符表数组
Dim aryCode(CHARNUM) As String ' 下标从 0 开始
Dim aryEncode(CHARNUM) As String
```

(2) 数组初始化：字符表的构建放在窗体的 Load 事件过程中。源码字符表中第一个字符放空格；查 ASCII 码表可知，英文字符的编码分两段，从 65 到 90 为“A”到“Z”；从 97 到 122 为“a”到“z”，因此，可以通过两

段循环语句，用 Chr()函数分别读入源码字符表。

在“立即窗口”中输入“? Asc(" A ")”可知，希腊字母“A”的ASCII码为-22879，由此开始用 Chr()函数读取 53 个字符便构成密码字符表。程序参考代码如下：

```
Private Sub Form_Load()
    Dim i As Integer
    ' 构建源码字符表
    aryCode(0) = Chr(32) ' 第一个字符为空格
    ' 将字母 A~Z 读入数组中
    For i = 1 To 26
        aryCode(i) = Chr(64 + i)
    Next
    ' 将字母 a~z 读入数组中
    For i = 27 To CHARNUM
        aryCode(i) = Chr(97+(i-27))
    Next
    ' 用希腊字母构建密码字符表
    For i = 0 To CHARNUM
        aryEncode(i) = Chr(-22879 + i)
    Next
End Sub
```

Asc("字符")和Chr(数值)互为逆函数，前者求一个字符的内部编码；后者求指定编码的字符。

(3) 字符串加密处理：这部分代码放在 cmdEncrypt 的 Click 事件过程中。通过外层循环语句，从 txtText 文本框中取出一个字符，然后，用内层循环顺序查找它在数组 aryCode 中的位置。如果找到，就用 aryEncode 中相应位置的字符代替，否则仍使用原字符。程序的参考代码如下：

```
Private Sub cmdEncrypt_Click()
    ' 定义变量
    Dim i As Integer, j As Integer
    Dim strChar As String
    ' 将加密文本框设为空
    txtEncrypt.Text = ""
    ' 依次取出待加密字符串中的字符，找出它在源码字符表中的位置
    For i = 1 To Len(txtText.Text)
        ' 在待加密字符串中取出一个字符
        strChar = Mid(txtText.Text, i, 1)
        ' 查找该字符在源码字符表中的位置
        For j = 0 To CHARNUM
            ' 如果找到就退出循环
            If strChar = aryCode(j) Then
                Exit For
            End If
        Next
    Next
End Sub
```

输入技巧：
希腊字母可以通过输入法语言栏的软键盘输入。


```

End If
Next
' 根据 j 判断查找结果, 若 j <= CHARNUM 表示已找到
If j <= CHARNUM Then
' 用密码字符表中相应字符替换
strChar = aryEncode(j)
End If
' 构造密码字符串
txtEncrypt.Text = txtEncrypt.Text + strChar
Next
End Sub
    
```

(4) 字符串解密处理: 这个过程和加密处理过程相似, 它是从密码字符串中取出一个字符, 查找该字符在密码字符表中的位置, 找到后用源码字符表中相应的字符替换, 否则仍使用原字符。程序运行结果如图 3-8 所示。



图 3-8 “英文字母加解密”程序运行结果



(1) 上例中, 字符串解密算法和字符串加密算法是相同的, 请参考加密程序编写解密程序。

(2) “明日之星英语演讲大赛”计分问题: 输入选手号和 10 名评委的评分, 通过顺序查找, 去掉一个最高分和一个最低分, 求出选手的最后平均得分, 输出结果。界面可参考图 3-9, 注意网格是由 10 个文本框组成的。



图 3-9 “明日之星”计分界面

3.3.3 二分法查找

如果查找的数据较多或频繁进行查找, 顺序查找法效率会比较低, 使用二分法查找则可以提高查找的效率。

二分法查找也叫折半查找，它要求被查数据是有序的，否则无法使用二分法查找。

查找时，设置一个上界和一个下界，然后取上下界间的中间元素与指定的关键值比对，如图 3-10 所示。

如果相符，表示找到，查找结束；

如果不相符，再判断关键值落在左半部还是右半部；

如果在左半部，则舍弃右半部，保持下界位置不变，将上界设在中间元素的前一个位置，重新查找；

如果在右半部，舍弃左半部，保持上界位置不变，将下界设在中间元素的后一个位置，重新查找。

如此反复进行，若下界大于上界，表明没有元素和关键值相匹配，查找失败。

二分法查找流程图如图 3-11 所示。

下面将上文的字符串加密处理程序中的顺序查找算法改为二分法查找算法。

(1) 定义 `intLow`、`intHigh` 和 `intMid` 三个整型变量分别表示下界、上界和中间元素的下标。

(2) 通过外层循环语句依次取出待加密字符串中的一个字符。

(3) 初始查找时，设置下界 (`intLow`) 为数组 (`aryCode`) 的第 1 个元素，上界 (`intHigh`) 为数组的最后一个元素，并取得中间元素的下标 (`intMid`)。

(4) 如果中间元素 (`intMid`) 和关键值 (`strChar`) 相符，则退出内层循环，本次查找结束；如果不相符，判断关键值 (`strChar`) 和中间元素 (`intMid`) 的大小，重设 `intLow` 或 `intHigh` 及 `intMid` 的值。

(5) 如果 $\text{intLow} \leq \text{intHigh}$ ，返回上一步；否则表示没找到，查找结束。

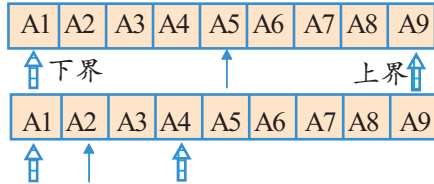


图 3-10 二分法查找示意图

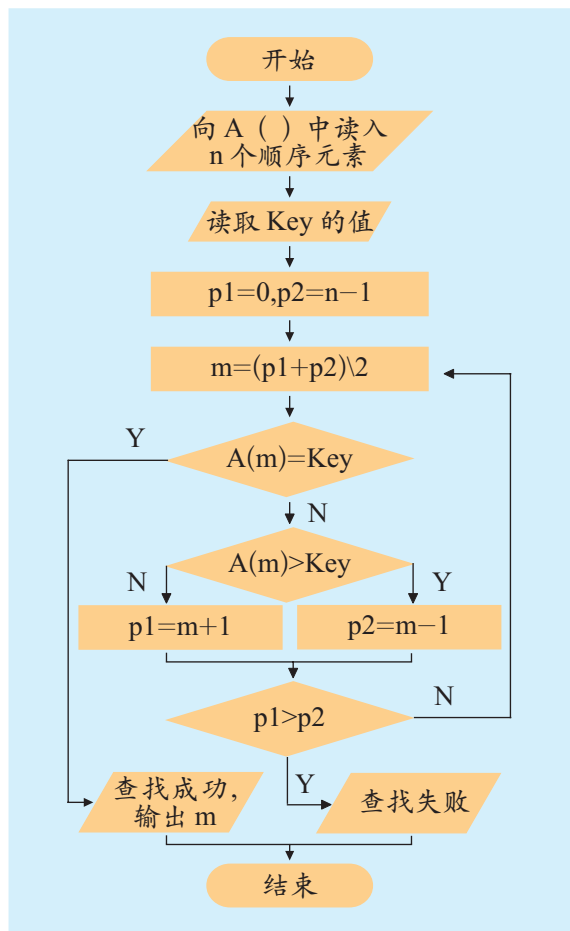


图 3-11 二分法查找流程图

程序的参考代码如下：

Do Until<条件表达式>/Loop:
当条件表达式成立时，循环结束。

Do While<条件表达式>/Loop:
当条件表达式不成立时，循环结束。

```
Private Sub cmdEncrypt_Click()
    Dim i As Integer
    Dim strChar As String ' 定义关键值变量
    ' 定义下界、中间元素和上界变量
    Dim intLow As Integer, intMid As Integer, intHigh As Integer
    txtEncrypt.Text = ""
    For i = 1 To Len(txtText.Text)
        ' 变量初始化
        intLow = 0
        intHigh = CHARNUM ' CHARNUM 为定义的常量
        ' 取出一个字符
        strChar = Mid(txtText.Text, i, 1)
        Do Until intLow > intHigh
            ' 设置中间值
            intMid = (intLow + intHigh) \ 2
            If aryCode(intMid) = strChar Then
                ' 目标和关键值相匹配，退出循环
                Exit Do
            Else
                ' 目标和关键值不匹配，重新设置下界或上界
                If aryCode(intMid) < strChar Then
                    ' 落在右半部分
                    intLow = intMid + 1
                Else
                    ' 落在左半部分
                    intHigh = intMid - 1
                End If
            End If
        Loop
        ' 构造加密字符串
        If intLow <= intHigh Then
            txtEncrypt.Text = txtEncrypt.Text & aryEncode(intMid)
        Else
            txtEncrypt.Text = txtEncrypt.Text & strChar
        End If
    Next
End Sub
```

查找算法是程序设计中比较成熟的算法，人们在不断的探索过程中，

积累了许多经典的算法，本节只选取了顺序查找和二分法查找两种比较简单的算法，有兴趣的同学可以进一步探索。



(1) 讨论采用二分法查找为什么要求被查找的数据是有序的。

(2) 用二分法查找编写解密程序，并与顺序查找算法进行对比，指出它的高效体现在哪里。

(3) 用二分查找算法编一个计算机猜数小游戏，界面设计可以参考图 3-12。

游戏时，首先指定一个猜数范围，并确定一个需要计算机“猜”的数，默记心中，然后让计算机“猜”这个数；如果猜中，告诉计算机“正确”，否则告诉它“大了”还是“小了”。

为了增加趣味性，可以同时找个同学和计算机比赛，看谁猜得快。通过人机比赛体会二分法查找算法的高效性。（源程序见光盘）



图 3-12 “二分法猜数”的界面

注意：英文字符的大小是按ASCII码排列的，可以直接进行比较；汉字和其他字符不能直接进行比较，但可以通过 Asc() 函数进行对比。

3.4 对数据进行排序

“明日之星英语演讲大赛”评分出来了，需要选出每组前三名进入决赛。我们需要设计程序，对选手成绩进行排序。通过本节的学习，你可以：

- 理解什么是排序算法
- 学会用冒泡法对数据进行排序

3.4.1 冒泡法排序算法

所谓排序，就是指将一个无序序列排列成一个有序序列的过程。

计算机编程中，排序算法的使用非常频繁。实践中，人们设计出了许多好的算法，如交换排序、插入排序、选择排序、归并排序、基数排序等，各种算法有其自身的特点与适用范围。

“冒泡法排序”也叫“起泡法排序”，是一种比较简单、易懂的交换排序方法，它通过将相邻元素进行比较和交换，逐步将一个无序序列排列为一

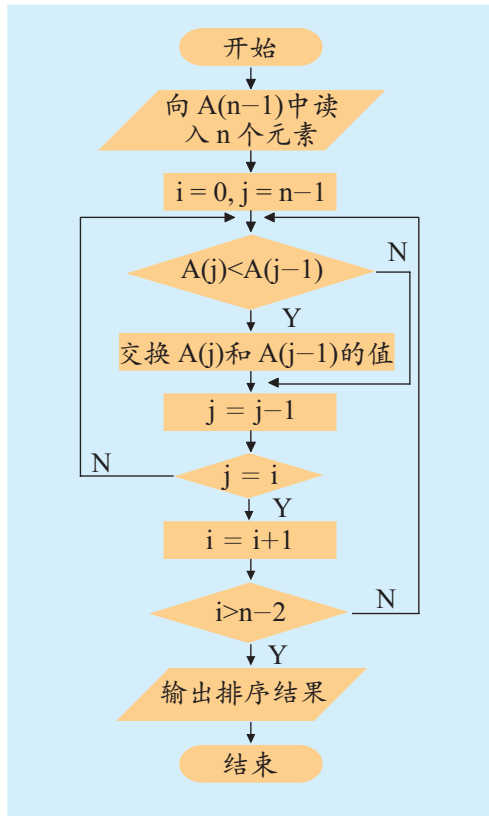


图 3-13 冒泡法排序流程图

个有序序列。将元素按从小到大的顺序排列称为升序排列，反之称为降序排列。使用冒泡法进行升序排序的算法流程图如图 3-13 所示。

对 5 个元素“4, 1, 2, 5, 3”的序列进行升序排序，用冒泡法排序共需 4 次排序，排序过程如图 3-14 所示，图中有标记的是进行对比的元素，颜色较深的是实际发生了交换的元素。



图 3-14 冒泡法排序示意图



参考图 3-14，请将下面的伪代码填写完整。

```

For i = 0 To _____
  For j = 4 To _____ Step-1
    If A(j) < A(j-1) Then
      交换 A(j) 和 A(j-1) 的值
    End If
  Next j
Next i
    
```

下面，我们就用冒泡法解决“明日之星英语演讲大赛”的排序问题。

1. 问题描述

组委会规定，各小组比赛前三名选手进入决赛。因此，我们需要对各小组成绩分别进行降序排序，找出各组前三名。

2. 算法分析

假定各小组都是 10 名选手；

定义两个数组，一个用于存储选手号(aryPlayer)，另一个存储选手对应的成绩(aryScore)；

程序运行时，读入小组选手号及其成绩；

通过两层循环对数组(aryScore)进行排序，外层循环控制排序的趟数，内层循环控制元素的对比与交换；

当成绩数组(aryScore)的元素需要交换时，同时交换选手数组(aryPlayer)的相应元素。

下面是算法的伪代码：

```
n = 10
将 10 名选手的编号读入数组 aryPlayer
将 10 名选手的成绩读入数组 aryScore
For i = 0 To n-2(共 9 次)
  For j = n-1 To i+1 Step-1
    If aryScore(j) > aryScore(j-1) Then
      交换 aryPlayer(j) 和 aryPlayer(j-1) 的值
      交换 aryScore(j) 和 aryScore(j-1) 的值
    End If
  Next j
Next i
```



(1) 找出 5 本书随机排为一列，用冒泡法对它们的定价进行手工排序，从中体会冒泡法排序的思想。

(2) 在“明日之星英语演讲大赛”的排序中，我们的假定是每组 10 名选手，实际上各组人数不尽相同，如何解决这一问题？

(3) 以小组为单位，自主探究其他排序方法，与冒泡法排序比较，找出它们各自的特点及适用范围。

3. 编程实现

新建一个 VB 标准 EXE 工程，在 Form1 窗体中添加两个 Label 控件和两个 Frame 控件；然后，在两个 Frame 控件中分别添加两个 Label 控件和两个

TextBox 控件；在窗体的下部添加一个 CommandButton 控件，控件布局如图 3-15 所示。

添加后按表 3-4 所示设置控件的属性值。



图 3-15 建立 VB 工程

表 3-4 对象的属性值

对象	属性	设置值	对象	属性	设置值
工程 1	(名称)	成绩排序	Label3,Label5	Caption	编号
	(名称)	frmSort		Label4,Label6	Caption
Form1	Caption	小组成绩排序	Text1		(名称)
	StartPosition	2-屏幕中心		Text	空
Label1	Caption	小组选手成绩排序	Text2	Appearance	0-Flat
	BackStyle	0-Transparent		(名称)	txtScore
	AutoSize	True		Text	空
	Font	隶书, 二号字		Appearance	0-Flat
Label2	ForeColor	&H00FF0000	Text3	(名称)	txtSNum
	Caption	小组选手成绩排序		Text	空
	BackStyle	0-Transparent		Appearance	0-Flat
Label2	AutoSize	True	Text4	Locked	True
	Font	隶书, 二号字		(名称)	txtSScore
	ForeColor	&H8000000F		Text	空
Frame1	Caption	选手成绩	Command-Button1	Appearance	0-Flat
Frame2	Caption	排序结果		Locked	True
				(名称)	cmdSort
				Caption	排序

控件数组是共享控件名称、事件的相同类型的控件集合，它们可以像一般的数组变量一样通过索引号被直接引用。

设置完毕后，移动 Label1 和 Label2，使它们重叠构成立体字；用鼠标单击“txtNum”控件，选择“编辑”菜单中的“复制”命令，复制 txtNum 控件；然后，单击“Frame1”控件，选择“编辑”菜单中的“粘贴”命令，会弹出如图 3-16 所示的消息框。通过这种方法添加的控件和原控件一起被称为“控件数组”。



图 3-16 创建控件数组

再用同样方法粘贴 8 个 txtNum 文本框，最后得到 10 个 txtNum 文本框和 10 个 txtScore 文本框，分别用于输入 10 名选手的编号和成绩。同理，在 Frame2 控件中也可以得到 10 个 txtSNum 文本框和 10 个 txtSScore 文本框，分别用于显示排序后的选手编号和成绩。

复制完成后，用“格式”菜单中的“对齐”命令对控件进行对齐操作。最后的界面如图 3-17 所示。

双击“cmdSort”命令按钮，选择“Click”事件过程，输入以下参考代码：



图 3-17 “小组成绩排序”界面

```
Private Sub cmdSort_Click()
    Dim aryPlayer(9) As Integer, intTemp As Integer
    Dim aryScore(9) As Single, sngTemp As Single
    Dim i As Integer, j As Integer
    ' 从文本框中读入 10 名选手的编号和成绩
    For i = 0 To 9
        aryPlayer(i) = txtNum(i).Text
        aryScore(i) = txtScore(i).Text
    Next
    ' 以下代码对成绩进行排序
    For i = 0 To 8 ' 控制排序趟数
        For j = 9 To i+1 Step -1
            If aryScore(j) > aryScore(j-1) Then
                ' 交换选手编号
                intTemp = aryPlayer(j)
                aryPlayer(j) = aryPlayer(j-1)
                aryPlayer(j-1) = intTemp
                ' 交换选手成绩
                sngTemp = aryScore(j)
                aryScore(j) = aryScore(j-1)
                aryScore(j-1) = sngTemp
            End If
        Next j
    Next i
    ' 输出选手编号和选手成绩
    For i = 0 To 9
        txtSNum(i).Text = aryPlayer(i)
        txtSScore(i).Text = aryScore(i)
    Next
End Sub
```

上段程序定义了两个数组aryPlayer和aryScore,分别用于保存选手的编号和成绩;定义了两个临时变量intTemp和sngTemp,用于排序时进行数据交换。程序首先通过循环语句,向数组中读入选手的编号和成绩;然后,通过两层循环对成绩进行排序;最后,再通过循环语句输出排序后的选手编号和成绩。



运行上面的程序,查看排序结果,你会发现成绩是由高到低排列的。如果要将成绩由低到高排列,该如何实现呢?

* 3.4.2 插入排序算法

插入排序也是一种比较常用的排序算法,其基本思想是:将一个数据序列看作两部分,前一部分是有序的,后一部分是无序的;排序时,把无序部分的数据元素逐个插入到有序部分,使得有序部分的元素个数不断增加,无序部分的元素个数相应减少,最后所有元素成为有序序列。

插入排序有直接插入排序、折半插入排序、链表插入排序、希尔排序等。其中,直接插入排序是一种比较简单的排序方法。下面以直接插入排序为例进行讲解。

具体来讲,直接插入排序的算法可以描述如下。

(1) 有一个无序序列 $A[1\dots n]$,初始状态下, $A[1]$ 自成有序部分, $A[2\dots n]$ 为无序部分。

(2) 当 $i=2$ 时,将 $A[2]$ 的值赋给Sentry(Sentry称为哨兵),比较Sentry和 $A[1]$ 的大小;如果 $A[1]>Sentry$,将 $A[1]$ 的值赋给 $A[2]$,再将Sentry的值赋给 $A[1]$,于是, $A[2]$ 就插在 $A[1]$ 前。这样,有序部分增加了一个元素,无序部分相应减少了一个元素。

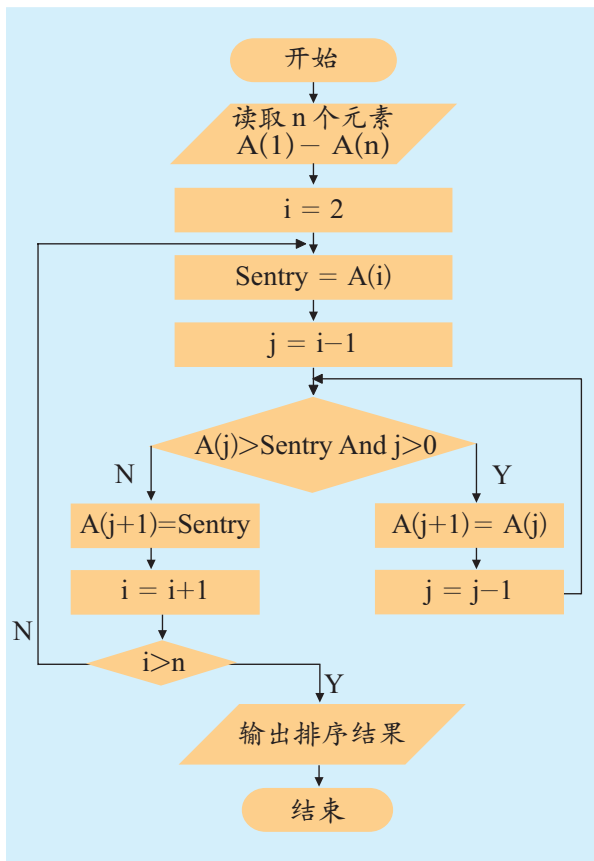


图 3-18 插入排序流程图

(3) 同理, 对于任意元素 $A[i](2 \leq i \leq n)$, $A[1 \dots i-1]$ 为有序部分, $A[i \dots n]$ 为无序部分; 先将 $A[i]$ 赋给哨兵 $Sentry$, 如果 $A[i-1] > Sentry$, 则将 $A[i-1]$ 的值赋给 $A[i]$, 即将 $A[i-1]$ 向后移动一位, 接着再比较 $Sentry$ 和 $A[i-2]$ ……直到 $Sentry$ 找到合适的位置, 完成 $A[i]$ 的插入排序。

(4) 依次类推, 将 $A[2]$ 至 $A[n]$ 之间的所有元素插入到有序部分, 排序完成。

直接插入排序的算法流程图如图 3-18 所示。

下面我们用直接插入排序算法模拟“明日之星英语演讲大赛”小组选手的成绩排序。

为了便于说明问题, 同时便于和冒泡法排序进行对比, 我们将用计算机产生随机数模拟选手的成绩。

程序运行时的参考界面如图 3-19 所示。程序用到两个列表框控件(ListBox), 左边的列表框(lstUnsorted)用于显示未排序的数据, 右边的列表框(lstInsSort)用于显示插入排序后的数据; 两个命令按钮, 其中, 左边的“生成序列(cmdMake)”命令按钮用于产生随机数来模拟成绩, 并将随机数添加到 lstUnsorted 列表框中; 右边的“插入排序(cmdInsSort)”命令按钮实现对数据的插入排序, 并将结果添加到 lstInsSort 列表框中。



图 3-19 插入排序程序界面

下面分析一下程序的参考代码。

首先在窗体的通用部分声明一个符号常量 MAXPLAYER, 用于在各个过程中对数据进行约束。参考代码如下:

```
Const MAXPLAYER = 50
```

下面是“生成序列”按钮对应的参考代码:

```
Private Sub cmdMake_Click()
    Dim i As Integer
    Dim sngElement As Single
    lstUnsorted.Clear ' 将列表框置空
    Randomize
    ' 随机生成 MAXPLAYER 个无序元素
    For i = 1 To MAXPLAYER
        ' 产生随机数, 乘以 10 后进行四舍五入, 结果保留 2 位小数
        sngElement = Round((Rnd() * 10), 2)
        ' 将产生的随机数添加到列表框中
        lstUnsorted.AddItem sngElement
    Next
    ' 设置插入排序按钮可用
    cmdInsSort.Enabled = True
End Sub
```


“插入排序”按钮的参考代码如下，其中，“哨兵”由 aryPlayer(0)元素充当。

```
Private Sub cmdInsSort_Click()
    Dim aryPlayer(MAXPLAYER) As Single
    Dim i As Integer, j As Integer
    ' 将无序列表中的数据读入排序数组中
    For i = 1 To MAXPLAYER
        aryPlayer(i) = lstUnsorted.List(i - 1)
    Next
    ' 初始状态下,数组的第1个元素被看作是有序的部分
    ' 从第2个元素起,依次将其插入到前面的有序部分
    For i = 2 To MAXPLAYER
        ' 将第i个元素的值赋给哨兵
        aryPlayer(0) = aryPlayer(i)
        ' 以下代码从前向后将有序部分大于哨兵的元素后移
        j = i - 1
        Do While aryPlayer(j) > aryPlayer(0)
            aryPlayer(j + 1) = aryPlayer(j)
            j = j - 1
        Loop
        ' 第j个元素不大于哨兵
        ' 则将哨兵(即第i个元素)插在第j个元素后
        aryPlayer(j + 1) = aryPlayer(0)
    Next
    ' 将排序后的数组添加到排序列表中
    lstInsSort.Clear
    For i = 1 To MAXPLAYER
        lstInsSort.AddItem aryPlayer(i)
    Next
End Sub
```



Timer()函数返回自午夜以来经过的秒数，精确度为 1/18 秒。

- (1) 将程序“小组选手成绩排序”中的冒泡法排序算法改为直接插入排序算法。
- (2) 不同的排序算法执行的效率是不同的，利用Timer()函数可以对它们执行的时间进行比较。新建如图3-20所示的窗体，分别用插入排序算法和冒泡排序算法编写“插入排序”按钮和“冒泡法排序”按钮的代码。在各自的过程中分别定义单精度型变量t用于计时，排序前用t = Timer()获取一个时间值，排序后再用表达式Timer() - t获取程序执行耗用的时间，并将结果放入对应的文本框中；通过重新定

义程序中的常量 MAXPLAYER 的值(例如, MAXPLAYER = 1000), 运行程序, 比较这两种算法的优劣。(参考源程序见光盘)



图 3-20 两种排序算法的对比

3.5 用递归法解决问题

“从前有座山, 山上有座庙, 庙里有个老和尚给小和尚讲故事, 讲什么呢? 从前有座山, 山上有座庙……”这个古老的故事大家可能都听过, 它的有趣之处在于故事里面嵌套了故事本身。在程序设计中, 如果一个函数也像这个故事一样, 自己调用了自己, 这种算法叫递归法(Recursion)。本节学习“兔子繁殖问题”的解决办法。通过本节的学习, 你可以:

- 理解什么是递归法, 学会用递归法思想分析问题
- 理解什么是自定义函数, 掌握自定义函数的定义方法
- 学会用递归法编写程序解决问题

3.5.1 什么是递归法

“从前有座山……”的故事蕴涵了典型的递归思想, 它的模型如图 3-21 所示, 函数 A 自己调用了自己, 生活中还有一种“相互调用”的现象。

如图 3-22 所示, 如果将两面镜子相对放在一起, 你会发现: A 镜子里有 B 镜子的映像, B 镜子里面又有 A 镜子的映像, 这样层层重叠, 无穷无尽。

在程序设计中, 像这种函数 A 调用函数 B, 函数 B 反过来再调用函数 A 的算法, 同样也是递归法, 如图 3-23 所示。

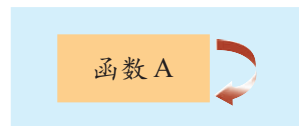


图 3-21 函数自身调用



图 3-22 镜子互照映像

如果一个函数在定义时，直接或间接地调用了自己，这种算法在程序设计中统称为递归法。

在“从前有座山……”的故事中，如果老和尚一直这样讲下去，这个故事将永无穷尽；“镜子照镜子”也是一样，两面镜子“你”照“我”，“我”照“你”，也是无穷无尽。但是，在程序设计中出现这种现象的话就麻烦了，它会将计算机的内存资源消耗殆尽。因此，必须在某个适当的时候结束这种调用。

递归法一般需要定义函数来实现。这里所讲的函数是自定义函数，它与第二章讲过的标准函数不同。

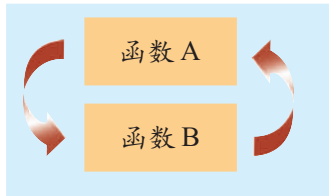


图 3-23 函数间互相调用

3.5.2 什么是自定义函数

语法说明：

“[]”中的内容为可选项；“<>”为必选项；“|”两侧为单选内容，即用“Public”就不能用“Private”。函数名的命名规则和变量名的命名规则相同。

我们知道，在程序设计中，函数是为了实现某种功能而编写的一段相对独立的程序，如 Abs()、Len()、Date()、MsgBox()等，这些都是 VB 的标准函数，可以直接使用，而我们自己编写的函数则称为自定义函数。

在 VB 中，自定义函数的形式如下：

```
[Public|Private] Function <函数名称> ([参数列表]) [As 类型]
    局部常量、变量定义
    语句组
    函数名称 = 返回值
End Function
```

既然有标准函数，为什么还要自定义函数呢？

标准函数定义了一般问题的解决方法，例如求字符串长度、截取字符串等，但许多问题单靠标准函数无法解决。例如，VB 的标准函数 Val() 只能取出字符串“123A456”中的“123”，而后三位数字则无法取出。有时我们需要从一个字符串中取出它包含的所有数字，这时可以自定义一个函数实现这个功能。

```
Private Function GetNum(ByVal strString As String) As String
    Dim i As Integer
    Dim strNumChars As String
    Dim strChar As String
    ' 用循环语句依次判断字符串中的每个字符是否为数字
    For i = 1 To Len(strString)
        strChar = Mid(strString, i, 1)
```

```

If IsNumeric(strChar) Then
    strNumChars = strNumChars & strChar
End If
Next
' 返回字符串中包含的数字字符
GetNum = strNumChars
End Function

```

新建函数可以通过 VB “工具” 菜单中的“添加过程”命令生成函数结构框架。

这个自定义函数带有一个参数 strString，传给它一个字符串，该函数可以返回字符串中包含的所有数字字符。

自定义函数的调用，可以有三种格式：

```

变量 = 函数名称 (参数)
Call 函数名称 (参数)
函数名称 参数

```

例如，对上例来说，执行语句 S=GetNum (“123A456”)后，则变量 S 的值为 “123456”。

自定义函数还可以实现代码重用，经过验证的自定义函数可以在多处反复使用而不用再次编写；同时，自定义函数还能使编写的代码更清晰、更健全。试想，如果一段代码有上百行，你很难保证它不出错，一旦出错，查找起来将非常困难。因此，实际设计中，往往把比较复杂的程序段拆分为若干个相对简单、相互独立的函数，这样不仅便于代码的编写，而且有利于错误的排查。

在 VB 中，说到自定义函数，就不能不提子过程。子过程的定义如下：

```

[Public|Private] Sub <子过程名称> ([参数列表])
    局部常量、变量定义
    过程语句组
End Sub

```

对于这个定义，我们并不陌生，前面用到的事件过程和这里的定义非常相似，请把两者加以对比，指出它们的不同之处。

子过程和函数本质是一样的，在 VB 中往往将函数看作特殊的子过程，只不过子过程用“Sub”关键字定义，函数用“Function”关键字定义；函数可以有返回值，而子过程则没有。

在其他一些语言，如 C / C++ 中，往往将这些处理过程看作函数而不是子过程。



- (1) 比较标准函数和自定义函数，指出它们的异同。
- (2) 子过程和函数有哪些区别？说出它们各自适用的场合。

3.5.3 递归法的实现

有人养了一对兔子,这对兔子以后每月生一对兔子,新生兔子从第三个月开始,也是每月生一对兔子,问12个月后这人有多少对新生兔子。

1. 问题分析

这个问题是公元13世纪意大利数学家斐波那契的名著《算盘书》里的问题。我们可以用图3-24来分析这个问题,图中每个色块表示一对兔子,其中白色色块表示新生兔子。从图中可以发现,每月新生兔子的对数为:1,1,2,3,5,⋯从第三个月起,当月新生兔子数为前两月新生兔子数之和。这个数列在数学上被称为“斐波那契数列”。

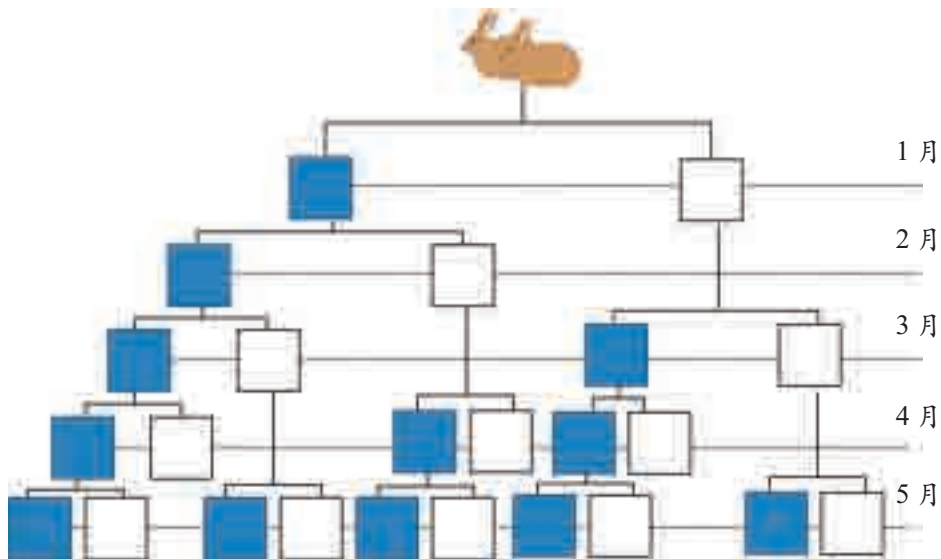


图 3-24 新生兔子示意图

2. 编程实现

这个问题如果不用递归法解决,其参考代码如下:

```
Private Function Hares(ByVal intMonth As Integer) As Integer
    Dim i As Integer
    Dim intCurMon As Integer ' 当前月新生兔子对数
    Dim intPreMon1 As Integer ' 前一个月新生兔子对数
    Dim intPreMon2 As Integer ' 前两个月新生兔子对数
    ' 前两个月分别为 1 对
    intPreMon1 = 1
```



```

intPreMon2 = 1
' 从第 3 个月起, 新生兔子为前两个月新生兔子之和
For i = 3 To intMonth
    intCurMon = intPreMon2 + intPreMon1
    intPreMon1 = intPreMon2
    intPreMon2 = intCurMon
Next
Hares = intCurMon
End Function

```

从斐波那契数列的特点看, 这个问题显然可以用递归法实现。递归法的参考代码如下:

```

Private Function Recursion(ByVal intMonth As Integer) As Integer
    If intMonth = 1 Or intMonth = 2 Then ' 第 1、2 个月, 新生兔子各为 1 对
        Recursion = 1
    Else ' 从第 3 个月起, 新生兔子为前两月新生兔子之和
        Recursion = Recursion(intMonth - 1) + Recursion(intMonth - 2)
    End If
End Function

```

在递归法中, 函数 Recursion() 自己调用了自己, 假如以 5 个月为例, 则函数的递归调用关系如图 3-25 所示。

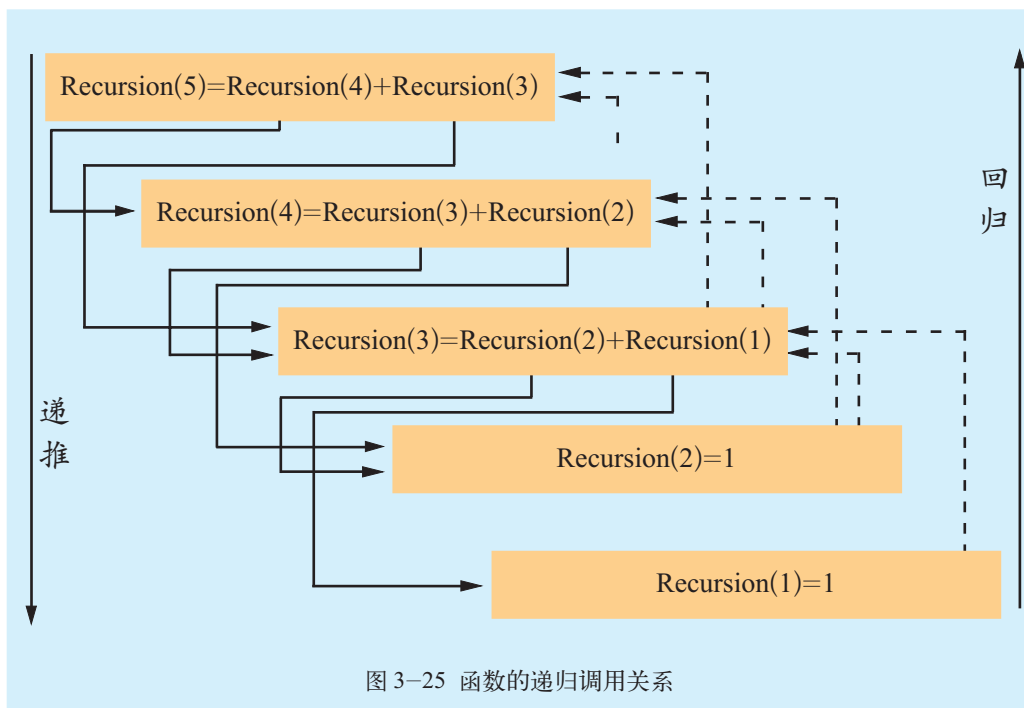


图 3-25 函数的递归调用关系

(1) 递推。

函数 Recursion() 先将 Recursion(5) 分解为 Recursion(4)+Recursion(3), 然后判断 Recursion() 中的参数, 如果参数不为 1 或 2, 则继续分解, 即: 将 Recursion(4) 分解为 Recursion(3)+Recursion(2); 将 Recursion(3) 分解为 Recur-

递归算法有两个必备条件: ① 递归分为递推和回归两个过程; ② 递推的结束条件 (如代码中的 If intMonth=1 Or intMonth=2 Then Recursion=1 就是递推的结束条件)。

ursion(2)+ Recursion(1)。

然后，再次判断 Recursion() 中的参数是否为 1 或 2，如果参数不为 1 或 2，则继续分解，直到 Recursion() 中的参数是为 1 或 2 为止。

(2) 回归。

将得到的 Recursion(1) 和 Recursion(2) 的值代入，并依次返回上一级。例如：Recursion(1)=1 和 Recursion(2)=1 代入，可得 Recursion(3)= Recursion(1)+ Recursion(2)=1+1=2，然后将 Recursion(3) 的值代入，可得 Recursion(4)= Recursion(3) + Recursion(2)=2+1=3。依次类推，最后算出 Recursion(5) 的值为 5。

函数的递归调用关系如图 3-25 所示。

从图 3-25 可知，递归法是靠层层调用实现的。函数先由上向下调用，当到达最底层后，再将函数值层层向上返回。

递归法虽然结构清晰，可读性较强，且容易用归纳法证明其正确性，但递归函数的调用方式是相当耗费计算机资源的，因而其效率比较低。



(1) 在图 3-25 中，跟踪递归法的调用路径，将函数的返回值代入函数调用处，体会递归法的特点。

(2) 比萨斜塔位于意大利中部比萨古城的教堂广场上，它是一座钟楼，始建于公元 1174 年，塔高 54.5 米，8 层圆柱形建筑，全部用白色大理石砌成。现从塔顶向下抛一小球，小球每次着地后反弹的高度是上次高度的四分之一，那么，小球静止前经过的垂直距离是多少？用递归法计算。

(3) 相传古印度婆罗门神庙的僧侣们流行一种被称为“汉诺塔”的游戏，游戏的装置是一个塔台，台上有 3 根柱子（记做 A, B, C），A 柱上自上而下放了 64 个金盘，按照由小到大的次序排放。游戏的规则是把 A 柱上的金盘全部移到 C 柱上；移动的过程中，一次只能移一个金盘，且大盘不能放在小盘上。他们相信当所有盘子移完的时候世界末日就到了。

试用递归法求解汉诺塔问题。求解时注意盘子的数量最好设为 10 个以下，否则计算机运行的时间可能太长。



比萨斜塔



(1) 讨论本章学习的五种算法，从程序的执行效率、代码编写的方便性、可读性等方面，分析它们各自的特点，指出其适用范围。

(2) 请用本章所学知识，设计程序绘制奥运会的五环图案。

第四章 程序设计思想和方法



结构化程序设计



面向对象程序设计

本章我们将从比较一段程序代码入手,引出结构化程序设计的概念,接着探究如何用结构化程序设计的方法解决问题,并逐步理解结构化程序设计的思想。面向对象程序设计是对结构化程序设计的继承和发展,我们将结合前几章学过的有关VB的知识,初步了解面向对象程序设计的思想和方法。



4.1 结构化程序设计

美国宇航局1963年的“水星计划系统”用了200万条指令；1967年的“双子星座计划系统”用了400万条指令；1973年的“阿波罗计划系统”用了1000万条指令；1979年的“哥伦比亚航天飞机系统”用了4000万条指令。

狄克斯特拉是一名荷兰计算机科学家，因最早提出以及首创结构化程序设计而闻名于世。

在20世纪40年代和50年代，由于硬件条件的限制，要求程序运行时占用尽可能少的内存空间。那时的程序设计，采用机器语言或汇编语言编写。一方面，程序设计需要很强的技巧，编程是少数“天才”的专利；另一方面，软件开发的周期长，编出的程序可靠性、可移植性差，且不便阅读与维护。

随着计算机应用领域的扩大，软件的需求量越来越大，对其复杂程度的要求也越来越高。到了20世纪60年代，软件业终于爆发了所谓的“软件危机”。为了克服这一危机，人们提出了采用结构化的程序设计方法以及用工程的方法开发软件等软件开发思想。

本节对“书店销售管理系统”进行分析。通过本节的学习，你可以：

- 理解结构化程序设计的思想
- 初步掌握结构化程序设计的方法

4.1.1 结构化程序设计的由来

1965年，狄克斯特拉(Dijkstra)曾提出“GoTo语句是有害的”、“一个程序的质量与程序中所含的GoTo语句的数量成反比”、“GoTo语句可以从高级语言中取消”。1968年狄克斯特拉给ACM通讯写了一篇短文，该文后以信件形式刊登，这就是具有历史意义的、著名的“GoTo Letter”。

他在信中建议：“GoTo语句很容易把程序弄乱，应从一切高级语言中去掉；只用三种基本控制结构就可以编写各种程序，而这样的程序可以由上到下阅读而不会返回。”这封信引起了激烈的讨论，人们逐渐认识到，这已经不是一个简单的GoTo语句问题了，而意味着一种新的程序设计思想与方法的产生。

表4-1中的两段程序都能在VB中正确运行，输入一位0~9的数，输出对应的中文大写数。你认为哪一段程序更清晰、更简单呢？

表4-1 程序比较

程序段 1	程序段 2
10 n = Val(InputBox("输入一位0~9数"))	n = Val(InputBox("输入一位0~9数"))
20 If n = 0 Then GoTo 120	Select Case n
30 If n = 1 Then GoTo 140	Case 0
40 If n = 2 Then GoTo 160	Print "零"
50 If n = 3 Then GoTo 180	Case 1
60 If n = 4 Then GoTo 200	Print "壹"
70 If n = 5 Then GoTo 220	Case 2
80 If n = 6 Then GoTo 240	Print "贰"
90 If n = 7 Then GoTo 260	Case 3
100 If n = 8 Then GoTo 280	Print "叁"

续表

程序段 1	程序段 2
110 If n = 9 Then GoTo 300	Case 4
115 GoTo 1000	Print "肆"
120 Print "零"	Case 5
130 GoTo 1000	Print "伍"
140 Print "壹"	Case 6
150 GoTo 1000	Print "陆"
160 Print "贰"	Case 7
170 GoTo 1000	Print "柒"
180 Print "叁"	Case 8
190 GoTo 1000	Print "捌"
200 Print "肆"	Case 9
210 GoTo 1000	Print "玖"
220 Print "伍"	End Select
230 GoTo 1000	
240 Print "陆"	
250 GoTo 1000	
260 Print "柒"	
270 GoTo 1000	
280 Print "捌"	
290 GoTo 1000	
300 Print "玖"	
1000 End	

4.1.2 结构化程序设计的思想和方法

结构化程序设计的思想规定了一套方法，要求程序设计者按照一定的结构形式来设计和编写。具体来说，结构化程序设计的思想包括以下三方面。

(1) 一个大的程序可划分为若干个模块，每个模块又可以继续分为更小的子模块，最终分出的模块能完成一个独立的功能，模块之间按层次关系进行组织。

(2) 程序由一些基本结构组成，任何一个复杂的程序都可以由三种基本结构组成，即顺序结构、选择结构和循环结构，如图 4-1 所示。

(3) 设计程序时，应采用自顶向下，逐步细化的

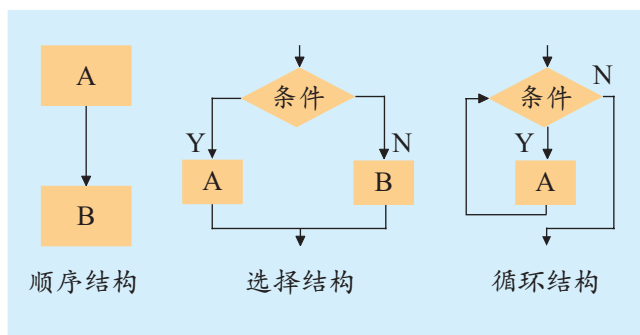


图 4-1 程序的基本结构

实施方法。先设计第一层，然后步步深入，逐层细化，逐步求精，直到整个问题可用程序设计语言明确地描述出来为止。这样，我们就可以集中精力考虑模块的结构，而不必急于考虑其细节，从而保证程序层次鲜明，逻辑清晰。

具有良好结构的程序易于设计、理解、调试和修改。下面我们用这种思想方法解决一个实际问题。

某书店需要设计一个售书管理系统，对进书、售书进行管理，并提供统计、分析、报表等功能。

对于这样一个系统，初看起来觉得无从下手，但如果从书店的业务流程分析出发，首先可以将它分为进书管理、零售管理、批发管理等若干个大的模块。每个模块又可以按其功能再分为若干个小的子模块，这样层层细分，直到不能再分为止。最底层的子模块具有独立的功能，可以用程序设计语言明确地描述出来。例如，零售管理模块又可再分为零售售书、零售退书、零售查询、零售盘点4个子模块，如图4-2所示，这时就可以用程序语言直接编程实现。

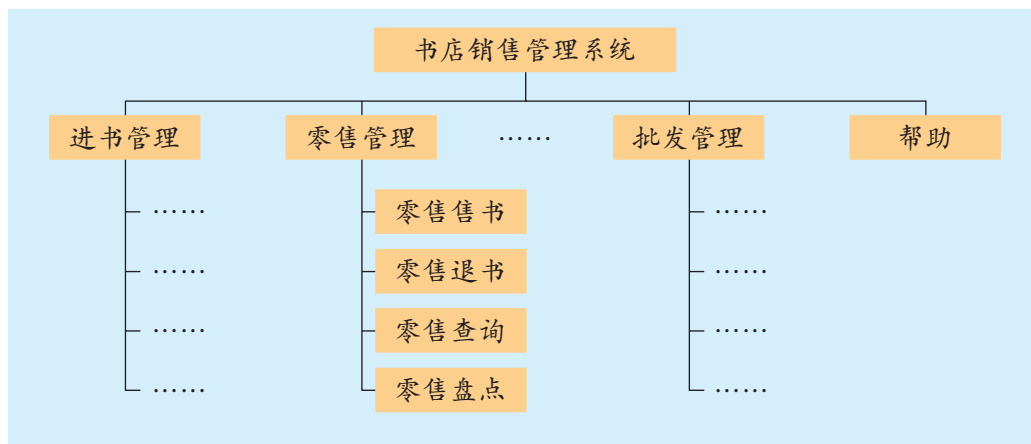


图 4-2 书店销售管理系统

回顾第三章学习的算法，其实它们都是一个个独立的子模块，每一个都完成一个独立的功能。例如，查找、排序都是具有单一功能的子模块，并且这些过程一般具有通用性，我们还可以将这些过程改造为通用过程或函数，以便在不同的程序中调用，从而实现代码的共享。



(1) 分组讨论，完善“书店销售管理系统”的模块划分，体会“自顶向下，逐步细化”的分析方法。

(2) 请根据本校的实际情况，运用结构化程序设计的思想与方法，为学校图书馆规划一个科学的图书管理系统。选取其中的一个子模块(如图书信息查询子模块)，尝试编程实现。

4.2 面向对象程序设计

结构化程序设计成功地处理复杂问题提供了有力的手段，然而到了20世纪80年代末，它的一些缺点越来越突出，当数据量增大的时候，数据与处理这些数据的方法之间的分离使程序变得越来越难以理解。为了克服这些缺点，人们提出了一种新的思路——面向对象的程序设计方法，该方法日益受到人们的重视，给软件产业带来了新的飞跃。通过本节的学习，你可以：

- 理解面向对象程序设计的思想
- 初步掌握面向对象程序设计的基本概念
- 能够应用面向对象程序设计的方法分析简单的问题

4.2.1 面向对象程序设计的思想

假如我们准备设计一个电子商务系统，按照结构化程序设计的方法，首先需要分析它的业务流程，然后据此进行设计如图4-3所示。

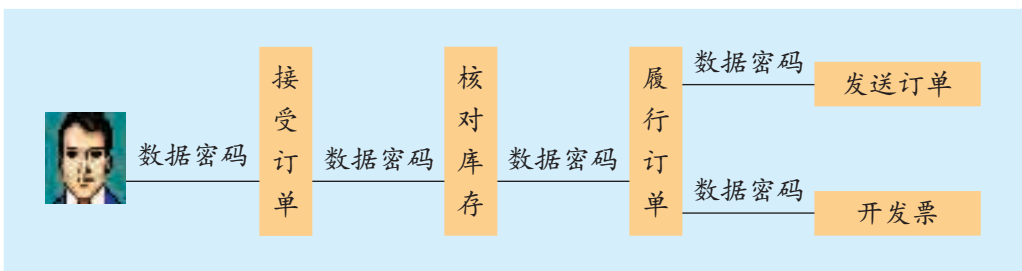


图 4-3 电子商务系统业务流程

在整个流程中，每个过程都必须与数据库交互。假如要改变订单类型，就需要修改数据库结构，业务流程的相应模块也将发生变化，这样，程序设计的许多过程都要重新编写，工作量将是巨大的。

如果采用面向对象的程序设计方法，就可以将数据与其对应的处理方法“捆绑”在一起，相互间通过接口进行交互。当产生新的订单类型时，原系统只需从原有的订单类型“派生”出新的订单类型就可以解决了，从而避免“牵一发而动全身”的现象发生。对程序而言，修改的工作量也将大大减少。

通过对比可以看出，结构化程序设计是从系统的功能入手进行描述的。在实际设计中，用户的需求和软、硬件条件是不断变化的，按照功能划分设计的模块必然也是易变的和不稳定的，这样开发出来的模块可重用性不高。

面向对象程序设计从所处理的数据入手，以数据而不是以功能为中心来描述系统。数据相对于功能而言，具有更强的稳定性，因此，以数据为中心设计出来的系统，具有较强的稳定性，模块的可重用性也比较高。

面向对象程序设计是目前最流行的程序开发技术，它的提出并不是对结构化程序设计的否定，而是对它的继承与发展，面向对象程序设计中依然渗透着结构化程序设计的思想与方法。

面向对象程序设计围绕真实世界的概念来组织模型，采用对象来描述客观实体，这和我们平时的思维习惯是一致的。我们考虑问题时，总是将事物看成一个一个的对象，小到微观粒子，大到宏观宇宙，我们关注的是对象本身，而不是对象的内部细节。比如，当我们把联合国大厦作为研究对象时，关注的是大厦里面有哪些部门、分别起什么作用等问题，没必要关注它的一砖一木。



Windows操作系统之所以如此易学易用，很大程度上也是依赖于面向对象的程序设计技术。打开一个文件夹窗口，它就会在窗口的状态栏中显示当前窗口内共有几个对象。讨论：在Windows中，还有哪些方面体现了面向对象程序设计思想？

4.2.2 面向对象程序设计的基本概念

面向对象的程序设计目标，在于编制能符合人们在真实世界中的思维模式的软件。学习面向对象的程序设计，需要掌握以下几个基本概念。

1. 对象

当我们乘汽车时，首先想到的是汽车的燃油子系统还是发动机的工作原理？两者都不是！在我们的意识中，是将汽车当作一个整体对象看待，我们没必要了解构成汽车的所有结构，作为乘客，只关心与我们有关的对象就可以了。

在现实世界中，“对象”无处不在，我们身边的一切事物都可以看作是对象，例如，一台计算机、一个闹钟、一支钢笔、一个人、一个国家都可以看作是对象。

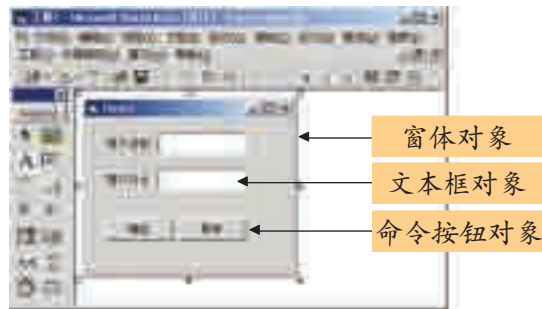
对象可以非常简单，也可以很复杂，复杂的对象往往又包含若干个简单对象。例如，一台计算机是对象，它的主机、显示器、键盘等也可以分别看作是对象。对象既可以是看得见、摸得着的，也可以是看不见、摸不着的，例如，计算机硬件是对象，计算机软件也可以看作是对象。

在VB中，我们总是和对象打交道。新建一个窗体就是新建了一个窗体

对象；从VB的工具箱中将一个控件拖放到窗体中，实际上也生成了一个对象，如图4-4所示。

利用面向对象的可视化开发工具的优点是：许多对象的功能已由开发系统设计好了，我们只需直接组合就可以完成应用程序的编写。

在面向对象的程序设计中，对象的特征、功能、对象之间进行通信的消息分别称为“属性”、“方法”、“事件”。



对象：Object
属性：Property
方法：Method
事件：Event

图4-4 VB中的对象

2. 属性

对象的属性是用来描述对象的外部特征的。例如，一个人具有性别、年龄、身高、体重等属性特征。在VB中，窗体对象的属性有名称、标题、字体、背景颜色；命令按钮对象的属性有按钮名称、标题、背景颜色等，如图4-5所示。



图4-5 VB中的对象属性

属性的引用方法为：

对象名.属性名 = 属性值 或 变量 = 对象名.属性名

例如，Form1.Caption = "打字测试软件"；strText = txtMyText.Text可以改变窗体的标题，并给字符串变量strText赋值。

3. 方法


方法是一个对象允许其他对象与之交互的方式，它表明了一个对象所具有的能力。例如，一个人会唱歌、会跳舞就是其方法。在VB中，打开Form1的代码窗口，输入“Form1.”，列表框中显示  标记的就是Form1的方法，如图4-6所示。



图4-6 VB中对象的方法

调用对象的方法为：

对象名.方法名 [参数列表]

例如，Form1.Refresh 可以刷新窗体显示内容。

4. 事件与事件驱动

当你给闹钟定时后，闹钟会在指定的时间触发振铃动作；当有人呼喊你的名字时，你也会作出相应的反应……日常生活中，很多对象都具有对消息的反应机制。

VB的执行机制与此类似，是按事件驱动方式工作的。例如，用户单击按钮就触发了Click事件，程序会执行Click事件过程；当我们在窗体中拖放一个Timer控件，并设置了它的“Interval”属性，系统就会在指定时间触发“Timer”事件。

VB为窗体和大多数控件都规定了一组事件，当应用程序被启动后，VB随时准备捕获各种事件。一旦发生了某个事件，VB就会执行与该事件相联系的事件过程；执行完后，VB将等待下一个事件的发生。

VB的这种事件驱动机制为编写程序带来极大的便利，我们只需根据具体情况，选择适当的事件编写代码就可以构建应用程序了，如图4-7所示。

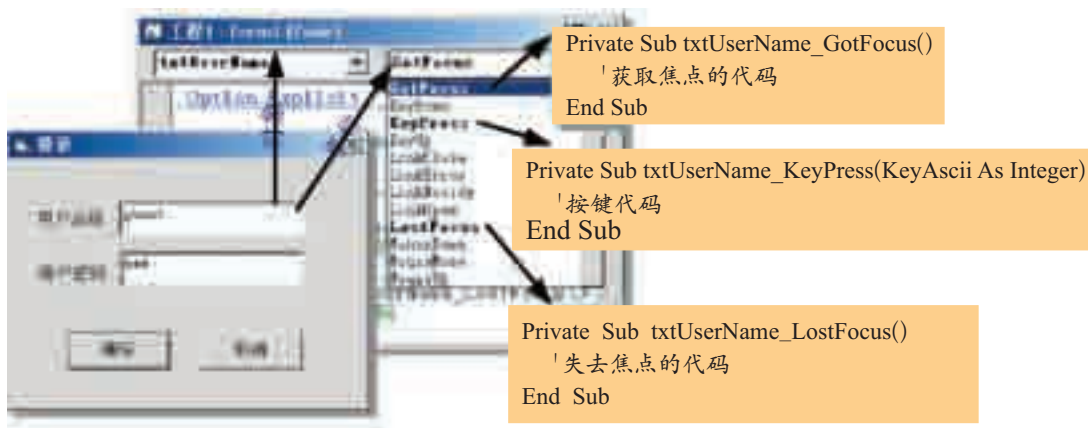


图4-7 对象的事件过程

5. 类(Class)

现实生活中，我们通常会把具有共同性质的事物划分成一类，得出一个抽象的概念“类”。

在面向对象程序设计中我们也往往把具有共同属性、方法或事件的对象归为“类”。类是在对象之上的抽象，是对象的模板；对象是类的具体化，称为类的实例。当我们说“标签”的时候，指的是标签类；当我们说某个标签的时候，指的是标签类的一个实例。类可以有子类，也可以有父类，类与

类之间形成层次关系。例如，动物——鸟——天鹅。

使用VB进行程序设计的方便之处就在于VB给我们定义了许许多多的类，从窗体、工具栏、状态栏、命令按钮、图片框、单选框、复选框到时钟、文件、文件夹、驱动器、滚动条、旋转按钮等，它提供了我们在程序设计中用到的多数类，这大大方便了程序的编写。编写一般的程序，我们只需要生成相应类的实例(即生成对象)就可以实现想要的功能。

VB工具箱中的类是可视类，设计程序时，将它们直接拖放到窗体中就可以生成类的实例(对象)。如果用“工程”菜单中的“引用”命令，还可以使用许多不可见类，这些类一般包含在以“.dll”为扩展名的文件中，称为动态链接库，利用动态链接库可以较好地实现代码的重用。例如，在Windows的应用程序中，你到处都能见到菜单、工具栏、状态栏这些对象，实际上它们大多来自相同的动态链接库文件。

在VB中，自定义类需要放在类模块中，类模块是以“.cls”为扩展名的文件。

添加类模块的方法是：在资源窗口中右击，选择“添加”→“添加类模块”命令就可以在当前工程中添加一个类模块。

类模块添加后可以通过属性窗口改变类的名称，在类模块中定义成员变量，然后通过Property Let/Get设计类的属性，类的方法定义和一般函数或过程的定义相似。为了使自定义类的属性和方法能在类的外部访问，类的属性和方法一般应该用Public修饰，如果不加指定，默认属性和方法为Public类型。

请看下面的学生类(Student)的例子：

' 以下定义类的内部成员变量

Private strName As String

Private intAge As Integer

Private dtmBirthday As Date

Private strClass As String

' 定义姓名属性

Public Property Let Name(sName As String)

 strName = sName

End Property

Public Property Get Name() As String

 Name = strName

End Property

' 定义生日属性

Property Let Birthday(dBirthday As Date)

 dtmBirthday = dBirthday

End Property

Property Get Birthday() As Date


```

        Birthday = dtmBirthday
    End Property
    ' 定义年龄属性, 它是只读属性
    Property Get Age() As Integer
        Age = Year(Now) - Year(dtmBirthday)
    End Property
    ' 定义班级属性
    Property Let Class(sClass As String)
        strClass = sClass
    End Property
    Property Get Class() As String
        Class = strClass
    End Property
    ' 定义学习方法
    Public Sub Study(strSubject As String)
        ' 在这里定义学习方法
    End Sub
    ' 定义舞蹈方法
    Public Function Dance() As Boolean
        ' 在这里定义舞蹈方法
    End Function

```

然后, 我们就可以在其他模块中生成 Student 类的实例了, 代码如下:

```
Dim myObj as New Student
```

生成类的实例后, 就可以像用 VB 内部类定义的对象那样使用, 在 myObj 对象后面键入一个圆点后, VB 同样也会给我们列出 Student 类中定义的属性和方法, 如图 4-8 所示。

同样, 我们可以给 myObj 对象赋属性值, 读取 myObj 的属性值, 访问 myObj 的方法。如:

```

myObj.Name = "LiMing"
myObj.Birthday = #5/10/1995#
myObj.Study("English")

```

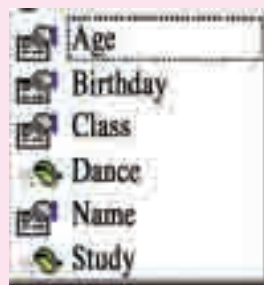


图 4-8 Student 类中的属性



- (1) 请看图 4-9, 就台灯这个对象, 讨论它可以有哪些属性和方法。
- (2) 新建一个窗体对象, 在上面添加文本框、命令按钮、单选按钮、复选按钮等控件, 然后打开代码窗口, 从左上角列表框中选择一个对象, 从右边的过程列表

框中查看该控件有哪些事件。结合 Windows 的使用经验,你认为哪些事件是最常用的?

(3) 举出生活中的类与对象的实例,讨论在面向对象程序设计中,类与对象的区别与联系。



对象: 台灯
属性:
方法:

图 4-9 台灯的属性和方法

4.2.3 面向对象程序设计的基本特征

面向对象程序设计的最突出的特性就是封装、继承和多态性。衡量某种程序设计语言是否为面向对象的程序设计语言,主要看它是否具有这三个特性。

1. 封装(Encapsulation)

在大多数面向过程的程序设计语言(如C语言)中,程序代码是作为功能模块编写的,由功能模块实现对数据的操作;而在面向对象程序设计语言中,利用封装功能,可将数据及其相关的操作组合为一个对象,对象的内部实现是它自己的事,对外它是一个黑匣子,外界也不需要了解它内部的实现细节,对象通过展示公共接口和外界进行信息交流。

上面的 Student 类为例,我们可以在 Study 方法中定义 Student 类的学习行为(方法)。不管它是复杂,还是简单,对外它只展示了接口 Study (strSubject As String),在对象外面是无法看到 Study 方法的实现细节的。类的使用者一般也不关心它的实现细节,而更多关心它的功能是否满足自己的要求。

同理,当我们调用了窗体的 Circle 方法,并传递圆心、半径等参数, Circle 方法就会按指定的参数画圆;至于它是用一个个点画出来的,还是用一段段短线画出来的,我们不得而知,也没必要知道。

在 VB 中,其实每一个控件都对自己的属性、方法等进行了封装。使用 Form1.PrintForm 就可以打印窗体,但我们不知道数据是如何被送往打印机的,就像我们乘汽车时不需要考虑发动机的工作原理一样。在代码窗口中键入一个对象名和一个圆点,VB 会给我们列出该对象的许多属性和方法,同样,我们也无法看到它们的实现细节。

VB 6.0 及其以前的版本只拥有面向对象的部分特性,所以,有人称其为基于对象的程序设计语言;VB.NET 才是真正的面向对象的程序设计语言。

在 Windows 中，对象的封装也得到了完美的体现，它的每一个对象其实都包含了许多复杂的操作，但我们不需要关心这些被屏蔽了的细节，只需在图形用户界面下进行简单操作即可完成任务。

2. 继承(Inherit)

孩子为什么长得像父母？因为他(她)继承了父母的遗传基因。程序设计中，类与类之间也具有这种关系。继承是类与类之间共享数据和方法的一种机制，被继承的类称为父类，也叫基类或超类；继承的类称为子类，也叫派生类。

比如，类 B(熊猫)继承类 A(熊科动物)，那么，类 B(熊猫)就具有类 A(熊科动物)的特性(属性)和能力(方法)。因此，要构造一个新类，只需继承基类，再描述与基类不同的少量特征(即增加一些新的属性和方法)就可以了，如图 4-10 所示。

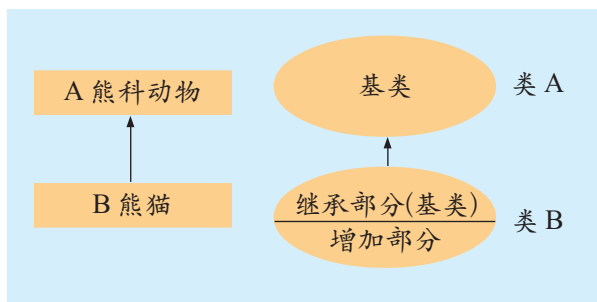


图 4-10 继承示意

代码重用是面向对象程序设计的精髓，继承是实现代码重用的重要手段。在面向对象程序设计中，继承意味着“自动拥有”，即子类中不必从头定义基类中定义过的属性和方法，它自动地、隐含地拥有基类的属性和方法；它使得软件开发不需要从零开始，这大大增加了程序的重用性，同时也使得程序更容易维护。

继承分为实现继承和接口继承，VB 6.0 的类模块只提供接口继承，而接口继承只是说明继承，因此，VB 6.0 的继承功能是非常有限的。这一状况在 VB.Net 中得到彻底改变，VB.Net 不但支持接口继承，而且支持实现继承。

3. 多态性(Polymorphism)

多态性是面向对象程序设计的又一个重要特性。它描述的是同一个消息，可以根据发送消息对象的不同，采用多种不同的行为方式，换句话说，不同的对象收到同一消息，可以产生完全不同的行为。利用多态性，用户只需发送一般形式的消息，而将所有的实现留给接收消息的对象，接收对象根据收到的消息作出相应的动作。例如，在 Windows 中，打开“我的电脑”，分别右击“C 盘”和“控制面板”对象，尽管都是单击事件(消息)，但由于

接收消息的对象不同，显示的快捷菜单也是不同的。

多态性的实现受到继承性的支持，利用类继承的层次关系，把具有通用功能的消息存放在高层，而实现这一功能的不同行为放在较低层次，使得在低层次上生成的对象能响应通用消息。在面向对象程序语言中，“重载”与“多态性”是紧密相连的。

重载是指允许存在多个同名函数，而这些函数的参数表是不同的（参数个数不同，或参数类型不同，或两者都不同）。例如：

```
Public Function Draw(Byval i As Single,Byval j As Single)
'画长方形语句
End Function
Public Function Draw(Byval i As Single)
'画正方形语句
End Function
```

需要说明的是，VB 6.0 及其以前的版本基本上不支持多态性，函数的重载在 VB 6.0 中也被认为是非法的，运行程序时，VB 会提示“发现二义性名称”。

多态性大大提高了程序的抽象程度和简洁性，而且它最大限度地降低了类和程序模块之间的耦合性，提高了类模块的封闭性，使得它们无需了解对方的具体细节，就可以很好地协同工作。这个优点，对程序的设计、开发和维护都有很大的好处。

面向对象程序设计的这三大特征是相互关联的，封装性是基础，继承性是关键，多态性是补充，而多态性又必须存在于继承的环境中。



回顾第二章中“可视化编程的一般步骤”部分有关面向对象的知识，讨论你编写的程序用到了哪些对象、方法和事件。




哥德巴赫猜想是数论中的一个著名难题，它是由法国数学爱好者克里斯蒂安·哥德巴赫于 1742 年在给著名数学家欧拉的一封信中提出的。

“哥德巴赫猜想”可以表述为：任何一个大于等于 4 的偶数都可以表示为两个素数之和。尽管这个问题看起来如此简明清晰，但 200 多年来，有许多数学家为其呕心沥血、绞尽脑汁，却始终无人能够证明或者证伪这个猜想。

请你设计一段程序，在有限的范围内验证“哥德巴赫猜想”，如果发现某个偶数不能被分解为两个素数之和，则证实了“哥德巴赫猜想”是错误的。(果真如此，则可称是数学史上的一大发现!)否则证实“哥德巴赫猜想”在所给的范围内成立。


第五章 软件是如何开发的



任务分析与系统设计



代码编写与调试



软件测试与维护

本章我们将通过一个实际问题的解决,学会综合运用VB开发工具,亲历软件开发的过程,使你了解软件开发的基本流程,加深对结构化程序设计和面向对象程序设计的思想和方法的理解。在这一完整的过程中,你将真正感受到软件开发的不易和问题解决后的无比喜悦。



5.1 任务分析与系统设计

“明日之星英语演讲大赛”决赛就要开始了。学校程序设计兴趣小组的同学们听说后，主动和组委会联系，希望发挥自己的特长，为大赛设计开发一个软件，支持大赛的举办。

本节将从问题分析入手，对“明日之星英语演讲大赛”软件做整体规划，为下一节的编程实现奠定基础。通过本节的学习，你可以：

- 知道任务分析和系统设计的一般步骤
- 学会应用程序设计方法进行系统设计和简单的模块划分

软件生命周期分为5个阶段：需求分析阶段、设计阶段、实现（编码）阶段、测试阶段、维护阶段。

5.1.1 任务分析

“明日之星英语演讲大赛”由校团委和学生会共同组织举办，本次大赛定于10天之后举行，大赛先分小组进行了预赛，最后总共有30名选手参加决赛。本次大赛邀请了10位专业教师担任评委，评分规则采用10分制，选手最后的得分为去掉一个最高分和一个最低分后的平均值。本次大赛要评出一等奖3名，二等奖5名，三等奖10名。

大赛组委会要求在决赛前设计出一个完整的软件，并要求该软件安全、方便、高效、可靠。

任务分析是软件开发的前提，只有科学地进行任务分析，才能保证开发出的软件具有针对性、实用性。因此，我们需要走访大赛组织者，对所要解决的问题进行详细地调查研究，弄清楚大赛的每一个流程及需要解决哪些具体问题。在此基础上，进行问题定义、可行性分析和需求分析，如图5-1所示。

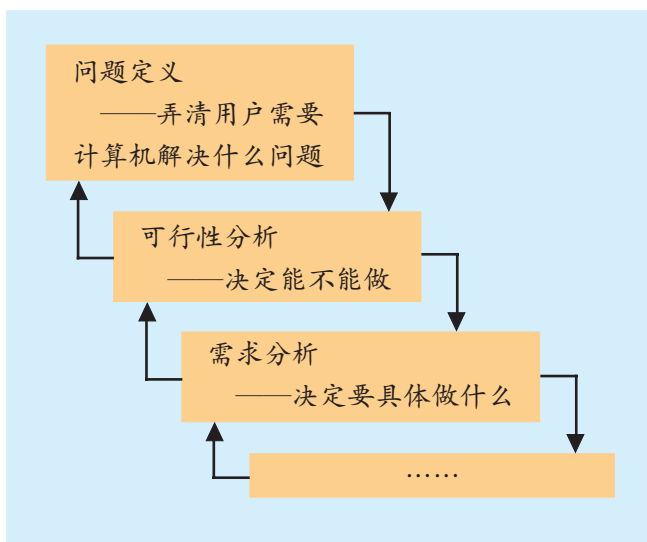


图 5-1 任务分析

1. 问题定义

问题定义主要是弄清楚用户需要计算机解决的问题。在程序设计中，

美国丹佛国际机场拥有最先进的机场技术。但曾经在开发它的自动化行李处理系统的时候,由于项目负责人将问题定义得过大,从而引发了轰动一时的进度超时灾难,并导致了项目的失败。

设计者根据自己对问题的理解,从自然问题描述中抽取出能用计算机解决的主要问题,并经用户审查认可,从而避免问题定义不准确带来设计上的偏差。

经过与“明日之星英语演讲大赛”组委会的多次沟通,程序设计兴趣小组讨论后认为,有些问题不便于或不需计算机来完成,而需要计算机解决的问题如下:

- (1) 输入并存储决赛选手的基本信息;
- (2) 输入并统计 10 位评委的评分结果;
- (3) 对选手得分进行排序。

2. 可行性分析

学校程序设计兴趣小组的同学们利用表 5-1 进行了可行性分析。

表 5-1 可行性分析

可行性分析表		
开发天数	10 天	这么长时间可以完成项目吗? 是 <input type="checkbox"/> 否 <input type="checkbox"/>
开发人手	3 人	这么多人可以完成项目吗? 是 <input type="checkbox"/> 否 <input type="checkbox"/>
开发工具	VB	利用 VB 可以编程解决问题定义中的“问题”吗? 是 <input type="checkbox"/> 否 <input type="checkbox"/>
开发技术难度	容易 <input type="checkbox"/> 比较难 <input type="checkbox"/> 很难 <input type="checkbox"/>	
结论	可以完成这个项目吗? 是 <input type="checkbox"/> 否 <input type="checkbox"/>	

经过分析,程序设计兴趣小组的同学们一致认为他们完全有能力完成这项开发任务。

3. 需求分析

整个系统千头万绪,需求分析工作从哪里入手呢?常用的方法就是建立和分析系统的模型。我们把用户目前使用的系统称为“当前系统”,应用计算机以后的系统称为“目标系统”,需求分析可以按照以下步骤进行。

- (1) 通过对现实环境的调查研究,获得当前系统的具体模型,如图 5-2 所示。
- (2) 剔除具体模型中的非本质因素,抽象出当前系统的逻辑模型。

大型项目的可行性分析一般包括经济、技术、人和社会环境等多种因素。经济方面主要针对成本与效益进行分析;技术方面主要解决能否在给定时间达到规定的质量问题;人的方面指是否有足够的相应素质的人员;社会环境方面主要包括市场与政策的研究。

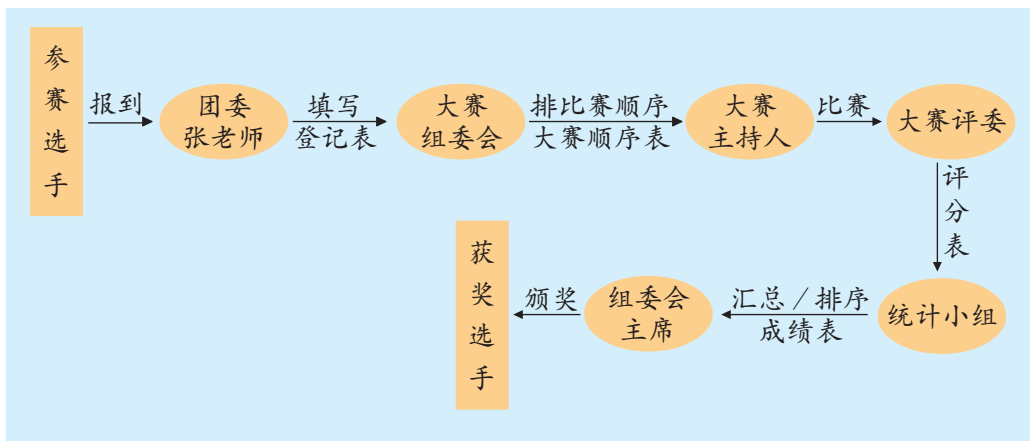


图 5-2 大赛具体模型

在图 5-2 中，团委张老师、大赛组委会、大赛主持人、大赛评委等具体的人是可能变动的，而身份确认、填写登记表、编排大赛顺序等则是相对稳定的。据此可以分析出大赛的逻辑模型，如图 5-3 所示。

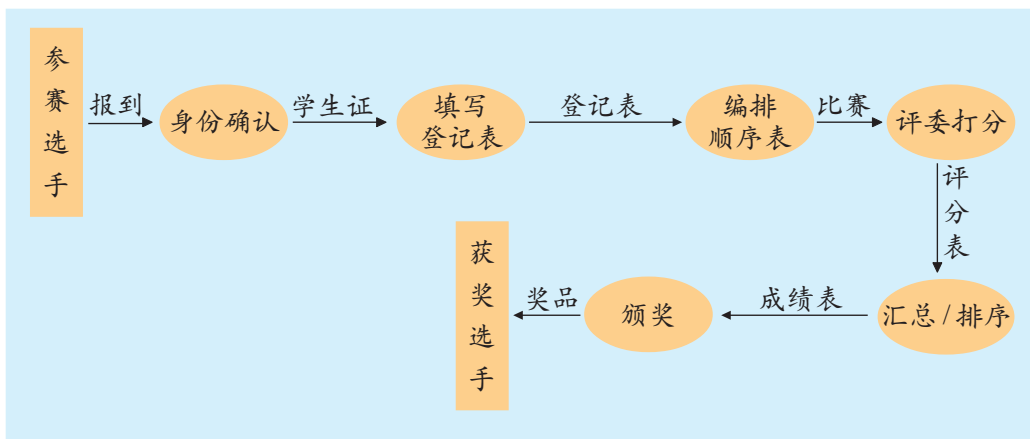


图 5-3 大赛逻辑模型

(3) 分析当前系统与目标系统的差别，建立目标系统的逻辑模型。

目标系统是使用计算机的系统，它是对当前系统的模拟，不一定和当前系统完全一样。例如，评分、颁奖等活动计算机无法完成，目标系统中需要剔除这些因素；而对组委会提出的安全要求，我们需要增加用户密码的确认这一项，如图 5-4 所示。

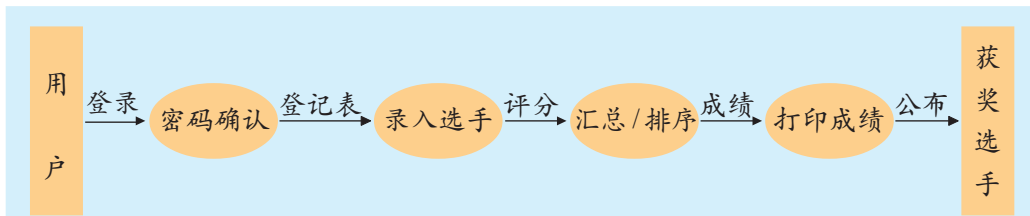


图 5-4 大赛计算机逻辑模型

(4) 对目标系统进行完善和补充，并写出完整的需求说明。

当目标系统提出后，我们需要反复多次将它和当前系统进行对比，仔细考虑还有哪些细节没有考虑到，目标系统中还有哪些因素需要进一步完

大型软件的需求分析非常烦琐费时，用计算机来辅助需求分析可以使软件开发更趋于工程化和标准化。目前已有 PSL/PSA、SREM、GASDT 等计算机辅助需求分析工具。

善等，最后写出详细的分析文档，并画出数据流图。

数据流图是以图形的方式描述数据在系统中流动和处理的过程，由于它只反映系统必须完成的逻辑功能，所以它是一种功能模型。

5.1.2 软件设计

任务分析完成后，就进入了软件设计阶段。在整个软件开发过程中，设计阶段是最具创造性的阶段。软件设计阶段就是把任务分析阶段产生的分析文档转换为用适当手段表示的设计文档。

根据任务分析，我们把“明日之星英语演讲大赛”软件分为以下几个大的模块：用户登录、信息处理、现场计分、成绩排序；为了方便用户使用，还需要添加帮助模块。对于信息处理模块，进一步划分可以得到录入记录和显示记录两个子模块，如图 5-5 所示。

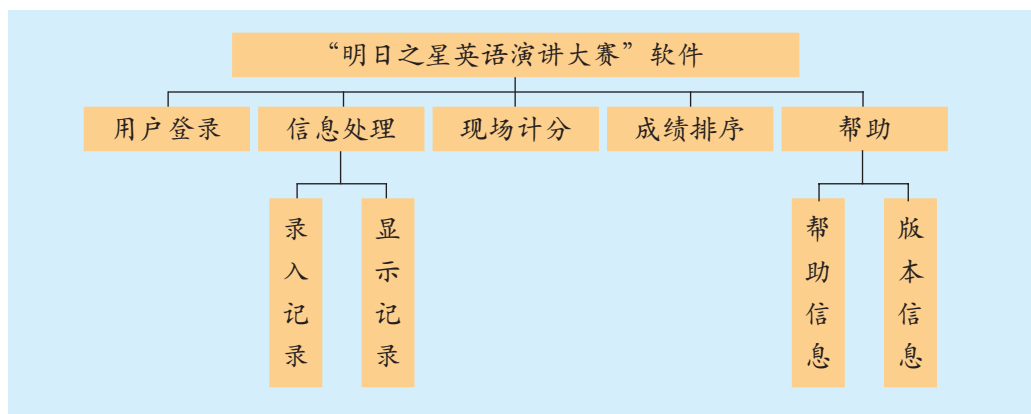


图 5-5 模块结构

模块划分之后，要思考表 5-2 中的问题，然后和其他同学讨论一下软件设计时还应考虑哪些因素。

表 5-2 软件设计问题

软件设计问题	你的思考
软件的功能满足需求分析中提出的所有要求吗	
软件功能的划分中有两个模块或两个以上模块的功能重合吗	

续表

软件设计问题	你的思考
划分好的软件功能模块还能够继续划分为更多的子模块吗	
你认为还要考虑的其他问题	① ② ③

对比建筑业、汽车制造业、计算机硬件业等产业,其形成规模化效应的成功道路都是将生产可复用的标准化构件与基于可复用构件的系统组装相结合。软件产业规模化也必然基于软件复用。软件作为人类知识的固化,具有复杂的逻辑性、构造性以及持续的演化等特性,这使得软件复用技术较为复杂,软件复用一直是软件工程研究和追求的目标之一。近年来随着面向对象技术的成熟、领域工程研究的深入,以及软件构件/构架技术(包括分布式对象技术,如CORBA、DCOM、EJB等)的发展,软件复用已经具备了现实的技术基础。其中,软件构件/构架技术是实现软件复用的核心技术。

我国重点支持的科技攻关课题——青鸟工程提出了软件生产线的概念和思想,基于构件/构架复用思想,突破传统开发模式,可形成软件产业内部的合理分工,实现软件工业化生产。青鸟软件生产线中的活动主要包括领域工程、构件管理、应用工程和质量保证等几个方面。青鸟软件生产线系统则对这些活动提供全方位的支持,整体分为方法(指南和标准规范)和相应拥有自主知识产权,先进实用的支撑工具体系两个方面,如图5-6所示。

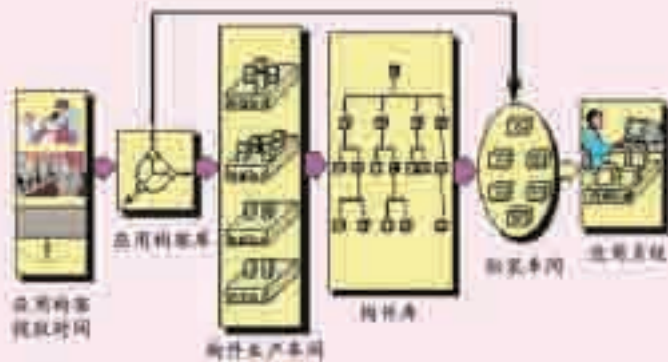


图 5-6 青鸟软件生产线概念示意



(1) 请思考“明日之星英语演讲大赛”的任务分析,你认为还需要考虑哪些方面的因素,根据你的分析进行修改。

(2) 请你对下面的几个问题做需求分析,填入表5-3中,并画出模块结构图。

问题一: 为学校运动会设计一个比赛评分软件;

问题二: 设计一个班级管理软件,例如成绩管理、考勤管理等;

问题三: 设计一个迷你英语词典软件,用它查找高中阶段常用的英语词汇。

表 5-3 需求分析表

软件名称	
需求 1	
需求 2	
需求 3	
.....	

(3) 请你上网查找大型项目需求分析的案例,从它们的成功和失败中,体会需求分析的重要性。

5.2 代码编写与调试

本节将针对“明日之星英语演讲大赛”问题,使用VB编程工具在计算机上实现编程。你将通过设计界面、编写代码和调试代码,最终编译生成在Windows环境中可直接运行的文件。通过本节的学习,你可以:

- 学会窗体的界面设计和菜单编辑器的简单使用
- 能分析选择合适的算法,书写正确的程序代码
- 掌握调试VB程序的主要工具和方法
- 学会编译生成可执行文件

5.2.1 设计界面

用户界面是软件非常重要的部分,它是人与计算机进行交互的主要途径。对用户而言,界面就是软件,他们感觉不到幕后正在执行的代码。

在设计软件界面时,我们需要考虑许多因素:需要哪些窗体、窗体如何布局、菜单与对话框的类型、如何显示操作提示等。

1. Splash 窗体

Splash 窗口是软件启动过程中闪现的窗口,即进入程序的欢迎界面。

Splash 窗体是运行软件的第一个窗口,设计要求美观大方、突出主题、有亲和力。Splash 内容通常包括软件的名称、版本号、版权所有等信息,如图5-7所示。Splash窗体使用的图片可以用图像处理软件事先处理好。



图 5-7 “明日之星” Splash 窗体

2. 用户登录窗体

用户登录窗体是确认使用者身份的窗口,如图5-8所示,要求输入用户名和密码,密码以“*”来显示,以满足软件的安全性要求。



图 5-8 用户登录窗体

3. 主窗体

主窗体是用户选择执行软件功能的地方,如图5-9所示。通常包括下拉菜单栏、工具栏、状态栏等。


在VB中如何制作Windows标准下拉菜单呢?操作步骤如下。



图 5-9 主窗体




图 5-10 菜单编辑器

(1) 单击VB工具栏中的“菜单编辑器”按钮,打开“菜单编辑器”对话框,如图5-10所示。

(2) 在菜单编辑器窗口的“标题”栏中输入“选手信息”。

(3) 在“名称”栏中输入“mnuInfo”,这时在菜单编辑器下面就显示出刚才输入的菜单项。

(4) 单击“下一个”按钮,输入“信息录入”菜单项,然后单击按钮将它调整为“选手信息”的子菜单项。

(5) 依次类推,输入其他菜单项。

工具栏、状态栏控件不是VB的标准控件,需要通过“工程”菜单中的“部件”命令进行添加。

(1) 选择“工程”菜单中的“部件”命令,打开“部件”对话框。

(2) 在控件列表框中选择“Microsoft Windows Common Controls 6.0”,单击“确定”按钮。

(3) 从工具箱中分别拖放一个工具栏控件(ToolBar)、一个图像列表框控件(ImageList)和一个状态栏控件(StatusBar),放在窗体中的相应位置,并设置控件的属性。

VB的常用控件参见教科书的附录三。

菜单的“标题”是显示在窗体中的内容,而菜单“名称”则在程序编码中引用。就好像标签控件的“标题(Caption)”与“名称”一样,两者不可混淆。

在菜单编辑器中,还可以添加菜单快捷键,像Word那样,按Ctrl+S组合键就可以保存文档。

4. 信息录入窗体

你会用 InputBox 语句录入数据吗? 与使用“文本框”控件录入数据相比, 它们各有什么特点?

信息录入窗体用于数据的输入, 其功能是将选手的参赛信息输入计算机, 如图 5-11 所示。选手的参赛信息通常包括“编号”、“姓名”、“班级”和“题目”等。当然, 你还可以增加其他需要录入的信息。



图 5-11 信息录入窗体

5. 版本信息窗体

加入世界贸易组织之后, 国务院出台了新的《计算机软件保护条例》。

版本信息窗体如图 5-12 所示, 通常包括软件名称、版本号、作者信息及软件说明等信息。利用版本信息一方面有利于保护软件的知识产权, 另一方面便于软件的升级维护。



图 5-12 版本信息窗体

VB 提供了“排列”、“水平/垂直间距”、“在窗体中央”等命令, 使得调整控件的间距、大小等非常容易。

版本信息窗体的设计, 需要注意界面美观、合理。具体地说, 就是首先应该注意以下几点。

(1) 控件的位置决定于它的重要性, 越重要的就放在越显著的位置上。还可以利用框架控件对功能相似的控件进行逻辑分组。

(2) 界面元素的一致性可以在软件中创造一种和谐, 使用户感觉非常舒服。如果界面缺乏一致性, 软件看起来非常混乱, 可能会引起用户对软件可靠性的怀疑。

(3) 在用户界面中使用空白空间有助于突出元素和改善可用性。空白空间并非只能是白色的——它是窗体与控件四周的空白区域。

(4) 适当地使用图片和图标可以增加软件视觉上的趣味性。

由于我们现在面对的不再是单个窗体而是多个窗体, 窗体上的对象也较多, 所以, 为了避免编写程序代码时将对象搞错, 按照表 5-4 所示设置各对象的名称属性。

表 5-4 “明日之星”软件的属性列表

对象	名称	说明
模块	subMain	存放程序的启动过程 Main 和定义一些全局变量
窗体	frmCover	Splash 窗体

续表

对象	名称	说明
窗体	frmLogin	登录验证窗体
窗体	mdiFrmMain	MDI 窗体，是程序的主窗体
窗体	frmCalc	现场计分窗体
窗体	frmHelp	系统帮助窗体
窗体	frmInput	信息录入窗体
窗体	frmOut	显示记录窗体
窗体	frmSort	排行榜窗体
窗体	frmAbout	版本信息窗体
菜单	mnuCalc	“现场计分”菜单项，单击打开现场计分窗体
菜单项	mnuHelp	“帮助信息”菜单项，单击打开系统帮助窗体
菜单项	mnuInput	“信息录入”菜单项，单击打开信息录入窗体
菜单项	mnuDisplay	“信息显示”菜单项，单击打开信息显示窗体
菜单	mnuSort	“分数排序”菜单项，单击打开排行榜窗体
菜单	mnuVersion	“版本信息”菜单项，单击打开版本信息窗体
菜单	mnuExit	“退出系统”菜单项，单击退出整个程序



- (1) 请你调查一下流行软件的版本信息窗体，谈谈你对它们的界面设计的想法。
- (2) 设计具有个性特色的“明日之星英语演讲大赛”版本信息窗体界面，同学之间进行交流，相互评价，填写表 5-5。

表 5-5 版本信息窗体界面设计评价表

评价标准	评价意见
控件位置安排是否合理	
界面风格是否一致	
空间留白是否合理	
色彩搭配是否恰当	
提示是否醒目	
……	

5.2.2 编写代码

前四章编写的代码只用到了一个窗体界面，编写的程序都是单窗体程序；本章涉及到多个窗体，需要编写多窗体程序。

整个程序是从模块 subMain 中的过程 Main 启动的，模块 subMain 中除

了有过程 Main 之外,还定义了一些整个程序都需要的变量,subMain 模块中全部的代码如下:

```
Public aryInfo(50, 5) ' 定义记录选手信息和分数的数组
Public intTotal As Integer ' 定义已得到分数的选手记录的个数
Sub Main()
    frmCover.Show ' 显示 Splash 窗体
    intTotal = 0
End Sub
```

多窗体程序的代码编写和单窗体程序的代码编写一样,但它们涉及到窗体与窗体之间切换的问题,这个任务通常在主窗体中完成。

下面是菜单项对应的代码:

```
' 以模态方式打开现场计分窗体
Private Sub mnuCalc_Click()
    frmCalc.Show vbModal
End Sub
```

```
' 以模态方式打开系统帮助窗体
Private Sub mnuHelp_Click()
    frmHelp.Show vbModal
End Sub
```

```
' 以模态方式打开信息录入窗体
Private Sub mnuInput_Click()
    frmInput.Show vbModal
End Sub
```

```
' 以模态方式打开信息显示窗体
Private Sub mnuDisplay_Click()
    frmOut.Show vbModal
End Sub
```

```
' 以模态方式打开排行榜窗体
Private Sub mnuSort_Click()
    frmSort.Show vbModal
End Sub
```

```
' 以模态方式打开版本信息窗体
Private Sub mnuVersion_Click()
    frmAbout.Show vbModal
End Sub
```

```
' 退出系统
Private Sub mnuExit_Click()
    Unload Me
End Sub
```

使用VB中的“Unload Me”命令可以关闭窗体;与Hide方法不同的是“Unload Me”是彻底关闭窗口,而Hide方法仅是隐藏,窗体仍在内存中。

从上面的代码可以看出，程序使用了“窗体.Show vbModal”方法，以模态显示子窗体，此时不能操作主窗体；当关闭子窗体后，主窗体又成为当前窗体，以此完成窗体的切换。

解决了多窗体的切换问题，我们还需要决定启动对象。默认情况下，应用程序中的第一个窗体为启动对象。在“明日之星英语演讲大赛”中，启动对象为 Splash 窗体(frmCover)，当程序运行时，单击它进入用户登录窗体，经密码确认后才能进入主窗体。改变应用程序启动对象的方法如图 5-13 所示。

(1) 单击“工程”菜单中的“Demo 属性”(Demo 为工程名)命令；单击“通用”标签。

(2) 在“启动对象”列表中选择 frmCover 作为启动对象，单击“确定”按钮。

多窗体程序有时需要全局变量以实现数据共享。在我们的“明日之星英语演讲大赛”中，需要用到一个二维数组 aryInfo()，用于存放参赛选手的信息及比赛成绩。

全局变量一般放在标准模块中，标准模块的添加方法如图 5-14 所示。

(1) 在工程管理器窗口中右击，选择“添加”→“添加模块”命令。

(2) 在“添加模块”对话框中选择“新建”标签，单击“打开”按钮，输入以下代码：

```
Option Base 1
Public aryInfo(50, 5) '定义记录选手信息和分数的数组
```

设计完程序总体结构后，下一步要实现每个功能模块。

在设计中，需要注意数据的存储问题，避免数据的重复录入。解决这一问题的简单方法是使用文件。

菜单项提供了窗体的切换功能，而具体功能的实现，我们将在各个子窗体的代码中实现。

信息录入模块中部分控件的名称如图 5-15 所示，下面给出信息录入模块中录入部分的参考代码。

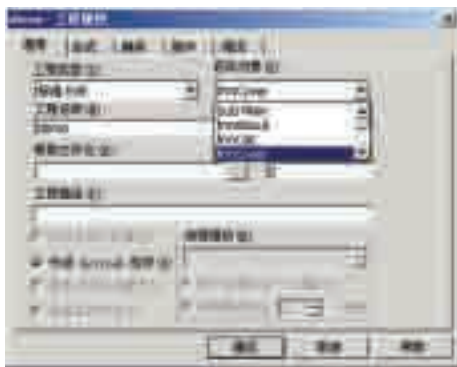


图 5-13 更改程序的启动对象

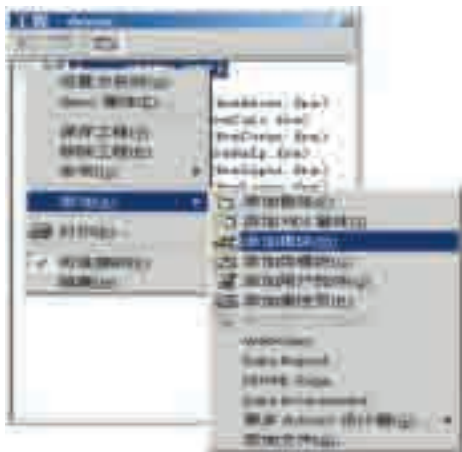


图 5-14 标准模块的添加



图 5-15 信息录入模块中各个控件的名称

VB 中除了有窗体模块外，还有标准模块(Bas)。标准模块没有界面，只包含代码。

```

Private Sub cmdSure_Click()
    Dim intNum As Integer
    Dim fileName As String
    On Error GoTo ErrorHandler
    '***** 以下进行数据校检 *****
    If Trim(txtNo.Text) = "" Then
        MsgBox "选手编号为空,请输入!", vbOKOnly + vbCritical,
"信息录入"
        txtNo.SetFocus
        Exit Sub
    End If
    If Val(Trim(txtNo.Text)) < 1 Or Val(Trim(txtNo.Text)) > 50 Then
        MsgBox "输入的号码无效,请重新输入!", vbOKOnly +
vbCritical, "信息录入"
        txtNo.SetFocus
        Exit Sub
    End If
    If Trim(txtName.Text) = "" Then
        MsgBox "请输入选手姓名", vbOKOnly+vbCritical, "信息
录入"
        txtName.SetFocus
        Exit Sub
    End If

    If Trim(txtClass.Text) = "" Then
        MsgBox "请输入选手班级!", vbOKOnly+vbCritical, "信
息录入"
        txtClass.SetFocus
        Exit Sub
    End If

    If Trim(txtTitle.Text) = "" Then
        MsgBox "请输入演讲的题目!", vbOKOnly+vbCritical,
"信息录入"
        txtTitle.SetFocus
        Exit Sub
    End If
    '*****

    '将选手信息读入数组
    intNum = txtNo.Text
    aryInfo(intnum, 1) = txtNo.Text
    aryInfo(intnum, 2) = txtName.Text
    aryInfo(intnum, 3) = txtClass.Text

```

```

aryInfo(intnum, 4) = txtTitle.Text

' 以下代码同时将选手信息写入文本文件中,以备下次使用
' 打开选手报名信息文件
fileName = App.Path+"\info.txt"
Open fileName For Append As #1 '想一想这条语句的作用是什么呢?
' 将信息写入文件
Write #1, aryInfo(intnum, 1), aryInfo(intnum, 2), aryInfo(intnum,
3), aryInfo(intnum, 4) '这条语句的功能是什么呢?
' 关闭文件
Close #1 '这条语句起什么作用呢?
MsgBox "选手信息已保存成功!", vbOKOnly+vbInformation,
"信息录入"
' 将录入文本框重新置空,以备下次录入
txtNo.Text = ""
txtName.Text = ""
txtClass.Text = ""
txtTitle.Text = ""
txtNo.SetFocus
Exit Sub

ErrorHandler:
MsgBox Err.Number & ":" & Err.Description, vbOKOnly +
vbExclamation, "错误提示"
End Sub

```

文件的种类

(1) 顺序文件。

顺序文件的结构比较简单,文件中的记录一条接一条地存放。在这种文件中,只知道第一条记录的存放位置。当要查找某条记录时,只能从文件头开始,一条记录一条记录地顺序读取,直到找到要查找的记录为止。

(2) 随机存取文件。

与顺序文件不同,随机文件的每条记录都有一个记录号,而且每条记录的长度都是固定的,在访问随机文件的数据时,不必考虑记录的排列顺序,可以根据需要访问文件中的任一记录。

文件的操作

在VB中,文件操作包括“打开文件”、“关闭文件”、“读出文件”和“写入文件”。

(1) 打开文件。

打开文件用 Open 语句,其格式如下:

```
Open 文件名 For 模式 As [#]文件号
```

Output: 顺序输出方式。

Input: 顺序输入方式。

Append: 顺序输出方式,与 Output 不同的是,如果对文件执行写操作,则写入的数据附加到原来内容的后面。

Random: 指定随机存取方式。

其中,文件名、文件号是必需的。文件名是一个字符串表达式;“模式”用于指定文件的打开方式,打开的方式有: Append、Input、Output或Random等。如果未指定打开方式,默认以Random访问方式打开文件。文件号为一个整型数值。

```
Open FileName For Append As #1
```

(2) 关闭文件。

当文件使用完后需要关闭。关闭文件用 Close 语句,格式如下:

```
Close [#文件号列表]
```

```
Close #1
```

可选的“文件号列表”参数为一个或多个文件号,多个文件号之间用逗号分隔,若省略“文件号列表”,则将关闭由 Open 语句打开的所有文件。

(3) 读文件。

读文件可以使用 Input 语句,其作用是从已打开的顺序文件中读出数据,并将数据指定给变量。其格式如下:

```
Input #文件号,变量列表
```

(4) 写文件。

写文件用 Write 语句,其格式如下:

```
Write #文件号,[表达式列表]
```

```
Write #1, aryInfo (intnum, 1), aryInfo (intnum, 2), aryInfo (intnum, 3), aryInfo (intnum, 4)
```

当有多个表达式时,需用逗号将表达式分隔开。



(1) 新建一个标准 EXE 工程,然后,用工程管理器将配套光盘中“明日之星英语演讲大赛”文件夹下的窗体模块和标准模块添加到你的工程中;运行程序,仔细体会应用程序的开发过程。

(2) “明日之星英语演讲大赛”软件还有许多不完善的地方,例如,没有对选手成绩进行保存,没有查找功能,选手人数固定等。根据你自己的分析和设计,参考第三章中的算法设计,改写“明日之星英语演讲大赛”软件。

(3) 请和同学们讨论,选取一个大家感兴趣的项目,合作开发一个实用小软件。

5.2.3 调试代码

无论多么仔细地编写代码,都可能会出现错误。在应用程序中,查找并修改错误的过程称为调试。

1. 错误类型

常见的错误类型分为三类:编译错误、运行错误、逻辑错误。编译错误是由于书写不正确的代码而产生的。例如,关键字书写错误,

遗漏了某些必需的标点符号，使用了一个For语句而没有对应的Next语句等。当VB编译应用程序时就会检测到这些错误。VB提供了自动语法检查功能，可以帮助我们减少这些错误。

运行错误是指应用程序运行期间，当一条语句试图执行一个不能

执行的操作时发生的错误。例如，当窗体上没有Label标签控件却要设置Label标签控件的属性，这时就会产生运行错误，如图5-16所示。

逻辑错误与前两种错误不同，逻辑错误一般不报告出错信息。从表面上看一切正常，但得到的结果却是错误的。这类错误往往最难发现，因此也最危险。减少和克服逻辑错误，没有捷径可寻，只有靠你的耐心、经验以及良好的编程习惯。

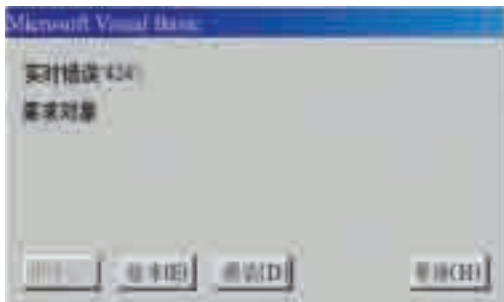


图 5-16 “运行错误”提示对话框

2. 调试工具与程序跟踪

VB提供了功能强大的调试工具栏，如图5-17所示。

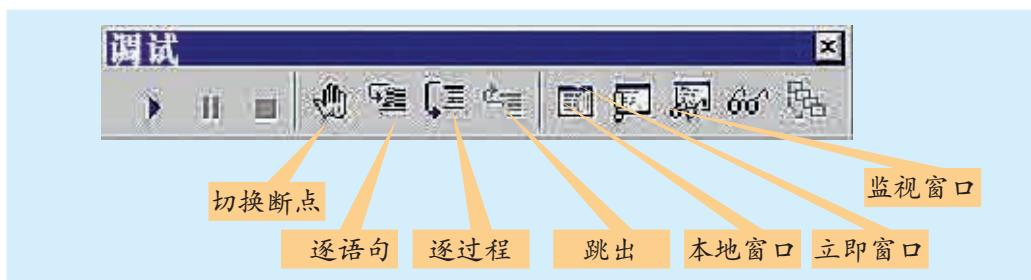


图 5-17 VB 代码调试工具栏

程序调试往往利用断点使程序运行到关键地方停住，程序就会自动进入中断模式；这时，就可以一部分一部分地对程序进行测试，观察程序的实际情况。利用断点，我们可以粗略地查出错误发生的大概位置，然后就可以“跟踪”程序的运行，找到发生错误的语句行。

设置断点非常简单，只要在程序代码窗口中把光标移到希望中断的语句上，然后单击调试工具栏中的“切换断点”按钮即可，或者直接按F9键。

VB中的跟踪方式可以用逐语句执行、逐过程执行和跳出执行。

逐语句执行就是程序运行后，在中断模式下单步执行程序语句，以此检测错误语句的位置。

逐语句执行的方法是：在中断模式下，单击调试工具条中的“逐语句”按钮或按F8键，程序执行一条语句，并把下一条语句作为待执行语句，待执行语句反相显示，如图5-18所示。逐语句执行时，可以用立即窗口、监视窗口等测试变量或表达式的当前值。

逐过程执行与逐语句执行类似，区别在于当程序执行到自定义过程(函

良好的编程习惯：

① 列出与应用程序有关的事件，认真定义每个事件过程和通用过程。

② 在关键的地方加上必要的注释。

③ 注意变量名称的一致性，可以强制所有变量显式声明。

设计模式：

启动VB后就进入该模式，此时，不能用调试工具。

运行模式：

选择“启动”后就进入该模式，此时，可以查看代码，但不能修改。

中断模式：

暂停程序执行就进入该模式，此时，可调试、修改代码。

数)时,将以自定义过程(函数)为单位执行,而不是只执行一条语句;在事件过程中,逐语句和逐过程没有区别。如果确信某个自定义过程(函数)不会出错,就可以用逐过程方式执行,从而加快调试速度。

跳出执行是指在逐语句执行到调用的过程时,使用跳出指令将从被调用的过程中跳出。

立即窗口可以用来输出当前的变量或表达式的值,方法为:

? 变量或表达式

当程序中包含语句“Debug.Print 表达式”时,其输出结果也显示在立即窗口中。

监视窗口能用来动态反映被监视表达式的当前值。



图 5-18 逐语句执行



请打开你编写的程序,综合运用各种调试方法找出程序中的错误,并将发现的错误归类。

5.2.4 生成可执行文件

最后要将编写的软件经过编译程序的编译生成可执行文件(.exe文件),这样才能使开发出的软件在 VB 的开发环境外运行,也只有这样才能保证代码的安全性。

生成可执行文件的操作步骤如下。

(1) 执行“文件”菜单中的“生成 demo.exe”命令,如图 5-19 所示。

(2) 输入将要生成的可执行文件名。默认的可执行文件名与工程名相同,其扩展名为.exe。如果不想使用默认名称,可以输入新的文件名。

(3) 单击“确定”按钮,即可生成可执行文件。



图 5-19 生成可执行文件对话框

5.3 软件测试与维护

虽然已经完成了界面的设计、代码的编写和调试,但我们的工作并没有全部完成。本节我们将要经历软件工程的最后两个阶段:测试与维护。通过本节的学习,你可以:

- 了解软件测试是发现软件中的错误和缺陷的主要手段
- 学会简单的软件测试方法
- 了解软件维护的概念和重要作用

5.3.1 软件测试

编程大师说：“任何一个程序，无论它多么小，总存在着错误。”

初学者不相信大师的话，他问：“如果一个程序小得只执行一个简单的功能，那会怎样？”

“一个这样的程序没有意义！”大师说，“但如果这样的程序存在的话，操作系统最后将失效，产生一个错误。”

初学者仍不满足，他问：“如果操作系统不失效，那么会怎样？”

“没有不失效的操作系统，”大师说，“但如果这样的操作系统存在的话，硬件最后将失效，产生一个错误。”

初学者仍不满足，再问：“如果硬件不失效，那么会怎样？”

大师长叹一声道：“没有不失效的硬件。但如果这样的硬件存在的话，用户就会想让那个程序做一件不同的事，这件事也是一个错误。”

1947年，被称为“编译器之母”的格蕾斯在用马克-2计算机进行一项计算时，突然机器停止了运行，她和同事在一个继电器中发现了一个死飞蛾，正是这个虫子造成了机器的故障，后来人们就把程序中的错误称为Bug（虫子），排除错误称为Debugging（除虫）。

1. 软件测试的目的

错误是一种程序缺陷，软件测试的目的就是为了发现尽可能多的缺陷，并期望通过改错来把缺陷全部消灭，以此提高软件的质量。这里的缺陷是一种泛称，它可以是功能性错误，也可以指性能低下、易用性差等。测试总是先假设程序中存在缺陷，再通过执行程序来发现并最终改正缺陷。

在软件的开发过程中，分析、设计与编码等工作都是建设性的，而测试是“破坏性”的，因为它是抱着“吹毛求疵”的目的，尽可能地寻找程序中可能存在的错误。如果你为了证明程序无错而进行测试，错误就可能在你的眼皮底下溜掉。

软件测试是一个为了寻找错误而执行程序的过程；

软件测试能证明错误的存在，但不能证明错误不存在；

一个成功的测试是指发现了迄今为止尚未发现错误的测试。



(1) 用VB 6.0新建一个标准EXE工程，打开Form1窗体，至少添加三个控件，关闭属性窗口，然后按住Ctrl键不放，同时反复单击这些控件，看看会发生什么？你认为这是不是VB的一个缺陷？

(2) 上网查找有关“冲击波病毒”的资料，然后和同学们讨论它是利用微软操作系统的什么缺陷给用户造成危害的。

(3) 你用的因特网浏览器是不是IE 6.0中文正式版，上网找找看它有没有小Bug。

2. 软件测试的方法

软件测试的方法和技术有很多,按设计测试用例是否涉及程序的内部结构分为白盒测试法和黑盒测试法,如图 5-20 所示。

白盒测试又称为结构测试、逻辑驱动测试或基于程序本身的测试。测试者从程序的逻辑结构入手,按照一定的原则来设计测试用例。所以被测试程序的结构对测试者是透明的、清楚的,如图 5-21 所示。白盒测试主要由程序员来完成。

黑盒测试又称为功能测试、数据驱动测试或基于规格说明的测试。测试者把被测程序看作成一个黑盒,完全用不着关心程序的内部结构。设计测试用例时,仅以程序的外部功能为根据,如图 5-22 所示。

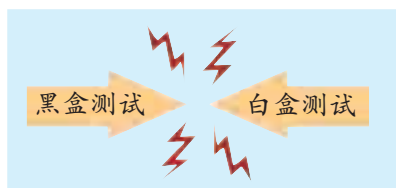


图 5-20 白盒与黑盒测试示意图



图 5-21 白盒测试



图 5-22 黑盒测试



请用适当的方法对“明日之星英语演讲大赛”软件进行测试,找出里面存在的缺陷,然后进行完善。

3. 软件测试的步骤

软件测试的步骤如图 5-23 所示。



图 5-23 软件测试的步骤

(1) 单元测试。

由设计开发人员自行测试,测试时针对单个或相关联的一组程序单元进行测试。

(2) 组合测试。

由开发人员与测试人员合作完成,针对不同单元以及它们的接口进行测试。

(3) 集成测试。

测试整个系统,以证实软件满足规定的功能、质量和性能等方面的要求。

5.3.2 软件维护

编程大师曾说:“哪怕程序只有三行长,总有一天你也不得不对它维护。”

你可能会问,软件产品在重复使用时并不像电器或车辆一样会磨损,为什么还要对软件进行维护呢?请先看看软件维护的定义。

软件维护就是在软件已经交付使用之后,为了改正错误或者满足用户新的需要而修改软件的过程。

软件维护包括对程序的维护、数据的维护、设备的维护等。一般来说,根据维护的目的,软件维护分为完善性维护、适应性维护、纠错性维护和预防性维护。

(1) 完善性维护:系统投入使用后,用户会提出增加新的功能、修改已有的功能等要求,例如,改变报表的输出格式,对用户权限进行分级等。为满足或部分满足这类要求而进行的软件维护都属于完善性维护。

(2) 适应性维护:由于软、硬件条件的变化,而对软件进行的修改称为适应性维护。

(3) 纠错性维护:也称改正性维护。对于一个大型软件来说,软件测试不可能找出系统中所有潜伏的错误,使用期间仍将可能发现这样或那样的错误,诊断和改正这类错误的过程称为纠错性维护。

(4) 预防性维护:为了进一步改进系统的可维护性和可靠性而对系统进行的修改称为预防性维护。



根据“明日之星英语演讲大赛”软件的功能,讨论需要进行哪方面的维护,并写出维护计划。

*第六章 综合实践



开发数据库软件



开发多媒体软件



开发网络应用软件

前面学习了程序设计与VB的知识，你是不是跃跃欲试了？本章你将领略VB广阔的应用领域，同时你还可以通过实践，自己动手设计并制作实用的小型应用程序软件，感受VB的巨大魅力。

6.1 开发数据库软件

翻开你的相册，回味昔日的快乐时光，是一件多么美好的事情。随着科技的进步，数码相机走入我们的生活，数码照片取代了传统的相片，你有没有想过利用学过的 VB 知识制作一个电子相册呢？

制作电子相册需要用数据库(Database)存储图片。VB提供了灵活的数据库链接和处理功能，我们只要编写少量的代码就可以方便地访问了。

许许多多的应用系统都需要数据库的支持，例如，学生管理系统、图书管理系统、销售管理系统等。利用VB进行数据库管理系统的开发也是非常方便的。

通过本节的学习，你可以：

- 掌握 Data 控件的使用方法
- 能借助 Data 控件开发简单的数据库应用系统

6.1.1 分析与设计

1. 需求分析

电子相册是我们日常生活中珍藏数码照片的好帮手。这个软件虽小，但要设计得让人满意，还是需要认真地做好需求分析工作。



请和你的同学、朋友、爸爸、妈妈一起讨论，在你的电子相册中，除了下面列出的需求以外还有哪些需求？把它填在下面的横线上。

①将数码照片保存到数据库中，以便于管理和浏览。

②可以浏览数据库中的数码照片，添加备注操作。

其他需求：_____。

2. 功能分析与设计

根据前面的分析，我们的“电子相册”需要借助数据库技术实现，VB中提供了Data控件，用它可以非常方便地对数据库中的图像数据进行访问。系统的主要功能如下。

(1) 添加数码照片功能：可以将计算机中的数码照片添加到电子相册数据库中。

(2) 数码照片浏览功能: 可以随时对数据库中的照片进行翻阅, 回味开心快乐的每一个时刻。

(3) “电子相册”支持多种图片格式: 能够保存的图片格式, 如.jpg、.bmp、.gif 等。

(4) 添加备注: 用于对照片进行必要的说明, 以帮助你回忆, 如图 6-1 所示。

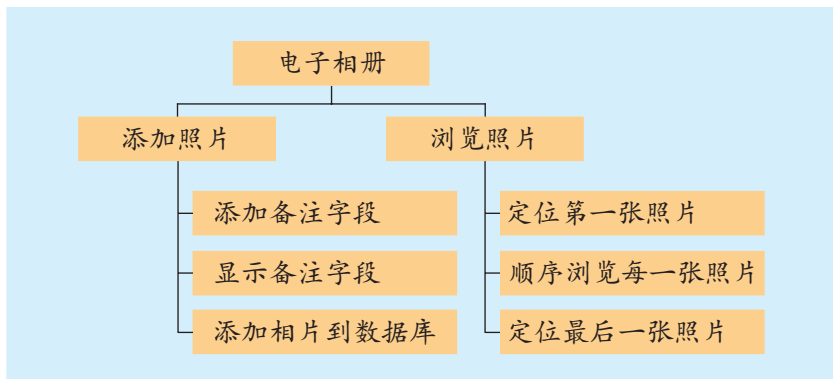


图 6-1 电子相册的模块设计图



将你分析的其他功能添加到图 6-1 所示的模块设计图中。

3. Data 控件介绍

VB 可以通过不同的方式与多种数据库进行连接, 最简单的方式是用 Data 控件, 如图 6-2 和图 6-3 所示。Data 控件在打开数据库时自动创建一个记录集 (Recordset), 通过对记录集的操作实现对数据库记录的添加、修改和删除等任务。



图 6-2 工具箱中的 Data 控件



图 6-3 窗体中的 Data 控件

下面我们学习 Data 控件的常用属性和方法。

(1) Data 控件的属性。

① DatabaseName: 确定 Data 控件使用的数据库名称。

② RecordSource: 指定 Data 控件所链接的记录源。记录源可以是数据库的表名, 也可以是一个查询。

(2) Data 控件的方法。

① AddNew: 用来添加一条新记录。例如, 给 Data1 的记录集添加一条新记录可以表示为: Data1.Recordset.AddNew。

② Delete: 用来删除当前一条记录。

③ Edit: 用来修改当前记录内容。

④ Refresh: 可以更新 Data 控件的记录集的内容, 以反映数据的最新变化。

⑤ Update: 用于将添加或修改的记录保存到数据库中, Update 需要和 AddNew、Edit 方法配对使用。

6.1.2 程序设计

1. 数据库设计

在“电子相册”中, 我们采用 Microsoft Access 数据库。

(1) 打开 Microsoft Access, 在弹出的对话框中选择新建一个“空 Access 数据库”, 单击“确定”按钮。

(2) 输入数据库的名称(db), 并设置保存路径(如“我的电子相册”), 单击“创建”按钮。

(3) 选择“使用设计器创建表”, 单击“设计”按钮, 在弹出的表结构窗口中, 按照图 6-4 设置相应的字段名称及其数据类型。

字段名称	数据类型
id	自动编号
pic	文本
memo	备注

图 6-4 创建 Access 表

id: 自动编号; pic: 存储图片文件的路径; memo: 备注字段, 存储图片的文字说明。

(4) 设置完成后, 单击窗口的“关闭”按钮。在弹出的对话框中输入表名“table”, 单击“确定”按钮, 建立如图 6-5 所示的表。

(5) 数据库格式转换。如果你是用 Access 2000 创建的数据库, 为了让 VB 的 Data 控件识别其格式, 需要用“工具”菜单中的“数据库实

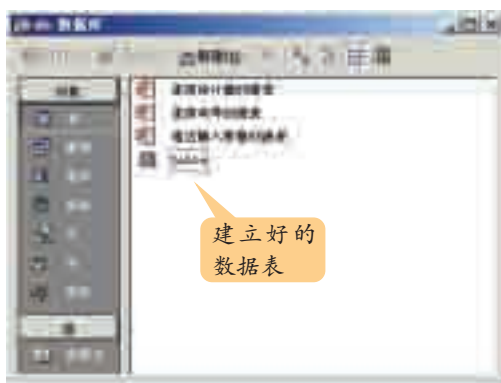


图 6-5 Access 表窗口

表(Table):
由多个同类型记录组成, 如“学生登记表”。

字段(Field):
一个数据项, 二维表的每一列。每个字段的名称叫做字段名。

记录(Record):
由多项数据组成, 二维表的每一行。每个记录的序号叫做记录号。

用工具”将其转换为早期 Access 数据库版本,如图 6-6 所示。



图 6-6 Access 转换数据库菜单

2. 窗体界面设计

新建一个标准 EXE 工程,在窗体 Form1 中添加一个 Data 控件、一个 TextBox 控件、一个 Label 控件、三个 Image 控件、两个 CommandButton 控件和一个 CommonDialog 控件。

界面设计如图 6-7 所示。

对象的名称、属性的设置值如表 6-1 所示。



图 6-7 程序界面及控件用途说明

表 6-1 对象设置

对象	名称	属性	设置值
窗体 Form	mainFrm	Caption	电子相册
图像框控件 Image	imgImage1	Picture	载入图片文件
	imgImage2	Picture	载入照片框文件
		Stretch	True
	imgShow	Stretch	True
命令按钮 CommandButton	cmdAddPhoto	Caption	添加新照片
	cmdReplace	Caption	替换照片
文本框控件 TextBox	txtMemo	Text	空白
标签控件 Label	lblAddMemo	Caption	说明
		BackStyle	0
数据控件 Data	Datal	Caption	浏览电子相册
对话框控件 CommonDialog	CommonDialog1	—	—

3. 程序代码的编写

设置完了各个控件的属性，我们来编写代码。编写完成之后，就可以运行程序了。

(1) 数据库链接与数据表链接。

```
Private Sub Form_Load()
    ' 动态链接数据库
    Data1.DatabaseName = App.Path & " \db.mdb "
    Data1.RecordSource = " table "
End Sub
```

(2) 得到图片文件名的通用过程。

```
Private Sub CopyFile(FileName As String)
    Dim strFileName As String, strSourceFile As String
    On Error Resume Next
    CommonDialog1.Filter= " 文件 bmp|.bmp| 文件 JPG|*.JPG| 文件 GIF|*.GIF "
    CommonDialog1.ShowOpen
    strFileName = CommonDialog1.FileName
    strSourceFile = CommonDialog1.FileName
    FileCopy strSourceFile, App.Path & " \mypics\ " & strFileName
    FileName = App.Path & " \mypics\ " & strFileName
End Sub
```

(3) 将照片数据添加到数据库中，并在窗体中显示照片。

```
Private Sub cmdAddPhoto_Click()
    Dim fm As String
    CopyFile fm
    If fm <> " " Then
        Data1.Recordset.AddNew
        Data1.Recordset.Fields( " pic " ).Value = fm
        Data1.Recordset.Fields( " memo " ).Value = Trim(txtMemo.Text)
        Data1.Recordset.Update
        imgShow.Picture = LoadPicture(fm)
    Else
        MsgBox " 没有选择照片! ", vbOKOnly + vbCritical, " 错误提示 "
    End If
End Sub
```

(4) 对数据库中的照片进行替换。

```
Private Sub cmdReplace_Click()
    Dim fm As String
    CopyFile fm
```

数据库的链接也可以在Data控件的属性设置中完成，分别对DatabaseName属性和RecordSource属性设置相应值。但这样链接的数据库的路径是固定的，一旦存放数据库的路径被改变，就会找不到路径，无法访问数据库。App.Path用来指明应用程序文件所在的位置。

```

If fm <> " " Then
    Data1.Recordset.Edit
    Data1.Recordset.Fields( " pic " ).Value = fm
    Data1.Recordset.Fields( " memo " ).Value = txtMemo.Text
    Data1.Recordset.Update
    imgShow.Picture = LoadPicture(fm)
Else
    MsgBox " 没有选择照片! ", vbOKOnly + vbCritical, " 错误提示 "
End If
End Sub

```

(5) Data 控件的 Reposition 方法, 用来判断数据库中的 pic 字段是否有值, 以决定 imgShow 控件中照片的显示。

```

Private Sub Data1_Reposition()
    If Not Data1.Recordset.EOF Then
        If Not IsNull(Data1.Recordset.Fields( " pic " ).Value) Then
            imgShow.Picture= LoadPicture(Data1.Recordset.Fields( " pic " ).Value)
        Else
            imgShow.Picture = LoadPicture( " " )
        End If
    End If
End Sub

```



比较cmdAddPhoto_Click()和cmdReplace_Click()两个事件过程代码的异同, 你能否将这两段代码改写成一個通用过程?

4. 调试程序并运行

按F5键运行程序, 如果程序有问题就根据相关错误信息提示进行更改, 如上面程序未考虑数据库为空、记录重复和文件重名等情况。反复调试直至通过, 编译生成一个可执行文件。然后试试自己亲手制作的电子相册吧! 程序的参考运行结果如图 6-8 所示。



图 6-8 “电子相册”运行结果

5. 保存文件

程序成功运行后,将窗体和工程保存在建立好的“我的电子相册”目录中。



(1) 与同学合作,收集家乡美丽的风景图片或者有趣的典故等,将这些信息存储在数据库中,利用我们学到的Data控件知识,设计一个小软件,制作家乡信息库来展示你可爱的家乡。

(2) “日月如梭,留下的是不变的记忆”,是否可以在VB中实现呢?与你的同学一起构想如何来创建这个电子日记簿。用你学到的软件工程方法和程序设计方法,进行分析与设计,完成代码的编写和调试。

(3) 通讯录是我们日常生活中与朋友联系最常用的文档,能不能在计算机中建立一个电子通讯录呢?将朋友们的联系方式输入到数据库中,在VB开发环境中对数据库中的数据进行操纵,实现简单的查询、修改等功能。试试看,对这个软件进行需求分析,确定功能模块,进行代码的编写与调试。

6.2 开发多媒体软件

多媒体技术在计算机领域占据着重要的地位，发展也非常迅速，越来越受到人们的关注。VB 给我们提供了多媒体控件，利用它可以方便地编写多媒体应用程序。本节通过制作一个多媒体播放器，来体会 VB 关于多媒体的处理机制。

通过本节的学习，你可以：

- 掌握多媒体控件 MediaPlayer 的使用方法
- 能够使用 MediaPlayer 控件编写简单的多媒体程序

6.2.1 分析与设计

请你和同学们一起讨论所使用的 MediaPlayer 软件有哪些基本功能，然后填写表 6-2。

表 6-2 MediaPlayer 基本功能

操 作	功 能
打开	打开所要播放的多媒体文件
播放	播放选中的多媒体文件

根据前面的讨论，能不能自己动手设计一个多媒体播放软件呢？要完成这个任务，需要用到 MediaPlayer 控件。

MediaPlayer 控件是 VB 6.0 中用于多媒体播放的扩展控件。使用这个扩展控件，需要先按以下步骤将它添加到工具箱中。

(1) 选择“工程”菜单中的“部件”命令，打开“部件”对话框。

(2) 在“控件”选项卡的对话框中选择“Windows MediaPlayer”，单击“确定”按钮，MediaPlayer 控件就出现在工具箱中了，如图 6-9 所示。

需要说明的是，本书是以 MediaPlayer 6.4 版本进行描述的，不同版本的 MediaPlayer 控件在工具箱中显示的图标是不同的，其属性与方法也有差异。



图 6-9 MediaPlayer 控件

下面学习 MediaPlayer 控件的基本属性和方法。

(1) MediaPlayer 控件的属性。

FileName: 用来指定要播放的多媒体文件名，可以在属性窗口中设置，也可以在运行期间指定。用 MediaPlayer 控件可以播放多种媒体文件，如

.avi、.mpg、.mp3、.wav、.mid 等。

(2) MediaPlayer 控件的方法。

① Play 方法：该方法用来播放多媒体文件。它的格式为：MediaPlayer1.Play。

② Pause 方法：暂停播放。它的格式为：MediaPlayer1.Pause。

③ Stop 方法：停止播放。它的格式为：MediaPlayer1.Stop。

VB 还可以提供其他多媒体控件。

(1) Microsoft Agent 控件：可以为应用程序添加动画人物。

(2) Multimedia MCI 控件：可以为应用程序添加声音和图像。

(3) Animation 控件：可以播放文件。

(4) RealAudio 控件：可以播放从因特网上下载的声音。

6.2.2 程序设计

1. 窗体界面设计

在这个小软件中，我们只需要设计一个主窗体就可以了，这个主窗体也就是我们的多媒体播放器的主界面，外观与 Windows 自带的媒体播放器有些相似，如图 6-10 所示。在这个窗体中，会用到 MediaPlayer 控件、一个文件菜单和一个对话框控件。新建一个“标准 EXE 工程”，按照表 6-3 所示在窗体中添加控件。



图 6-10 媒体播放器界面

表 6-3 控件的名称和属性

控 件	名 称	属 性	设 置 值
窗体 Form	mainFrm	Caption	媒体播放器
媒体播放器控件	MediaPlayer1	—	—
对话框控件	CommonDialog1	Filter	.avi、.mpg、.mp3、.wav、.mid

CommonDialog 控件的 Filter 属性值的设置决定了可以选取的文件的类型。



你成功地将“MediaPlayer 控件”添加到工具箱中了吗？使用同样的方法将“对话框控件”添加到工具箱中。

2. 菜单设计

执行“工具”菜单中的“菜单编辑器”命令，建立如图 6-11 所示的菜单，其属性如表 6-4 所示。



图 6-11 菜单编辑器

表 6-4 菜单编辑器中的属性

标题	名称	复选	有效	可见
文件	mnuFile	否	有效	可见
打开	mnuOpen	否	有效	可见
播放	mnuPlay	否	无效	可见
—	mnuBar	否	有效	可见
退出	mnuExit	否	有效	可见

3. 程序代码的编写

(1) 分别为“打开”、“播放”和“退出”菜单项建立 Click 事件，参考代码如下：

```
Private Sub mnuOpen_Click()
    On Error Resume Next
    CommonDialog1.ShowOpen
    If CommonDialog1.FileName <> "" Then
        MediaPlayer1.URL = CommonDialog1.FileName
        Me.Caption = CommonDialog1.FileTitle
        mnuPlay.Enabled = True
    End If
End Sub
```

```
Private Sub mnuPlay_Click()
    If MediaPlayer1.playState = wmppsPlaying Then
        MediaPlayer1.Controls.pause
    Else
        MediaPlayer1.Controls.play
    End If
End Sub
```

```
Private Sub mnuExit_Click()
    End
End Sub
```

(2) 为 MediaPlayer1 控件建立 PlayStateChange 事件代码，参考代码如下：

```
Private Sub MediaPlayer1_PlayStateChange(ByVal NewState As Long)
    Select Case NewState
        Case mpPaused, mpStopped
            mnuPlay.Caption = "播放"
        Case mpPlaying
            mnuPlay.Caption = "暂停"
    End Select
End Sub
```

4. 程序调试与运行

按 F5 键运行程序，打开一个视频文件或者是 MP3 文件，在窗体标题栏中可以看到正在播放的文件名，如图 6-12 所示。



图 6-12 程序运行结果

5. 保存文件

程序成功运行后，将窗体和工程保存在建立好的“媒体播放器”文件

夹中。

在前面的例子中，我们体验了使用 VB 提供的 MediaPlayer 控件开发应用程序的过程。除此之外，VB 还提供了许多多媒体控件，其中包括 VB 自身提供的内部控件，如图片框、图像框等，还有许多第三方提供的多媒体控件。如果你感兴趣，可以到因特网中搜索相关内容。



(1) 了解 VB 提供的多媒体控件 MCI 及其相关的属性和方法，试着开发一个 CD 播放器或者视频播放器。

(2) 每次看到 Windows 操作系统中删除或者移动文件时的画面，有没有想到，你也可以自己制作一个。学完本节后，利用 VB 提供的 Animation 控件试一试。

(3) 查阅相关资料，比较各个多媒体控件的功能特点，填写表 6-5。

表 6-5

多媒体控件	功能特点
MCI 控件	
Animation 控件	
MediaPlayer 控件	
ActiveMovie 控件	

6.3 开发网络应用软件

因特网是令人激动的，因特网将信息世界联系在了一起，VB 提供的内置网络功能，可以很容易地开发基于因特网和 Web 的应用程序。VB 可以使程序员搜索到整个世界的信息，并与其他计算机中运行的程序进行协作和通信，甚至可以使运行在同一计算机中的应用程序之间进行通信。

你对网络聊天一定不陌生吧？在聊天室里可以结交来自世界各地的朋友，交流彼此的思想，网络实现了沟通无阻碍。有没有想过自己设计制作一个网络聊天工具，和自己的朋友畅所欲言。本节我们将通过分析一个网上交谈软件的开发过程，学习 VB 中应用比较广泛的网络通信控件 Winsock (如图 6-13 所示)，介绍网络程序的编制方法。

通过本节的学习，你可以：

- 认识 Winsock 控件
- 能够使用 Winsock 控件编写简单的网络通信程序



图 6-13 Winsock 控件

6.3.1 分析与设计

1. 分析与设计

在这个例子中,我们将使用TCP(Transmission Control Protocol,传输控制协议)来演示一个客户端/服务器(Client/Server,简称C/S)应用程序。我们可以设想客户端就是一位学生,而服务器端是一位老师。如果在课堂中学生需要向老师提出问题,学生要先举手,以引起老师的注意,在老师的允许下学生提出问题,老师给学生解答。这个过程非常类似于客户端与服务器端信息的沟通机制。首先,服务器等待一个客户的连接请求,一旦建立了连接,则在双方的窗口显示消息,以说明连接成功;然后,客户端和服务器的用户可以通过在各自的文本框中键入并发送消息来进行通信;最后,发送的消息将出现在客户端与服务器的窗口中,并说明是谁发送的消息。

通过上面的分析,我们可以得到如图6-14所示的设计构图。

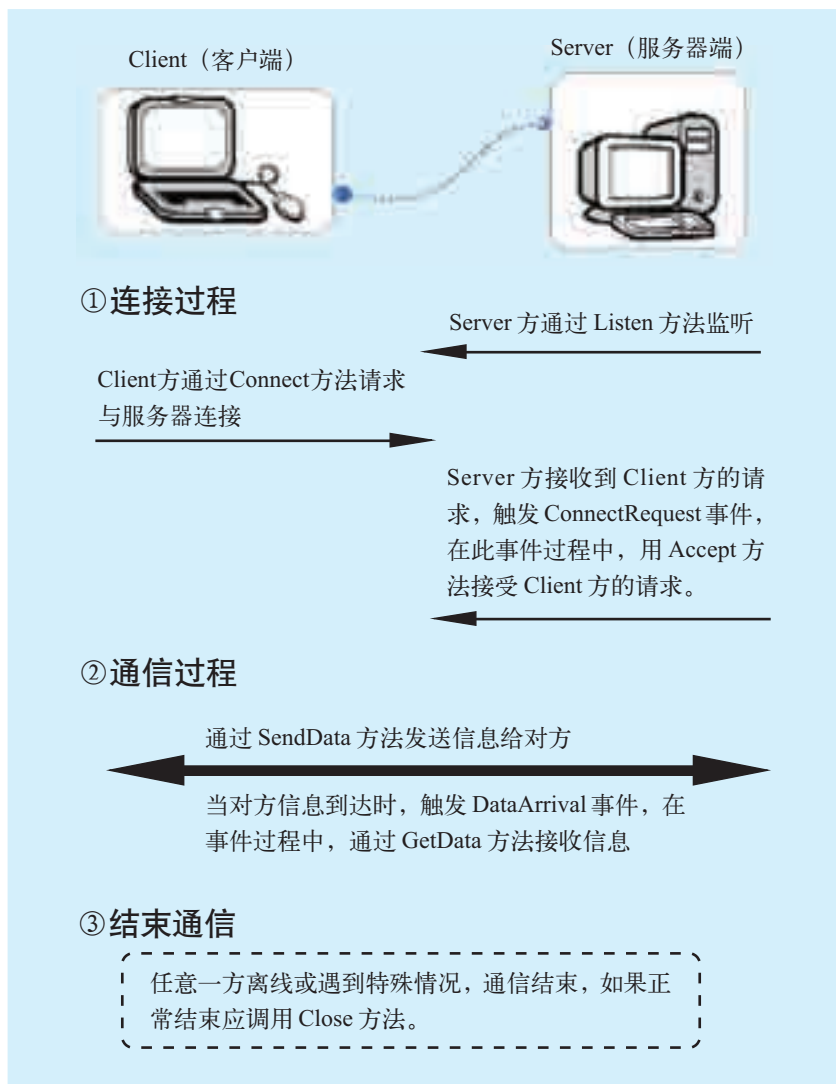


图 6-14 程序的功能结构

2. Winsock 控件介绍

Winsock 控件提供了访问 TCP 和 UDP(User Datagram Protocol, 用户数据报协议)的方便途径。利用 TCP 的通信类似于电话交谈,即在通信前必须先连接客户端与服务器端。TCP 通常指的是基于连接或面向连接的协议。利用 UDP 的通信类似于发送普通邮件,创建分组(也称数据包)、添加地址并通过网络发送,在源地址与目的地址之间没有直接连接。UDP 通常是指无连接协议。

当在服务器端与客户端之间要传输大量数据以及可靠性很重要时,一般使用 TCP 协议,这种连接保证了数据的完整性;当发送少量数据或间断地发送数据时可以使用 UDP 协议。

套接字(socket)使得应用程序能够像文件的输入/输出那样将网络看成字节流,即程序可以像读写文件那样读写套接字。套接字连接就像电话呼叫,电话机代表套接字,电话的讲话部分是输出流,接听部分是输入流。交谈(连接)将持续进行,直到通话的某端挂断电话。

下面让我们看一看 Winsock 控件的基本属性和方法。

(1) 属性。

- ① LocalHostName 属性: 返回本地计算机名。
- ② LocalIP 属性: 返回本地机器的 IP 地址。
- ③ LocalPort 属性: 返回或者设置所用到的本地端口。
- ④ Protocol 属性: 返回或者设置 Winsock 控件所使用的 TCP 协议或者 UDP 协议。
- ⑤ RemoteHost 属性: 返回或者设置远程计算机,控件向它发送数据或从它那里接收数据。

(2) 方法。

- ① Accept 方法: 仅适用于 TCP 服务器应用程序。在处理 Connection-Request 事件时用这个方法接收新连接。
- ② Close 方法: 关闭连接。
- ③ GetData 方法: 获取当前的数据块并将其存储在变量中。
- ④ SendData 方法: 将数据发送到远程计算机。
- ⑤ Listen 方法: 创建套接字并将其设置为侦听模式。该方法仅适用于 TCP 连接。
- ⑥ Connect 方法: 客户端连接到服务器端的方法。

6.3.2 程序设计

1. 窗体界面设计

由于网络通信建立在客户端和服务器端,所以我们需要建立两个窗体,一个用来表示客户端界面,另一个则用来表示服务器端的界面。这两个界面的布局和使用的控件可以是一致的。在这里,我们仅介绍客户端界面的设

计,如图6-15所示。客户端界面中需要使用一个 Winsock 控件、两个文本框控件(分别用来输入和显示消息)、一个标签控件和一个命令按钮。

控件的名称和属性的设置值如表6-6所示。



图6-15 客户端的界面设计

表6-6 控件的属性设置

控 件	名 称	属 性	设 置 值
窗体 Form	frmTCPClient	Caption	网络即时通 - 客户端
文本框控件	txtOutput	MultiLine	True
		ScrollBars	2
	txtSend	Text	空
标签控件	lblLabel1	Caption	欢迎来到网络聊天室
命令按钮	cmdSend	Caption	发言
Winsock 控件	tcpClient	Protocol	0

2. 程序代码的编写

(1) 在窗体 frmTCPClient 的 Form_Load() 事件中输入以下代码, 与服务器的计算机进行连接。

```
Private Sub Form_Load()
    ' 在建立连接前使 cmdSend 按钮无效
    cmdSend.Enabled = False
    ' 建立本地端口, 等待连接
    tcpClient.RemoteHost = InputBox("Enter the remote host IP address",
    "IP Address", "localhost")
    ' 如果没指定 IP, 将 RemoteHost 指定为本机 IP
    If tcpClient.RemoteHost = "" Then
        tcpClient.RemoteHost = "localhost"
    End If
    tcpClient.RemotePort = 1024           ' 服务端口
    tcpClient.Connect                     ' 连接到服务器
End Sub
```

(2) 与服务器端建立连接后, 与服务器端进行通信。

```
Private Sub tcpClient_Connect()
    ' 激活 cmdSend 按钮
    cmdSend.Enabled = True
    ' 在 txtOutput 文本框里显示服务器的 IP 地址和端口号
    txtOutput.Text = "Connected to IP address:" & tcpClient.RemoteHostIP &
vbCrLf & "Port #:" & tcpClient.RemotePort & vbCrLf & vbCrLf
End Sub
```

```
Private Sub cmdSend_Click()
    ' 发送数据, 同时说明消息是来自客户端的
    Call tcpClient.SendData(" 学生 说:》" & txtSend.Text)
    txtOutput.Text = txtOutput.Text & " 学生 说:》" & txtSend.Text &
vbCrLf & vbCrLf
    txtOutput.SelStart = Len(txtOutput.Text) ' 设置文本的插入点
    txtSend.Text = "" ' 将 txtSend 文本框置空
End Sub
```

```
Private Sub tcpClient_DataArrival(ByVal bytesTotal As Long)
    Dim message As String
    ' 当数据从服务器端到达客户端时, 执行此过程, 并将接收的数据保
    存在 message 字符串中
    Call tcpClient.GetData(message) ' 从服务器端获取数据
    txtOutput.Text = txtOutput.Text & message & vbCrLf & vbCrLf ' 显示文本
    txtOutput.SelStart = Len(txtOutput.Text)
End Sub
```

```
Private Sub tcpClient_Error(ByVal Number As Integer, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal
HelpContext As Long, CancelDisplay As Boolean)
    ' 如果在 Winsock 对象上发生了错误, 则执行 tcpClient_Error 事件过程
    Dim result As Integer
    result = MsgBox(Source & ":" & Description, vbOKOnly, "TCP/IP ERROR")
End
End Sub
Private Sub tcpClient_Close()
    cmdSend.Enabled = False
    tcpClient.Close ' 关闭客户端连接
    txtOutput.Text = txtOutput.Text & "Server closed connection ." & vbCrLf
    txtOutput.SelStart = Len(txtOutput.Text)
End Sub
```

(3) 终止应用程序调用 Form_Terminate() 过程, 该过程调用了 tcpClient.Close 来保证正确地关闭客户端。

```
Private Sub Form_Terminate()  
    tcpClient.Close  
End Sub
```

3. 程序调试与运行

按 F5 键运行程序, 这时弹出消息框, 如图 6-16 所示, 需要你输入远程计算机的 IP 地址。如果你没有输入相应的 IP 地址, 程序会自动填写使用 localhost。单击“确定”按钮后, 就进入“网络即时通”应用程序客户端界面, 如图 6-17 所示。这时, 需要你同时运行服务器端的应用程序, 在客户端和服务器端建立连接。一旦连接建立好之后, 就会显示相应的信息。接下来, 你就可以和朋友们畅快地聊天了。

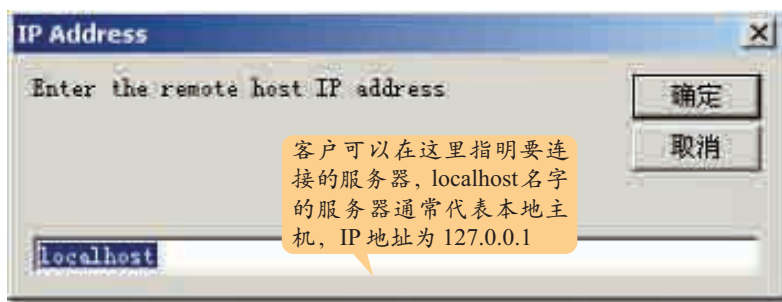


图 6-16 指定 IP 地址

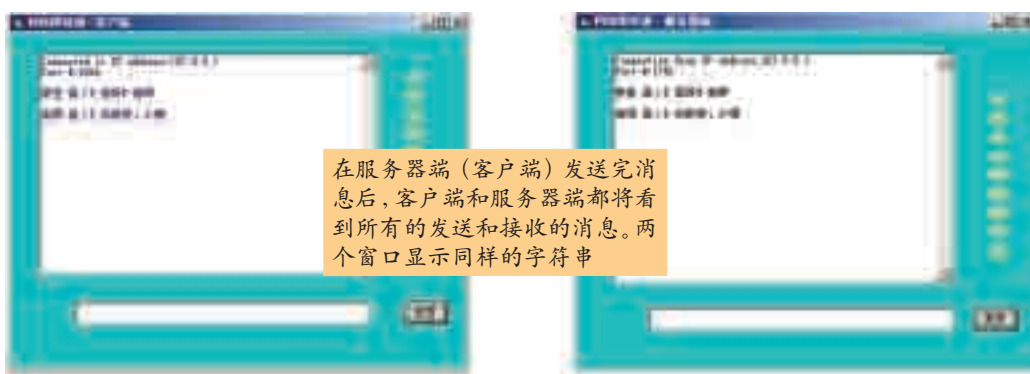


图 6-17 程序运行结果

服务器端程序与客户端程序的实现有些不同, 主要表现在服务器端关闭后, 在服务器端需要使用 Listen 方法侦听下一次连接, 如图 6-18 所示。另外, 当客户端连接到服务器时, 产生 ConnectionRequest 事件过程, 接收客户的请求。程序中的 requestID 参数表示客户请求连接的 ID 号, 用 Accept 方法接收一个请求连接。

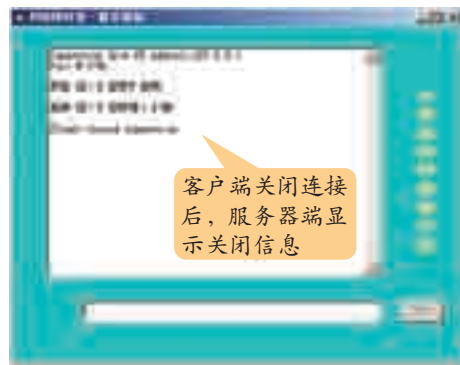


图 6-18 服务器与客户端

```

Private Sub cmdSend_Click()
    ' 发送数据到客户端
    Call tcpServer.SendData("老师说: 》" & txtSend.Text)
    txtOutput.Text = txtOutput.Text & "老师说: 》" & txtSend.Text & vbCrLf &
    vbCrLf
    txtSend.Text = ""
    txtOutput.SelStart = Len(txtOutput.Text)
End Sub

Private Sub Form_Load()
    ' 在建立连接前使 cmdSend 按钮无效
    cmdSend.Enabled = False
    ' 建立本地端口, 等待连接
    tcpServer.LocalPort = 1024
    tcpServer.Listen
End Sub

Private Sub Form_Terminate()
    tcpServer.Close
End Sub

Private Sub tcpServer_Close()
    cmdSend.Enabled = False
    tcpServer.Close    ' 客户端关闭连接, 服务器端也关闭
    txtOutput.Text = txtOutput.Text & "Client closed connection." & vbCrLf &
    vbCrLf
    txtOutput.SelStart = Len(txtOutput.Text)
    tcpServer.Listen    ' 侦听下一次连接
End Sub

Private Sub tcpServer_ConnectionRequest(ByVal requestID As Long)
    ' 在接收一个新的连接之前保证服务器是关闭的

```

```

If tcpServer.State <> sockClosed Then
    tcpServer.Close
End If
cmdSend.Enabled = True
Call tcpServer.Accept(requestID) '接收连接
txtOutput.Text = "Connection from IP address:" & tcpServer.RemoteHostIP &
vbCrLf & "Port #:" & tcpServer.RemotePort & vbCrLf & vbCrLf
End Sub

Private Sub tcpServer_DataArrival(ByVal bytesTotal As Long)
    Dim message As String
    Call tcpServer.GetData(message) '从客户端获取数据
    txtOutput.Text = txtOutput.Text & message & vbCrLf & vbCrLf
    txtOutput.SelStart = Len(txtOutput.Text)
End Sub

Private Sub tcpServer_Error(ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext
As Long, CancelDisplay As Boolean)
    Dim result As Integer
    result = MsgBox(Source & ":" & Description, vbOKOnly, "TCP/IP ERROR")
    End
End Sub

```

4. 保存文件

程序调试运行成功后,将客户端和服务器的窗体、工程保存在建立好的“网络即时通”文件目录中。



上文给出的程序代码并不是C/S结构的典型实例,一般情况下,网络环境中应该有一个Server和多个Client,试编写这样一个比较典型的C/S结构的应用程序。

附录

附录一

常用字符与 ASCII 代码对照表

ASCII 值 (十进制)	字符	控制字符	ASCII 值 (十进制)	字符	ASCII 值 (十进制)	字符	ASCII 值 (十进制)	字符
000	(null)	NUL (空白符)	032	空格符	064	@	096	`
001	☺	SOH (序始符)	033	!	065	A	097	a
002	●	STX (文始符)	034	"	066	B	098	b
003	♥	ETX (文终符)	035	#	067	C	099	c
004	◆	EOT (送毕符)	036	\$	068	D	100	d
005	♣	ENQ (询问符)	037	%	069	E	101	e
006	♠	ACK (应答符)	038	&	070	F	102	f
007	(beep)	BEL (告警符)	039	,	071	G	103	g
008	▣	BS (退格符)	040	(072	H	104	h
009	(tab)	HT (横表符)	041)	073	I	105	i
010	(line feed)	LF (换行符)	042	*	074	J	106	j
011	(home)	VT (纵表符)	043	+	075	K	107	k
012	(form feed)	FF (换页符)	044	,	076	L	108	l
013	(carriage return)	CR (回车符)	045	-	077	M	109	m
014	🎵	SO (移出符)	046	.	078	N	110	n
015	☀	SI (移入符)	047	/	079	O	111	o
016	▶	DLE (转移符)	048	0	080	P	112	p
017	◀	DC1 (设控 1)	049	1	081	Q	113	q
018	↕	DC2 (设控 2)	050	2	082	R	114	r
019	!	DC3 (设控 3)	051	3	083	S	115	s
020	⌣	DC4 (设控 4)	052	4	084	T	116	t
021	§	NAK (否认符)	053	5	085	U	117	u
022	⌞	SYN (同步符)	054	6	086	V	118	v
023	↕	ETB (组终符)	055	7	087	W	119	w
024	↑	CAN (作废符)	056	8	088	X	120	x
025	↓	EM (载终符)	057	9	089	Y	121	y
026	→	SUB (取代符)	058	:	090	Z	122	z
027	←	ESC (扩展符)	059	;	091	[123	{
028	└	FS (卷隙符)	060	<	092	\	124	
029	◆	GS (组隙符)	061	=	093]	125	}
030	▲	RS (录隙符)	062	>	094	^	126	~
031	▼	US (元隙符)	063	?	095	_	127	DEL

附录二

VB 常用函数

1. 类型转换函数

格 式	含 义
CBool (X)	将 X 转换为“布尔”(Boolean) 类型
CByte (X)	将 X 转换为“字节”(Byte) 类型
CCur (X)	将 X 转换为“货币”(Currency) 类型
CDate (X)	将 X 转换为“日期”(Date) 类型
CDbl (X)	将 X 转换为“双精度”(Double) 类型
CInt (X)	将 X 转换为“整型”(Integer) 类型
CLng (X)	将 X 转换为“长整型”(Long) 类型
CSng (X)	将 X 转换为“单精度”(Single) 类型
CStr (X)	将 X 转换为“字符串”(String) 类型
CVar (X)	将 X 转换为“变体型”(Variant) 类型
Int (X)	取 $\leq X$ 的最大整数值
Fix (X)	取 X 的整数部分，直接去掉小数

2. 常用数学函数

格 式	含 义
Abs (N)	取绝对值
Cos (N)	余弦函数，N 用弧度表示
Exp (N)	以 e 为底的指数函数
Log (N)	以 e 为底的自然对数
Rnd ()	产生 0~1 的随机数
Sin (N)	正弦函数，N 用弧度表示
Sgn (N)	符号函数，取正负号
Sqr (N)	平方根
Tan (N)	正切函数，N 用弧度表示
Atn (N)	反切函数

3. 字符串类函数

格 式	含 义
Asc (X)	返回字符串 X 的第一个字符的字符码
Chr (X)	返回字符码等于 X 的字符
Len (X)	计算字符串 X 的长度，空字符串长度为 0，空格符也算一个字符，一个中文字虽然占用两个字节，但也算一个字符

续表

格 式	含 义
Mid (X, n)	由 X 的第 n 个字符读起, 读取后面的所有字符
Mid (X, n, m)	由 X 的第 n 个字符读起, 读取后面的 m 个字符
Replace (X, S, R)	将字符串 X 中的字符串 S 替换为字符串 R
StrReverse (X)	返回 X 参数反转后的字符串。X="abc" P=StrReverse (X), 输出结果: P="cba"
Lcase (X)	将 X 字符串中的大写字母转换成小写
Ucase (X)	将 X 字符串中的小写字母转换成大写
InStr (X, Y)	从 X 第一个字符起找出 Y 出现的位置, 若在 X 中找到 Y, 则返回值是 Y 第一个字符出现在 X 中的位置。若 X 为空字符串, 或在 X 中找不到 Y, 则都返回 0。若 Y 为空字符串, 则返回 1
InStr (n, X, Y)	从 X 第 n 个字符起找出 Y 出现的位置, 若在 X 中找到 Y, 则返回值是 Y 第一个字符出现在 X 中的位置。若 X 为空字符串, 或在 X 中找不到 Y, 则都返回 0。若 Y 为空字符串, 则返回 n 的值

4. 日期时间类函数

格 式	含 义
Year (X)	取出 X “年” 部分的数值
Month (X)	取出 X “月” 部分的数值
Day (X)	取出 X “日” 部分的数值
Hour (X)	取出 X “时” 部分的数值, 数值是 0~23 之间
Minute (X)	取出 X “分” 部分的数值
Second (X)	取出 X “秒” 部分的数值
DateSerial (Y, M, D)	合并年、月、日成为日期, 其中 Y 是年份, M 为月份, D 为日期。M 值若大于 12, 则月份从 12 月起向后推算(M-12)个月; M 若小于 1, 则月份从 1 月起向前推算(1-M)个月。若日期 D 大于当月的日数, 则日期从当月的日数起, 向后推算(D-当月日数); 若 D 小于 1, 则日期从 1 日起向前推算(1-D)日
TimeSerial(H, M, S)	合并时、分、秒成为时间, 其中 H 为小时数, M 为分钟数, S 为秒数
Date()	读取系统的日期, 此函数无参数
Time()	读取系统的时间, 此函数无参数
Timer()	返回午夜以来已过的秒数
Now()	读取系统的日期时间, 此函数无参数
MonthName (X)	返回月份名称, X 参数可传入 1~12, 则返回值为“一月”、“二月”……但是在英文 Windows 环境下, 返回的是“January”, “February”……
WeekdayName (X)	返回星期名称, X 参数可传入 1~7, 则返回值依次为“星期日”, “星期一”……但是在英文 Windows 环境下, 返回的是“Sunday”, “Monday”……

1. 窗体 (Form)

属 性	说 明
Name	名称
Caption	标题
Enabled	决定对象是否响应用户生成事件
Visible	决定对象是否可见
BorderStyle	边框样式
Height	高度
Width	宽度
Left	对象左边缘与容器左边缘的距离
Top	对象顶端与容器顶端的距离
BackColor	背景色
ForeColor	前景色
Picture	显示的图形
MaxButton	窗体的最大化按钮
MinButton	窗体的最小化按钮
ControlBox	运行时, 在窗体上是否显示控件菜单栏
Icon	返回运行时窗体最小化所显示的图标
MDIChild	决定窗体是否显示为 MDI 子窗体
WindowState	设置 / 返回一个窗体窗口运行时的可见状态
AutoRedraw	True 时允许窗体图像在覆盖后的重建, 即自动做 Print, Cls, Pset, Circle, Line 等方法的重画
事 件	说 明
Click	单击
DbClick	双击
Load	窗体装入时触发, 可以用它来初始化需要的变量以及有关控件的各种属性
Activate	用户激活或程序中用 Show 方法显示窗体, 或 SetFocus 到该窗体
Unload	关闭操作
Paint	窗体重绘、移动、放大
方 法	说 明
Print	打印
Cls	清除窗体中用 Print 等方法写的内容
Move	移动
Load	装入窗体, 并不显示

续表

方 法	说 明
Unload	卸载已装入的窗体
Show 模式	显示窗体。0 模式时，鼠标可以出窗体操作其他窗体；1 模式时，鼠标出不去，关闭本窗体后，才可以出去操作
Hide	暂时隐藏窗体

2. 命令按钮(CommandButton)

属 性	说 明
Caption	标题
Enabled	决定对象是否响应用户生成事件
Visible	决定对象是否可见
Default	决定窗体的默认命令按钮
Style	控件的外观
Picture	显示的图形
事 件	说 明
Click	单击
KeyPress	字符键被单击
方 法	说 明
Drag	拖动
Move	移动
Refresh	刷新
SetFocus	设置焦点，将当前焦点强制设置到命令按钮对象上

3. 文本框 (TextBox)

属 性	说 明
BorderStyle	边框样式
Enabled	决定对象是否响应用户生成事件
Visible	决定对象是否可见
Font	字体
MultiLine	多行文本设置
Text	控件中包含的文本
ScrollBars	添加滚动条
事 件	说 明
Click	单击
DbClick	双击
Change	文本内容改变
GetFocus	获得焦点
LostFocus	失去焦点

续表

方 法	说 明
SetFocus	设置焦点，将当前焦点强制设置到文本框对象上
Move	移动
Refresh	刷新

4. 标签(Label)

属 性	说 明
Caption	标题
Alignment	对齐方式。0 表示左对齐；1 表示右对齐；2 表示居中对齐
BorderStyle	边框样式
AutoSize	自动尺寸，判断标签是否根据 Caption 属性值自动调整大小
WordWrap	换行显示，确定标签的 Caption 属性的显示方式是否换行

5. 列表框(ListBox)

属 性	说 明
Columns	列表的列数
MultiSelect	多重选择
ListCount	项目总数
ListIndex	项目数组下标
List()	项目数组
Selected	选中标志
SelCount	选中项目的数目
Sorted	指出控件是否自动按字母顺序排列
NewIndex	新加入项目的索引号
ItemData	项目编号
事 件	说 明
Click	单击
DbClick	双击
方 法	说 明
AddItem Item, index	在 index 指示的位置，将 Item 的内容添加到列表框中
RemoveItem index	删除位置索引号 index 指示的位置上的内容
Clear	清除列表框中的所有内容

6. 组合框(ComboBox)

属 性	说 明
Style	组合框的样式。0 下拉式组合框；1 只有简单组合框(列表一直显示)；2 下拉式列表框
Text	用户所选文本或编辑区输入文本
List	列表，是字符型数组
ListIndex	当前选中的序号，运行时读写

续表

属 性	说 明
ListCount	列表中总数
ItemData	项目编号
事 件	说 明
Click	单击
DbClick	双击
Change	文本内容改变

7. 图片框(PictureBox)

属 性	说 明
Picture	显示图片，不仅可以接受其他控件，用以制作工具条、工具箱等，还可以完成类似分组控件 Frame 的功能
AutoSize	决定容器是否能自动调整大小与显示的图片匹配
事 件	说 明
Click	单击
方 法	说 明
Print	打印
Circle	画圆
Line	画直线
Point	画点
Pset	画轨迹

8. 图像框(Image)

属 性	说 明
Picture	显示图片
Stretch	False 时，可根据图片调整控件的大小；True 时，根据控件的大小调整图片的大小
事 件	说 明
Click	单击
方 法	说 明
Print	打印
Circle	画圆
Line	画直线
Point	画点
Pset	画轨迹

9. 单选框(OptionButton)

属 性	说 明
Caption	标题
Value	表示单选框是否被选中

续表

事 件	说 明
Click	单击

10. 复选框(CheckBox)

属 性	说 明
Caption	标题
Value	0 表示复选框被选中; 1 表示复选框处于选中状态, 即前面的方框显示“√”
事 件	说 明
Click	单击

11. 分组框(Frame)

属 性	说 明
Caption	标题
Visible	决定分组框及其内部所有控件是否可见
Enabled	决定分组框内所有对象是否被屏蔽
事 件	说 明
Click	单击

12. 水平滚动条(HScrollBar)和垂直滚动条 (VScrollBar)

属 性	说 明
Value	对应于滚动块在滚动条中的位置
LargeChange	指定滚动块的移动量, 用于单击滚动条的情况
SmallChange	指定滚动块的移动量, 用于单击滚动条两端箭头的情况
事 件	说 明
Change	在滚动块移动后发生
Scroll	滚动事件, 在移动滚动块时发生, 在单击滚动箭头或滚动条时不发生

13. 定时器(Timer)

属 性	说 明
Enabled	确定定时器是否能对用户产生的事件做出反应
Interval	指定 Timer 事件之间的间隔, 单位为毫秒
事 件	说 明
Timer	告诉 VB 在每次 Interval 时间间隔到时该做什么

14. 驱动器列表框(DriveListBox)

属 性	说 明
Drive	显示计算机上与之相连的磁盘名

续表

事 件	说 明
Change	文本内容改变,每次重新设置驱动器列表框的 drive 属性时,都会引发 Change 事件

15. 目录列表框(DirListBox)

属 性	说 明
Path	显示当前工作驱动器下的目录和路径
ListIndex	目录列表框中的每个目录都对应一个 ListIndex 值
ListCount	记录了目录列表框中当前目录下的下一层共有多少个子目录
事 件	说 明
Change	文本内容改变,用户通过双击目录来打开或修改列表框中显示的当前目录,修改的路径影响到该控件的 Path 属性,系统自动将用户选中的新目录赋予 Path 属性中,同时引发 Change 事件

16. 文件列表框(FileListBox)

属 性	说 明
Pattern	设置在执行时要显示的某一类文件
FileName	返回或设置所选文件的路径和文件名
ListCount	返回控件的列表项目的个数
ListIndex	返回或设置控件中当前选择项目的索引
List	返回或设置控件的列表部分的项目

附录四

中英文术语对照表

(按英文字母顺序排列)

中文	英文全称或缩写	页码
算法	Algorithm	5
美国国家标准协会	ANSI(American National Standard Institute)	7
数组	Array	50
初学者通用符号指令代码	BASIC (Beginners All-Purpose Symbolic Instruction Code)	14
客户端 / 服务器	C/S(Client/Server)	117
类	Class	78
计算机程序	Computer Program	3
常量	Constant	21
数据结构	Data Structure	5
数据类型	Data Type	20
数据库	Database	105
封装	Encapsulation	81
事件	Event	15
事件驱动	Event Driven	19
表达式	Expression	25
字段	Field	107
流程图	Flow Chart	7
公式翻译语言	Formula Translator	12
继承	Inherit	82
方法	Method	19
微软开发者网络	MSDN(Microsoft Developer Network)	32
对象	Object	15
面向对象程序设计	OOP(Object-Oriented Programming)	19
运算符	Operator	25
多态性	Polymorphism	82
程序设计	Program Design	3
程序设计语言	Programming Language	3
属性	Property	16
伪代码	Pseudocode	8
记录	Record	107
递归	Recursion	65
套接字	Socket	118
表	Table	107
传输控制协议	TCP(Transmission Control Protocol)	117
用户数据报协议	UDP(User Datagram Protocol)	118
变量	Variable	22