

普通高中课程标准实验教科书

信息技术

选修 1

算法与程序设计

高一高二年级用



上海科技教育出版社

编写人员名单

主 编: 应吉康

主要编写人员(按姓氏笔画):

丁 祎 卢 源 朱晴婷
孙时敏 胡 杨 洪如蕙

上海科技教育出版社欢迎广大师生来电来函指出教材的差错和不足,提出宝贵意见。

上海科技教育出版社地址:上海市柳州路218号。

邮政编码:200235。

联系电话:021-64367970转202分机。

传真电话:021-64702835。

写给学生的话

亲爱的同学们：

你是否为在网上快速搜索到想要的资料而感到兴奋？为一部部奇幻电影的特效画面而感到惊叹？为广告设计师制作出的一幅幅精美广告而打动？课堂上，老师们使用生动的多媒体课件传授着知识；工厂里，一台台计算机精确地控制着生产流程；医院里，病人的CT分析报告正由计算机输出……这一切都离不开计算机。

计算机何以能如此神奇？其实都要归功于程序。可以说所有计算机应用技术，如多媒体技术、数据库技术、网络技术等都离不开程序。没有程序，计算机将只能成为一堆空壳，更无从造就这许许多多的神奇。

当你循着《算法与程序设计》课本中“活动”的足迹——

为学生电脑作品竞赛安排奖项和奖品，体验使用计算机解决问题的全过程；

为书城促销活动计算购书券，掌握程序设计的基本方法；

从趣味数学问题寻求答案中，体会各种算法程序实现的奥妙；

使用图形界面，设计时间显示器，了解最新的面向对象程序设计方法；

……

当你沉浸于计算机程序的无穷魅力时，一定希望自己能成为一名非凡的程序员，开发出各种实用的软件，发挥科学技术的无穷力量！

学习算法与程序设计，能够使你对计算机解决问题的过程有更为深刻的理解。

学习算法与程序设计，能够使你掌握积极而活跃的思维方法，体会“重在设计”的理念，大大提高分析和解决问题的能力。

算法与程序设计是一门实践性很强的学科，既充满乐趣也富有挑战性。单纯的传统的“学习”很难让你在这门学科中“登堂入室”。你必须亲自去设计，才能体会到该“如何设计”。

为使你在“做”中学算法与程序设计有更大的收获，课本里安排了这样一些栏目：



背景 每章开头都呈现了一些背景情况，表明需要解决的真实问题的来源，提出值得思考的问题。



活 动 以任务串的形式,呈现每一章中每一节的主体内容。在这一实践园地里,你与同伴将共同合作经历这样的过程:探索解决问题的方法,产生主动学习相应技术的欲望,尝试选择合适的手段和方法解决问题、完成任务……



提 示 板 在你解决问题、完成任务的过程中伴你同行,适时地在技术上向你提个醒或适当地为你释疑解惑。



学习指引 为你解决问题、完成任务提供技术知识的支持,为你形成技术能力和能力的迁移提供具体的帮助。



体验活动 为你提供不同程度的实践机会。你可以根据自己的需要作出选择,使自己的技术本领得到发挥。



后续活动 设在每章之后,为你与同伴综合运用相应技术合作解决问题,提供了实践机会。



评 价 是你关注自己学习过程的向导,有助于你感受探索的乐趣,总结学习技术、运用技术解决问题的经验,体会成功的喜悦。



参考资料 旨在拓展相关内容,反映程序设计语言的发展历程。

在这些栏目的帮助下,相信你一定能很快掌握算法与程序设计的基本原理和方法,为以后进一步学习打下坚实的基础。

编 者

目录



第 1 章 计算机解决问题的过程	1
第一节 了解计算机解决问题的过程	2
活动一 运用已有的知识解决问题	2
活动二 在计算机上解决问题	4
学习指引	5
■ 分析问题	6
■ 设计算法	7
■ 编写程序	7
■ 调试运行程序	7
第二节 算法和算法描述	9
活动一 认识算法	10
活动二 了解描述算法的三种方法	12
活动三 了解算法的作用	15
学习指引	18
■ 算法的概念	18
■ 算法的特征	18
■ 算法的描述	18
■ 算法在用计算机解决问题过程中的地位和作用	20
第三节 程序和程序设计语言	21
活动一 认识程序	21
活动二 了解程序的结构	23
活动三 了解程序的生成和执行	25
学习指引	26
■ 程序及其特征	26
■ 程序的基本结构	27
■ 程序设计	28
■ 程序设计语言的产生与发展	28
■ 程序的编辑和翻译	30
后续活动	32
评价	33
参考资料	33
第 2 章 程序设计基础	35
第一节 VB 的可视化开发环境	36
活动一 创建一个新的应用程序	36
活动二 了解属性窗口的使用	37
活动三 了解标签控件的使用	38
活动四 了解命令按钮控件的使用	39

学习指引	40
■ VB 的可视化开发环境	40
■ VB 的三种工作模式	43
■ 可视化编程	43
■ VB 可视化编程的一般步骤	46
第二节 VB 语言概述	48
活动一 观摩、分析程序语句	48
活动二 修改、完善程序	51
学习指引	53
■ 数据类型和常量、变量	53
■ 运算符、函数和表达式	55
■ 赋值语句与输入输出语句	58
■ VB 程序的调试	60
第三节 选择结构	63
活动一 用 If 语句实现简单选择结构	63
活动二 用 If 语句的嵌套实现多重选择结构	65
活动三 使用多重选择语句实现活动二的程序	66
学习指引	67
■ 控制条件	67
■ 实现选择结构的语句	69
第四节 循环结构	74
活动一 用 For...Next 语句实现循环结构	74
活动二 在循环结构中数组的应用	76
活动三 在循环结构中深入应用数组	78
活动四 用 Do...Loop 语句实现循环结构	81
学习指引	85
■ 循环结构	85
■ 实现循环结构的语句	86
■ 循环的嵌套	91
■ 数组	91
第五节 模块化程序设计	95
活动一 了解自顶向下、逐步求精的程序设计方法	95
活动二 使用函数和过程实现程序代码	101
学习指引	103
■ 自顶向下、逐步求精的程序设计方法	103
■ 过程和函数	104
■ 模块化设计的思想	107
后续活动	109
评价	112
第 3 章 常用算法及程序实现	113
第一节 解析法	114

活动 用解析法求解“铁丝问题”	114
学习指引	117
■ 解析法的基本概念	117
■ 用解析法求解问题的实践示例	118
第二节 穷举法	120
活动 用穷举法求解“金鱼问题”	120
学习指引	123
■ 用穷举法解决问题的基本思路	123
■ 用穷举法求解问题的实践示例	123
第三节 排序和查找	126
活动一 分析比赛成绩排序算法并编写程序	126
活动二 分析查找运动员比赛成绩的算法并编写程序	127
活动三 实现运动会管理的应用程序	128
学习指引	133
■ 排序算法	133
■ 查找算法	135
■ 变量的作用范围	136
第四节 递归法	140
活动 用递归法求解“兔子问题”	140
学习指引	142
■ 递归法	142
后续活动	146
评价	147
第4章 面向对象程序设计初步	149
第一节 初识面向对象程序设计思想	150
活动一 比较面向过程程序与面向对象程序	150
活动二 从VB的类模块中认识类、对象和实体	153
学习指引	155
■ 两种不同的程序设计方法	155
■ 面向对象程序设计思想的产生	156
■ 对象和类	157
第二节 面向对象程序设计	158
活动一 创建类	159
活动二 编写程序完成时间显示器的制作	162
学习指引	164
■ 对象、类和实例化	164
■ 类成员的访问	165
■ 面向对象的程序设计思想概括	166
后续活动	169
评价	169

附录	171
一、计算机软件及软件工程	171
二、部分名词术语中英文对照	176

• 江科教材网

第1章

计算机解决问题的过程



当 今社会，计算机已成为人们工作学习和日常生活不可或缺的工具。打字、画画、听音乐、看电影、玩游戏……计算机几乎渗透到了人们工作生活的所有领域，帮助人们完成各种任务，实现种种愿望。计算机之所以会有如此高超的本领，是借助了功能强大的各种不同的计算机程序。字处理程序帮助人们完成图文并茂的文稿编辑；绘图程序帮助人们制作出画面精美的作品；网页浏览程序帮助人们在因特网上纵横……甚至“神舟五号”上天、石油勘探入海都离不开计算机程序的支持。

功能各异的计算机程序是如何由程序设计人员用计算机语言予以实现的？

让我们一同步入计算机程序设计的殿堂，一起撩开计算机解决问题的神秘面纱，共同经历用计算机解决问题的过程，在分析问题、确定算法、编程求解等实践的体验中，感受计算机解决问题的精妙之道，汲取人类智慧的丰富养分，从而提高自己分析问题和解决问题的能力。

学习目标

- ★ 经历分析问题、确定算法、编程求解等用计算机解决问题的基本过程，了解过程中每一阶段的主要任务。
- ★ 了解算法的基本概念、特点，学会用自然语言、流程图或伪代码描述算法的方法。
- ★ 认识在用计算机解决问题的过程中，算法和程序设计的地位和重要作用。
- ★ 认识计算机程序的基本概念，能理解程序的执行顺序。
- ★ 了解程序设计和程序的编辑、编译、连接和执行的方法。
- ★ 了解程序调试的意义。

第一节 了解计算机解决问题的过程

学校即将组织学生电脑作品竞赛。竞赛将评出10名获奖者,分为一、二、三等奖,奖项设置一等奖获得者人数需少于二等奖人数,二等奖获得者人数需少于三等奖人数。学校还准备为获奖者颁发奖品。可购买奖品的金额共计1000元。奖品安排需满足一等奖奖品金额大于二等奖奖品金额,二等奖奖品金额大于三等奖奖品金额。奖品将在以下物品中选择:

- A1——电子词典256元/本; A2——可移动磁盘205元/只;
 A3——复读机182元/只; A4——名著138元/本;
 A5——文具73元/套; A6——书包55元/只。

竞赛组委会该如何确定各奖项的名额数及如何选择各奖项奖品?

孙强同学自告奋勇地协助老师解决各奖项的名额数和各奖项奖品确定问题。他略加思考,用Visual Basic语言编写了一段程序,很快就解决了问题。

孙强同学是怎么做的?

让我们试着与孙强同学一起做一做。

活动一 运用已有的知识解决问题

1. 分析问题。

分小组对上述问题进行讨论,将讨论的结果填入下表。

已知的条件	要解决的问题	它们之间的关系

提 示 板

解决任何问题都需要经历分析问题的过程。

分析问题就是从具体问题中明确条件和需要,找出条件和需要之间的关系,并尽量用数学语言给以描述。详见学习指引。

2. 列出数学表达式。

将讨论的结果用数学表达式表示:

(1) 确定一、二、三等奖的人数。

设一、二、三等奖的人数为 x, y, z ,

根据题目条件, x, y, z 应满足:

$$\begin{cases} x + y + z = 10 \\ 1 \leq x < y < z \leq 10 \end{cases} ; \quad (1.1)$$

(2) 确定奖品。

设一、二、三等奖的奖品分别为: A_i, A_j, A_k ; i, j, k 分别为相应奖品的序号。

根据题目条件, A_i, A_j, A_k 应满足:

(1.2)

解题目标是求出恰当的 x, y, z , 及 A_i, A_j, A_k 。

3. 写出求解步骤——算法分析。

解题步骤:

根据题意和分析方程(1.1)可知 x, y, z 是整数, 且满足 $1 \leq x, 2 \leq y, 3 \leq z$ 。采用穷举法, 在许可范围内, 即可求得满足条件的几组 x, y, z 的值;

根据题意和分析方程(1.2)可知奖品 A_i, A_j, A_k 的序号 i, j, k 也是整数, 且一等奖奖品 A_i 的序号 $i \geq 1$, 二等奖奖品 A_j 的序号 $j \geq 2$, 三等奖奖品 A_k 的序号 $k \geq 3$; 同样采取穷举法, 在许可范围内可求得满足条件的各奖项的奖品及数量。

具体的解题步骤如下:

第一步: 求出在 $1 \leq x, 2 \leq y, 3 \leq z$ 条件下的所有满足方程(1.1)的 x, y, z 的值;

第二步: _____;

第三步: _____

。

将上述算法再继续求精, 可写出更为详细的步骤:

第一步求精: 求出在 $1 \leq x, 2 \leq y, 3 \leq z$ 条件下的所有满足方程(1.1)的 x, y, z 的值;

(1) 对每一个 x ($x=1, 2, \dots, 10$), 做下一步;

(2) 对每一个 y ($y=2, 3, \dots, 10$), 做下一步;

(3) 对每一个 z ($z=3, 4, \dots, 10$), 做下一步;

(4) 逐一检查 x, y, z 是否满足 $x+y+z=10$, 如果满足, 则保存相应的 x, y, z 值;



第二步求精: _____;

结果是: _____

提 示 板

写解题步骤的过程也就是对算法进行分析的过程。

算法是解决问题方法的精确描述。

针对具体问题, 寻求解决问题的方法, 并进行精确描述, 称作设计算法。详见学习指引。

活动二 在计算机上解决问题

1. 输入程序。

(1) 孙强同学在确定了算法后,用计算机程序设计语言编写了下面的“确定各奖项名额数和各奖项奖品”程序,并输入了计算机。

```

Private Sub Form_Load()
Dim A(6) As Single
Dim p(10, 10) As Single
Dim x, y, z, I, J, K, f, m
Form1.AutoRedraw = True
For x = 1 To 10
    For y = 2 To 10
        For z = 3 To 10
            If x < y And y < z And x + y + z = 10 Then
                Print "一等奖" & x & "名", "二等奖" & y & "名", "三等奖" & z & "名"
                f = f + 1
                p(f, 1) = x
                p(f, 2) = y
                p(f, 3) = z
            End If
        Next z
        Next y
    Next x
    For I = 1 To 6
        A(I) = Val(InputBox("输入奖品" & I & "价格"))
    Next I
    For m = 1 To f
        For I = 1 To 6
            For J = 2 To 6
                For K = 3 To 6
                    If I < J And J < K And A(I) * p(m, 1) + A(J) * p(m, 2) + A(K) * p(m, 3) = 1000 Then
                        Print "一等奖奖品" & p(m, 1); "件, 价格为" & A(I)
                        Print "二等奖奖品" & p(m, 2); "件, 价格为" & A(J)
                        Print "三等奖奖品" & p(m, 3); "件, 价格为" & A(K)
                    End If
                Next K
            Next J
        Next I
    Next m
End Sub

```

提 示 板

此程序是孙强同学在经历了活动一的“分析问题”“设计算法”的基础上,用计算机程序设计语言编写而成。

编写程序就是用机器指令或机器能理解的语言将已确定的解决问题的算法描述出来。详见学习指引。

(2) 打开光盘中的“奖品方案.txt”，将其拷贝在 VB 软件环境的代码窗口中，见图 1-1。(或将“确定各奖项名额数和各奖项奖品”程序逐条输入。)

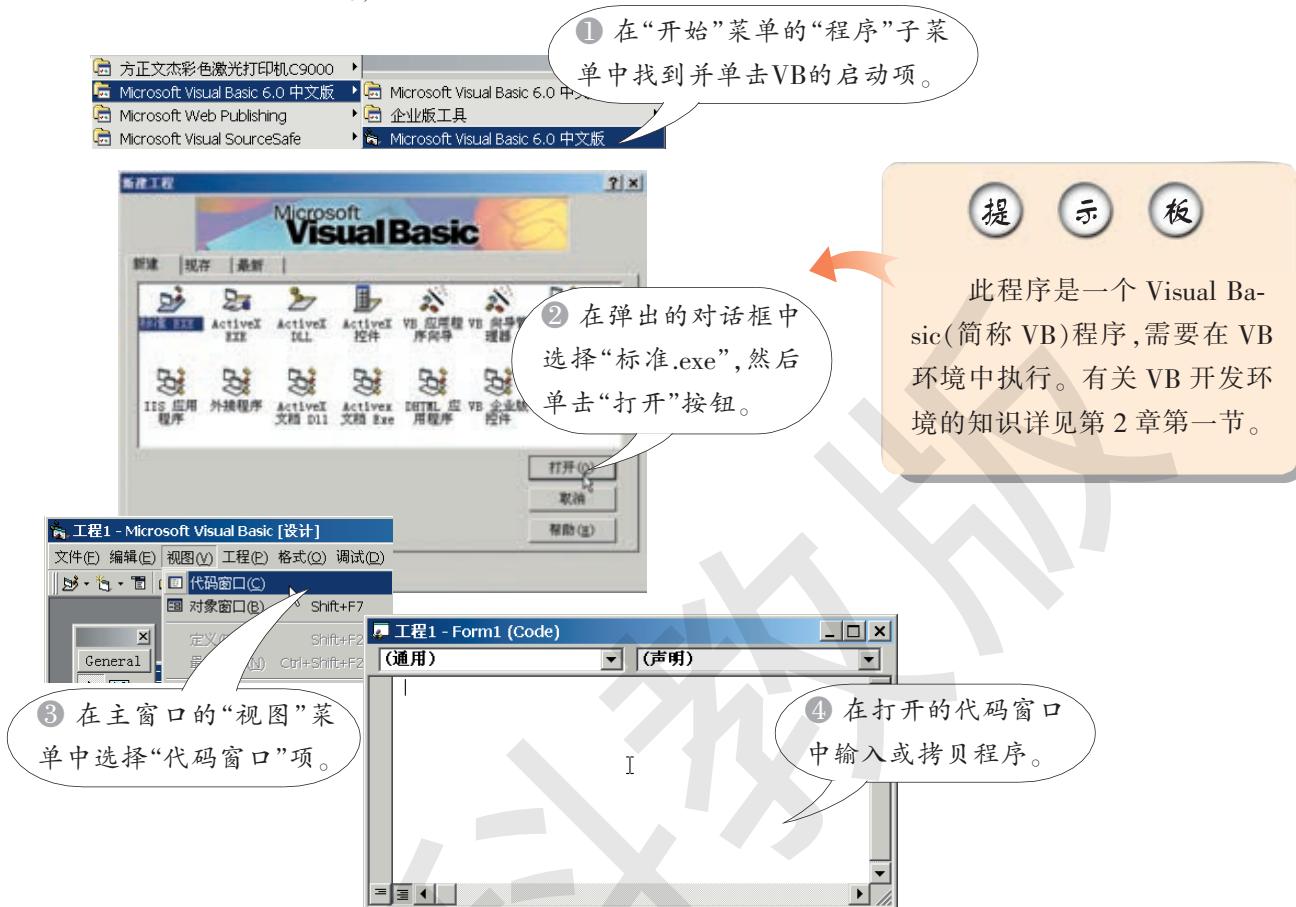


图 1-1 打开代码窗口

2. 调试运行程序并求解。

(1) 按 **F5** 键，启动运行程序。

(2) 分析验证结果。

结果是否符合问题解决的要求：

符合； 不符合，检查程序输入是否正确。

(3) 最后获得的结果：

各奖项名额数：_____，

各奖项奖品及金额：_____。

提 示 板

程序编写好后须验证其是否正确，亦即须进行调试。详见学习指引。



学习指引

跟随着孙强同学，我们实践了一回用计算机解决“确定各奖项名额数和各奖项奖品”的问题，亲历了从分析问题、设计算法、把所要解决的问题转化成的程序输入到计算机、经调试后让计算机执行这个程序，最终达到利用计算机解决问题的过程。

事实上,用计算机解决问题的过程,可以分为以下几个环节:

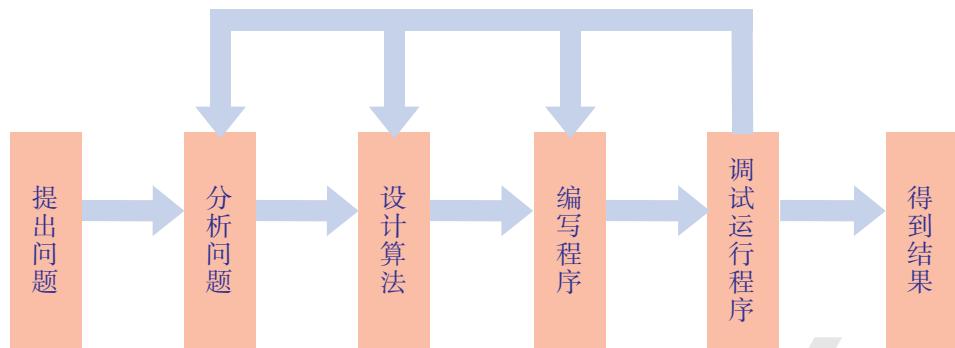


图 1-2 用计算机解决问题的过程

■ 分析问题

分析问题是用计算机解决问题过程中最基本也是最基础的工作。分析问题首先要弄清楚需要计算机解决什么问题。人们在生活、学习和工作中遇到的需要用计算机解决的问题不全是条件清楚准确、不多不少,结论唯一的那种情况。因此,需要找出对问题有影响的各种因素,确定解决问题需要考虑的因素,也就是要确定在什么范围内考虑要解决的问题。这在计算机软件工程中被称作问题定义,即明确解决问题需要考虑的已知信息和需要达到的目标。以下面的问题为例:

小磊欲购买全套《哈利·波特》,他在因特网上经过了一番搜索,得到的信息如下表所示:

	原价 (元)	网上书店 1	网上书店 2
哈利·波特和魔法石	19.20	7折	6折
哈利·波特和密室	22.00	6折	7折
哈利·波特和阿兹卡班的囚徒	26.50	7折	6折
哈利·波特和火焰杯	39.80	7折	7.5 折
哈利·波特和凤凰社	59.00	7.5 折送一本 18 元的图书	7.5 折送一本 14 元的图书
送货费		5元	4元

小磊到哪里去购书合适呢?

影响这一问题解决的因素,除了书的价格、折扣大小这些直接因素,还有其他诸如是否需要赠品、送货点是否在送货区域等因素。考虑这些因素与不考虑这些因素将导致解决这一问题的方法不同,获得的结果更是大相径庭。

总之,分析问题需要弄清问题的内容、性质、规模;收集与问题相关的资料,分析与问题有关的已知信息、该输入的数据,要达到的目标、要求输出的数据;确定所需解决的问题的功

能要求或性能要求等。

分析问题最好采用建立数学模型的方式。这一方式可以使问题的表达更清晰,更易理解。一般,在用计算机解决的问题中,有的问题其前提和目标都很清晰,求解的途径是已知的或容易获取的,能够通过形式化(或公式化)方法描述和求解,如本节的“确定各奖项名额数和各奖项奖品”问题即为一例。有些是非数值类的数据处理,无现成的公式可循,这时就需要依靠分析问题来构造模型。

■ 设计算法

经过对问题的分析,建立了数学模型,确定了应该做什么,接着就应该解决怎么做的问题,即寻找求解的方法和确定具体的求解步骤,亦即设计算法。

从数学课中已经知道,算法(algorithm)是指可以解决某一类问题的方法。它应该是精确的、确定的和有限的。算法给出怎么做的过程中的精确步骤,算法需用适当的方式给予准确的描述。

■ 编写程序

确定了“怎么做”后,就需要将其“转告”计算机。这项“转告”计算机“怎么做”的工作就是编写程序。

编写程序是要用一种计算机能接受的程序设计语言将求解问题的算法描述出来。这一过程犹如将中文翻译成外文,即将中文句子按选定的某种外文(如英语)的单词、语法和规则进行转换。程序设计语言同样有着自己的词汇、语法和规则,编写程序就是按照某一种程序设计语言的词汇、语法和规则将算法予以转换。

能被计算机接受的程序设计语言有很多,它们各有其比较适用的场合,如有的比较适合于编写网络上应用的程序,有的比较适合于编写科学计算程序,有的则适合于编写具有良好人机对话界面的应用程序。编写程序,一般先选定一种比较合适的程序设计语言,使程序编写比较方便,程序开发与执行的效率比较高。本节“确定各奖项名额数和各奖项奖品”程序是用Visual Basic语言编制的。VB语言是一种比较流行的方便开发的语言。

■ 调试运行程序

程序编写完,需要试运行,以便发现错误和修改错误。在程序设计工作中,无论是初出茅庐还是编程高手都免不了出错,因此调试程序是用计算机解决问题过程中一个非常重要的环节。

调试程序的任务就是发现错误,找到错误的原因和位置,并进行改正处理。得到运行结果后,还需要对结果进行分析和验证,判断是否正确、完整。若结果不合理,还需返回到先前几步进行修改,甚至重新经历从分析问题开始的工作过程。

综上所述,我们能感受到计算机可以帮助我们解决问题,但计算机不能自动地自己解决问题。人参与完成了从分析问题、设计算法、到把所要解决的问题转化成计算机程序等一系列的工作,这就是程序设计工作,而计算机则是按人们的意愿及安排好的步骤执行程序,完成任务。



体验活动

1. 讨论与交流。

比较人工求解问题和用计算机求解问题的异同。

求解方式	经历的过程	内容参考
人工求解		<input type="checkbox"/> 分析问题 <input type="checkbox"/> 建立数学模型(列数学表达式) <input type="checkbox"/> 设计算法 <input type="checkbox"/> 人工计算 <input type="checkbox"/> 编写程序 <input type="checkbox"/> 调试程序 <input type="checkbox"/> 执行程序
用计算机求解		

人工求解的特点:_____。

用计算机求解的特点:_____。

两种方法的相同之处:_____。

两种方法的不同之处:_____。

2. 尝试分析问题和设计算法。

(1) 问题:某家用电器厂根据市场调查分析,决定调整产品生产方案,准备每周(按 120 个工时计算)生产空调器、彩电、冰箱共 360 台,且冰箱至少生产 60 台。生产这些家电产品每台所需工时和每台产值如下表:

家电名称	空调器	彩电	冰箱
工时	1/2	1/3	1/4
产值(千元)	4	3	2

问:每周应生产空调器、彩电、冰箱各多少台,才能使产值最高?最高产值是多少(以千元为单位)?

- ① 分析问题,找出已知和未知,明确它们之间的关系,列出数学表达式;
- ② 写出解题过程;
- ③ 尝试写出解题的算法。

(2) 笼子里有鸡和兔共 50 只,只知道鸡和兔子的腿共有 160 条,问鸡和兔各有多少只?

- ① 分析问题,列出数学表达式;
- ② 写出解题的算法。

> 接上页

(3) 输入一个自然数 M , 判断这个自然数是否是素数。如果是素数, 则输出“这是一个素数! ”, 否则输出“这不是一个素数! ”。

- ① 分析问题, 建立数学模型;
- ② 写出解题的算法。

(4) 有两个一样大小的瓶子 A 和 B, A 瓶中放着红球, B 瓶中放着黄球, 现在要求互换两个瓶中的小球, 使得 A 瓶中放黄球, B 瓶中放红球。

- ① 分析问题, 建立数学模型;
- ② 写出解题的算法。

第二节 算法和算法描述

秦九韶是我国古代大数学家, 他的传世名著《数书九章》是我国 13 世纪数学成就的代表作之一。

《数书九章》是一部有二十多万字的科学巨著。书中共分九大类, 列出 81 道题。有趣的是, 从作者的名字、书名到题目共四个九, 即“九韶”、“九章”、“九类”、“九题”。这部书的每一题都有“术”, 即都有解题的原理和解题步骤, 它继承了我国数学发展的突出特色: 算法化。这部书的另一重要特色是, 理论联系实际。书中大多数问题都来自实际, 如, “九类”中的大衍、天时、田域、测望、赋役、钱谷、营建、军旅和市物方面的问题。秦九韶对当时的生产和生活的各种问题进行了深刻的思考和分析, 并抽象为数学问题, 进而研究这些问题的算法。这部凝聚着秦九韶艰辛劳动的名著, 在中国和世界数学史中具有重要的地位。值得称道的是, 秦九韶创造的一些算法至今仍是世界上最好的算法。以秦九韶的一项重要成果“三斜求积公式”为例, 其题意是: 已知三角形地块的三边长分别为 13 步、14 步、15 步, 求它的面积。秦九韶的解法用现代的符号表示, 就是: 设三角形的三边长分别为 a 、 b 、 c , 则面积为:

$$S = \sqrt{\frac{1}{4} \left[a^2 b^2 - \left(\frac{a^2 + b^2 - c^2}{2} \right)^2 \right]}$$

秦九韶的算法与古希腊数学家海伦研究的结果有异曲同工之美。学生们熟悉的海伦—秦九韶公式, 其具体过程如下:

设三角形的三边长分别为 a 、 b 、 c , 面积为 S ,

$$(1) \text{ 计算 } p = \frac{a + b + c}{2};$$

$$(2) \text{ 计算 } S = \sqrt{p(p - a)(p - b)(p - c)}.$$

当我们享受着用现成的算法解决一个个具体问题的便捷之时, 我们感受到算法对于具体问题的解决是何其重要且美妙。

活动一 认识算法

1. 阅读几个问题解决的算法。

(1) 渡河问题——一个趣味数学问题及其算法。

提 示 板

算法是解决问题过程所需的有限步骤。

算法的每一个步骤都应该是有明确意义、可以执行的,每个步骤的执行顺序是确定的,并且能够在有限步骤内执行完毕。

问题

两个大人和两个小孩一起渡河。渡口只有一条小船,一次只能渡过一个大人或两个小孩,他们四人都会划船,但都不会游泳。他们怎样渡过河去?请写出你的渡河方案。

设计算法

渡河的方法与步骤:

- 第1步:两个小孩同船渡过河去;
- 第2步:一个小孩划船回来;
- 第3步:一个大人独自划船渡过河去;
- 第4步:对岸的小孩划船回来;
- 第5步:两个小孩再同船渡过河去;
- 第6步:一个小孩划船回来;
- 第7步:余下的一个大人独自划船渡过河去;
- 第8步:对岸的小孩划船回来;
- 第9步:两个小孩再同船渡过河去。

(2) 个人理财问题。

问题

现今社会,个人理财问题已受到很多市民的关注。存款、国债、股票、黄金产品都是市民理财的内容。随着存款加息周期的到来,市民越来越关心存款利息的收益。某一时期,银行一年期定期储蓄年利率为2.25%,如果存款到期不取继续留存,银行会根据存款时约定的转期自动将本金及80%的利息(20%利息缴纳利息税)转存为一年期定期储蓄。

某人以一年期定期储蓄存入银行20万,那么3年后,这笔钱款扣除利息税后的本利和是多少?

分析问题

设:本金为A;银行一年期定期储蓄年利率为R;存款时间为T;扣除利息税后的本利和为P。

则,一年后的本利和为: $P_1 = A \times (1 + R \times 80\%)$;

二年后的本利和为: $P_2 = P_1 \times (1 + R \times 80\%)$;

三年后的本利和为： $P_3 = P_2 \times (1 + R \times 80\%)$ 。

设计算法

上述问题的算法如下所示：

- ① 输入 A、R、T 的值；
- ② 令 $I=0$ ；
- ③ $P=A$ ；
- ④ 如果 $I < T$ ，则执行⑤，否则执行⑦；
- ⑤ $P=P \times (1+R \times 80\%)$ ；
- ⑥ $I=I+1$ ，转④执行；
- ⑦ 输出结果 P ；
- ⑧ 结束。

(3) 找出最大数的问题。

问题

找出十个数中最大的数。

分析问题

找出十个数中最大的数，可以采取如同打擂台的办法，即台上先站一个人，第二个人上台与之比较，胜者留在台上，然后第三个人上台再与台上的人（胜者）比较，胜者留在台上……比完九轮之后，在台上的人便是最后的胜者。即

- ① 将输入的第一个数存放在存储单元max中；
- ② 将输入的第二个数与之相比较，大者占据max；
- ③ 然后第三个数再与max相比，大者占据max；
- ④
- ⑤ 一直到比完九轮为止。

设计算法

- ① 输入一个数，放在存储单元max中；
- ② 设置用来累计比较次数的计数器i，初始值为1；
- ③ 输入一个数，放在存储单元x中；
- ④ 比较max和x中的数，若 $x > max$ ，则把x的值赋给max，即令 $max=x$ ；否则max的值不变；
- ⑤ i增加1，即 $i = i + 1$ ；
- ⑥ 若 $i \leq 9$ ，转第③步执行，否则输出max中的数，此时max中的数就是10个数中的最大数；
- ⑦ 结束。

2. 分析算法。

提 示 板

用计算机解决问题时,算法确定了,解题的方法和过程也就确定了。

一个可以用计算机解决的问题的算法应具有五个基本特征。详见学习指引。

分析算法	个人理财问题	找出最大数问题
各执行了多少步		
每一步是否都有确切的含义		
各有多少个输入		
各有多少个输出		
是否都有结果		

活动二 了解描述算法的三种方法

1. 阅读用自然语言描述的算法。

提 示 板

此例及前面出现过的个人理财问题、找出最大数问题等算法都是用自然语言描述的。**自然语言**就是人们日常使用的语言。

除了自然语言,描述算法的方法还有**流程图**和**伪代码**。详见学习指引。

- ① 输入X;
- ② 若 $X < 0$,则执行③;否则执行⑥;
- ③ 令 $Y = -X$;
- ④ 输出Y;
- ⑤ 结束;
- ⑥ 若 $X = 0$,则执行⑦;否则执行⑩;
- ⑦ 令 $Y = 0$;
- ⑧ 输出Y;
- ⑨ 结束;
- ⑩ 令 $Y = X$;
- ⑪ 输出Y;
- ⑫ 结束。

上述算法解决的问题是:_____。

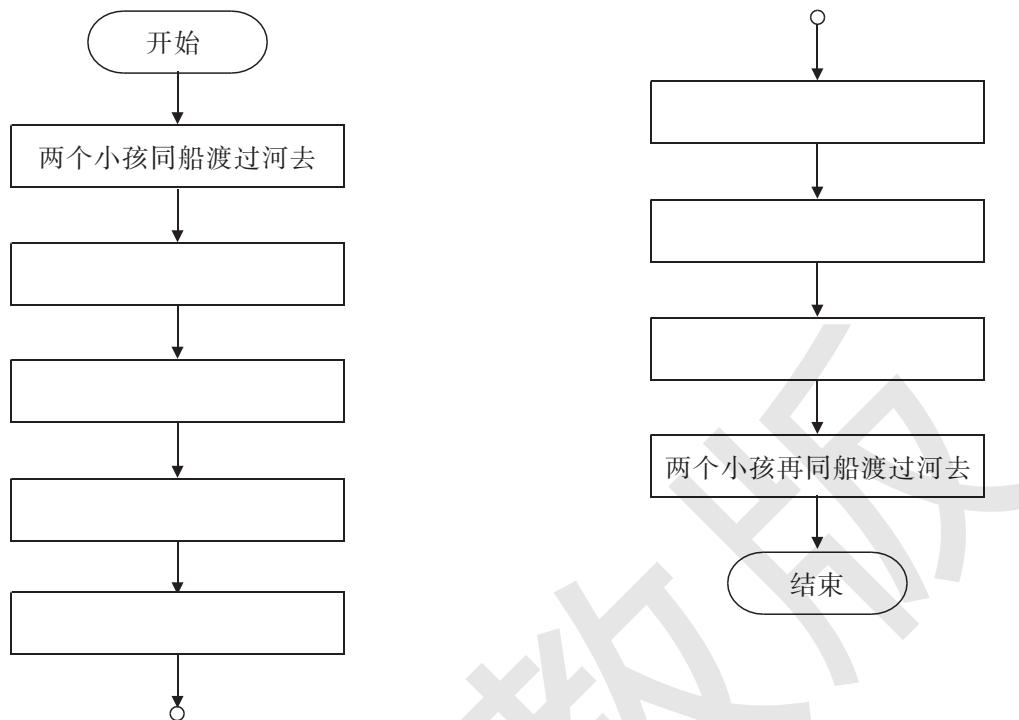
2. 认识流程图常用的基本图形及其功能。

用数学课中学过的知识,或通过上网检索,梳理流程图常用的基本图形及其功能,并填写在下表内。

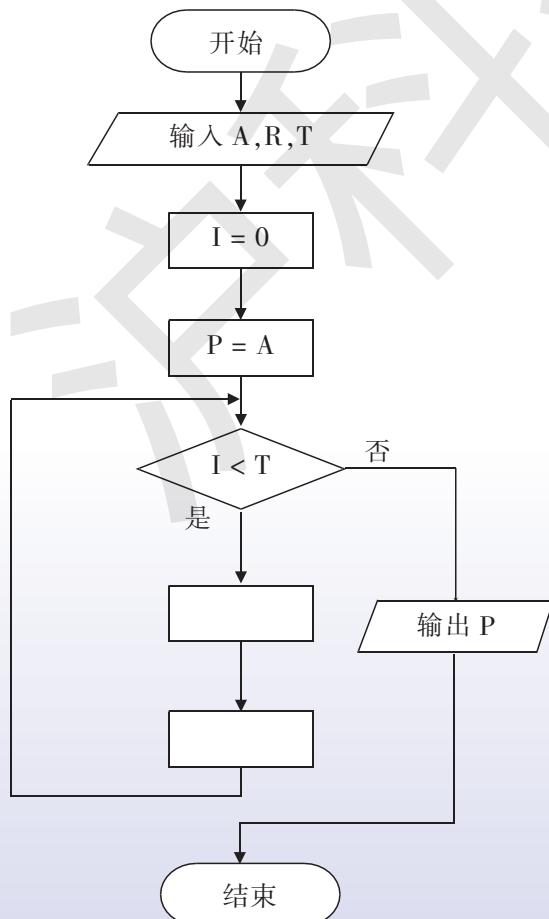
图形符号	符号名称	功能
○		

3. 试用流程图描述算法。

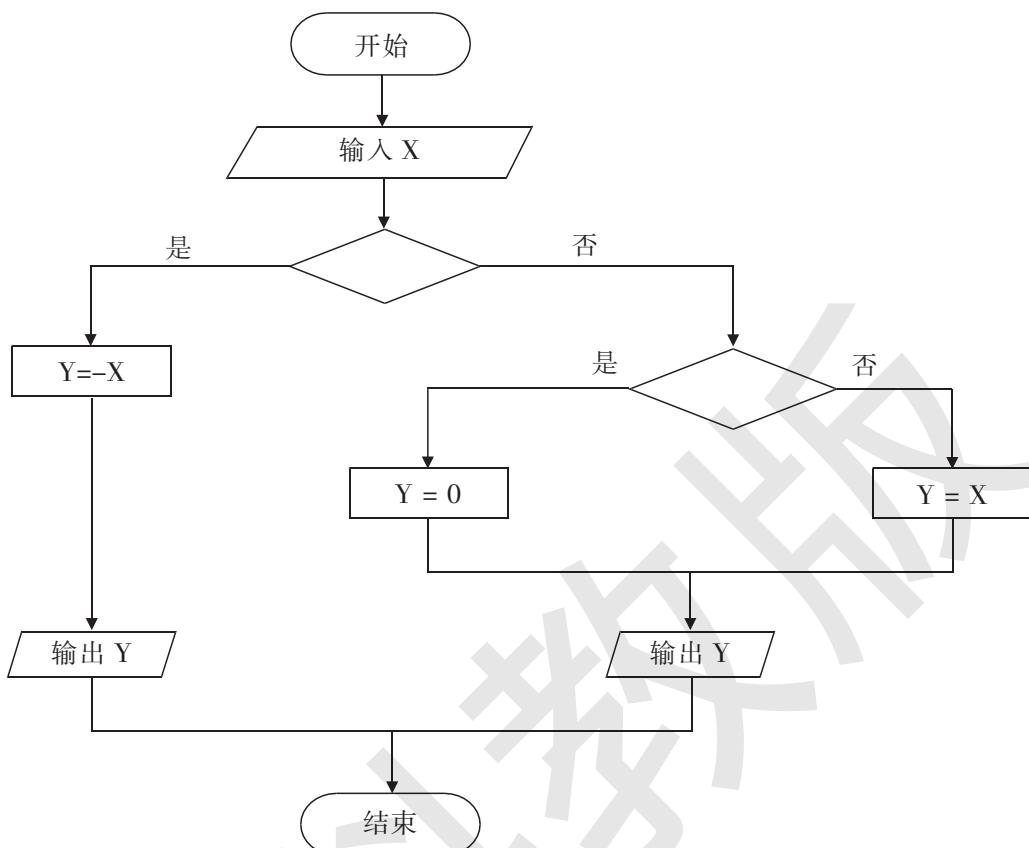
(1) 完成渡河问题算法的流程图。



(2) 完成个人理财问题算法的流程图。



(3) 完成上述活动二 1 中用自然语言描述的算法的流程图。



4. 了解用伪代码描述算法。

阅读用伪代码描述的个人理财问题算法。

提 示 板

伪代码是用介于自然语言和计算机语言之间的人工语言来描述算法的工具。

详见学习指引。

```

INPUT A,R,T
I = 0
P = A
DO WHILE I < T
  P = P × (1 + R × 80%)
  I = I + 1
LOOP
PRINT P
  
```

讨论

比较自然语言、流程图和伪代码三种算法描述方法,它们各有什么特点? 各有何优点和不足?

下页续

> 接上页

算法描述方式	优点	缺点
自然语言		
流程图		
伪代码		

活动三 了解算法的作用

问题:求两个正整数的最大公约数。

我国古代数学家和希腊古代数学家都研究了解决该问题的算法。以下即为他们各自的算法。

1. 了解“更相减损之术”及其算法。

(1) 算法分析。

我国古代数学家研究的算法称“更相减损之术”，其方法是以两数中较大的数减去较小的数，获得的差与原先较小的数构成新的一对数，再以大的数减去小的数……如此以往，用同样的方法操作，直至产生一对相等的数。

以求 16,12 这两个数的最大公约数为例。以两数中较大的数减去较小的数，即 $16-12=4$ ，以差数 4 和较小的数 12 构成新的一对数，对新的一对数再用大数减小数，以同样的方法一直做下去，直到产生一对相等的数，这个数就是最大公约数。具体操作如下：

$$(16, 12) \rightarrow (4, 12) \rightarrow (4, 8) \rightarrow (4, 4)$$

4 是 12 和 16 的最大公约数。

(2) 完成算法的自然语言描述。

求两个正整数的最大公约数算法(设给定的两个正整数为 m 和 n):

- ① 输入两个正整数 m 和 n;
- ② 若 $m < n$, 则交换 m 和 n;
- ③ 以 m 减去 n, 令所得的差为 r;
- ④ 若 $r = 0$, 则输出结果 , 算法结束; 否则, 继续步骤 ;
- ⑤ 若 $r < n$, 则令 $m = n, n = r$; 否则令 $m = r$, 并返回步骤 继续进行。

2. 了解“辗转相除法”及其算法。

(1) 算法分析。

古希腊数学家研究的求两个正整数最大公约数的方法

是辗转相除法(欧几里得算法)。具体方法是用较大的数除以较小的数,余数和较小的数构成新的一对数,继续上面的除法,直到大数被小数除尽,则较小的数就是最大公约数。

以求 288 和 123 的最大公约数为例,具体操作如下:

$$(288, 123) \rightarrow (42, 123) \rightarrow (42, 39) \rightarrow (3, 39)$$

3 是 288 和 123 的最大公约数。

(2) 完成算法的自然语言描述。

求两个正整数的最大公约数算法(设给定的两个正整数为 m 和 n):

- ① 输入两个正整数 m 和 n;
- ② 若 $m < n$, 则交换 m 和 n;
- ③ 以 m 除以 n, 令所得的余数为 r;
- ④ 若 $r = 0$, 则输出结果 ____, 算法结束; 否则, ____, 继续步骤 ____。

3. 运行根据两种算法编写的程序,比较效率。

(1) 运行“更相减损之术”程序,记录运行快慢。

① 启动 VB, 在弹出的对话框中单击“打开”按钮,在“视图”菜单中选择“代码窗口”,输入“更相减损之术”程序。

```
Private Sub Form_Click()
    Dim m As Long, n As Long, r As Long
    m = Val(InputBox("输入数字 1"))
    n = Val(InputBox("输入数字 2"))
    If m < n Then r = m: m = n: n = r
    r = m - n
    Do While r <> 0
        If r < n Then
            m = n
            n = r
        Else
            m = r
        End If
        r = m - n
    Loop
    Print "最大公约数为"; n
End Sub
```

② 在弹出的数据输入窗口中,输入两次数据(或自定数据),分别记录运行时间:

输入数据(例)	运行时间(秒)
第一次:356727,58263	
第二次:264062214,39	

(2) 运行“辗转相除法”程序,记录运行快慢。

① 启动 VB,在弹出的对话框中单击“打开”按钮,在“视图”菜单中选择“代码窗口”,输入“辗转相除法”程序。

```
Private Sub Form_Click()
    Dim m As Long, n As Long, r As Long
    m = Val(InputBox("输入数字 1"))
    n = Val(InputBox("输入数字 2"))
    r = m Mod n
    Do While r <> 0
        m = n
        n = r
        r = m Mod n
    Loop
    Print "最大公约数为"; n
End Sub
```

② 在弹出的数据输入窗口中,输入两次数据(或自定数据),分别记录运行时间:

输入数据(例)	运行时间(秒)
第一次:356727,58263	
第二次:264062214,39	

(3) 比较两个程序的运行效率。

当输入的数据为 264062214 和 39(或自定数据)时,填写下表:

比较项目\算法	更相减损之术	辗转相除法
结果		
程序运行时间	<input type="checkbox"/> 较快 <input type="checkbox"/> 较慢	<input type="checkbox"/> 较快 <input type="checkbox"/> 较慢
对算法的理解	<input type="checkbox"/> 容易理解 <input type="checkbox"/> 不易理解	<input type="checkbox"/> 容易理解 <input type="checkbox"/> 不易理解



讨论

在“求两个正整数的最大公约数”问题的解决中,除了以上两种算法外,是否还有其他算法?



学习指引

在数学课的学习中,已经了解到,计算机解决任何问题都要依赖于算法。只有将解决问题的过程分解为若干个明确的步骤,即算法,并用计算机能够接受的“语言”准确地描述出来,计算机才能够解决问题。

■ 算法的概念

算法是在有限步骤内求解某一问题所使用的一组定义明确的规则。通俗地说,算法是用计算机求解某一问题的方法,是可以被计算机机械地执行的一组动作,这些动作必须是有穷的,即在有限步内可以完成。

■ 算法的特征

一个需要用计算机解决问题的算法应该具有以下五个重要的特征:

(1) **有穷性**:一个算法的步骤定义是有限的,并且必须保证执行有限步之后结束。例如,前述的更相减损术中,由于 m, n 均为正整数,在执行了步骤③后, r 必小于 m ,若 $r \neq 0$,下次执行③时, m 的值已经减小,而正整数的递降序列最后必然要终止。因此,无论给定的 m 和 n 的原始值有多大,步骤③的执行都是有穷次的,而不会是无限的,永无结束的。

(2) **确定性**:算法的每一步骤必须有确切的定义,无二义性。例如,在更相减损术中,步骤③中明确规定“以 m 减去 n ”,而不能有类似的“以 m 减去 n 或 n 减去 m ”这样有两种不确定做法的规定。

(3) **输入**:一个算法有零个或多个输入,以刻画运算对象的初始情况。例如,更相减损术和辗转相除法中的输入都是两个: m 和 n 。所谓零个输入是指算法本身定义了初始条件。

(4) **输出**:一个算法有一个或多个输出,以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。例如,更相减损术中的输出只有一个,即步骤④中的 n 。

(5) **可行性**:算法中有待执行的运算和操作是相当基本的,能够精确地运行的。算法的执行者甚至不需要掌握算法的含义,只要根据算法的具体步骤要求,按部就班地进行操作,最终即可得到正确结果。

■ 算法的描述

算法是人求解问题的思路和方法,是解题过程的精确描述,需要用恰当的方式表述出来。算法的表述方法很多,其中主要的有自然语言、流程图和伪代码三种。

1. 自然语言

自然语言就是人们平时交流使用的语言,如汉语、英语、法语、德语等。用自然语言描述

算法,通俗易懂,且不需要专门训练如何使用语言。前面活动一中个人理财问题、找出最大数问题等都是用自然语言描述的算法。再如,“求三个数中值最大的数”的算法,用自然语言可描述如下:

- 第1步:输入三个正整数 a, b, c ;
- 第2步:如果 a 大于 b ,则将 a 的值赋给 max ,否则将 b 的值赋给 max ;
- 第3步:如果 c 大于 max ,则将 c 的值赋给 max ;
- 第4步:打印:“最大值为” max 。

但是用自然语言描述算法,也有其不足之处:

- (1) 文字语言表述不如图形表述直观,过程与结构不如图形清晰;
- (2) 自然语言表示的算法不便翻译成计算机容易理解的程序设计语言。

2. 流程图

在数学课中,已经了解到,流程图是描述算法的常用工具,它由如前所述的一组规定了功能的图形符号表示算法。例如,“求三个数中值最大的数”的算法,用流程图描述如下:

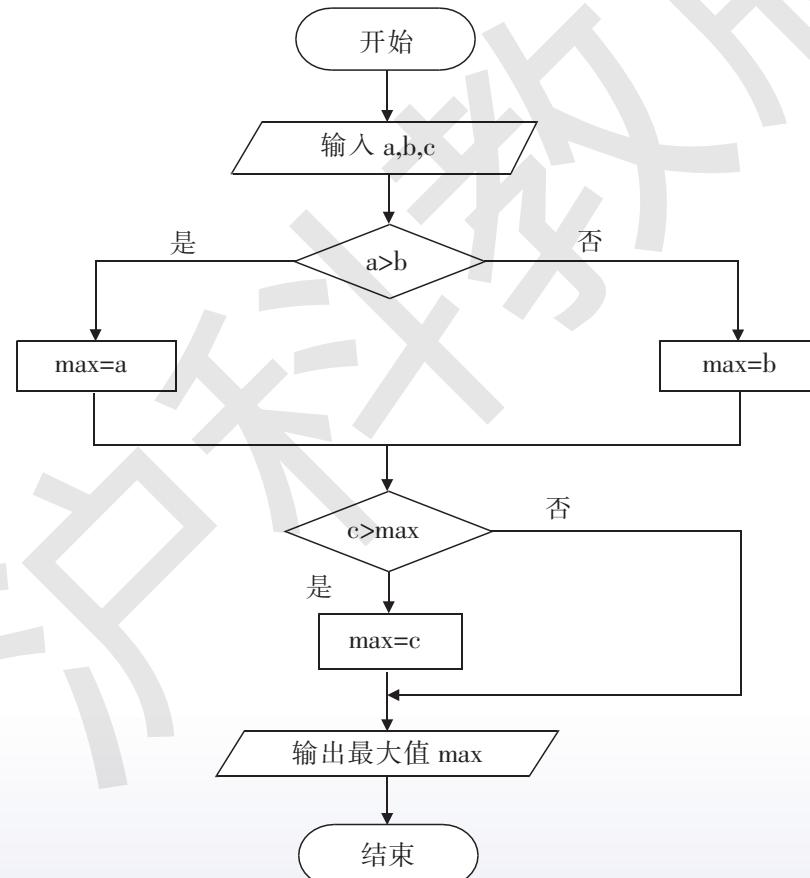


图 1-3 “求三个数中值最大的数”流程图

用流程图表示算法,流程的表达清晰简洁,算法结构的表达明确。流程图不依赖于任何计算机和计算机程序设计语言,有利于为任何环境下的程序设计服务。

3. 伪代码

伪代码是用介于自然语言和计算机程序设计语言之间的人工语言来描述算法的工具。

如,“求三个数中值最大的数”的算法,用伪代码描述如下:

```

input three integer numbers
print “请输入三个整数”
input a,b,c
maximum
if a > b then
    max=a
else
    max=b
if c > max then
    max=c
output the result
print “最大数是”:max

```

伪代码书写方便,容易理解。由于伪代码有接近于计算机程序设计语言的特点,因此用伪代码书写,格式紧凑、表达精练。伪代码不需要让计算机直接接受,又吸收了计算机程序设计语言简捷严谨的特点。用伪代码描述的算法易于向计算机程序设计语言过渡。

然而,正是因为自然语言和计算机程序设计语言种类繁多,因此伪代码的语句与关键词无法规范。

■ 算法在用计算机解决问题过程中的地位和作用

如前所述,同样是求两个数的最大公约数,我国古代数学家研究出了更相减损术,西方数学家研究出了欧几里得算法——辗转相除法,算法设计可谓各不相同。亦即解决同一个问题,算法不是唯一的,可以是多种的。

然而,在很多用计算机解决问题的场合,对于同一个问题,由于算法不同,计算机的运行效率会有很大不同。就更相减损术和辗转相除法来说,当输入的一对数据数值较小时,两个程序的运行效率相差无几。当输入的一对数据数值较大时,会发现程序的运行效率有明显的差异。再以求自然数 1~100 之和为例:

算法一:依次将自然数 1,2,3,……,100 累加,做 99 次加法,得到最后结果。

算法二:从首尾对称地取两数相加,(1+100),(2+99),(3+98),……,(50+51),如此可得到 50 个 101。最后可得到结果为(50×101)。

上述两个算法中,算法一比较简单、清晰,一目了然,但要重复多次加法运算。算法二运算次数较少,但需经历简便运算的分析,可理解性较算法一差。就本例而言,由于问题很简单,利用计算机运算速度快的特点,直接用算法一并无大碍。但在通常情况下,需用计算机解决的问题相对较复杂,因此对算法的设计就提出了一定的要求。

对于软件工程人员来说,一个“好”的算法,除正确性外,还应考虑:

高效性,主要指运行速度快和占用的存储空间小;

可读性,有助于人对算法的理解,以方便程序的调试和今后的维护;

健壮性,当输入数据非法时,算法也能作出适当反应,进行必要的处理,不至于引发系统崩溃。

由此可见,在应用计算机解决问题的过程中,算法设计有着举足轻重的地位和作用。算法是计算机解决问题中的核心和灵魂。算法的好坏将直接影响着程序的通用性和有效性,影响着解决问题的效率。



体验活动

1. 用自然语言写出自己在家里烧开水的过程(或你每天起床)的算法。
2. 分别标有1、2、3、4、5、6六个号码的小球,有一个最重。请设计一个挑出其中最重球的算法,用自然语言把你的算法描述出来。
3. 试用流程图将一位班长组织全班同学进行一次春游活动的过程描述出来。
4. 用流程图描述活动三中“求两个正整数的最大公约数”的两个算法。

第三节 程序和程序设计语言

色香味俱全的美味佳肴是厨师根据菜谱精心烹调而成;涤荡心弦的音乐是演员按照乐谱成功地演奏而成;风格迥异的建筑是建筑工人按照设计师绘就的蓝图精心施工而成;功能各异的软件是程序员依照精妙设计的算法编制而成。计算机神机妙算的高超本领是由于它忠实地执行了程序。

如同按照菜谱烹饪要有烹饪工具,按照乐谱演奏需要乐器,按照蓝图施工需要建筑工具和机器,编制程序需要语言工具,计算机执行程序同样需要识别和执行程序的工具。

那么,什么是程序?程序又是如何被计算机执行的?

活动一 认识程序

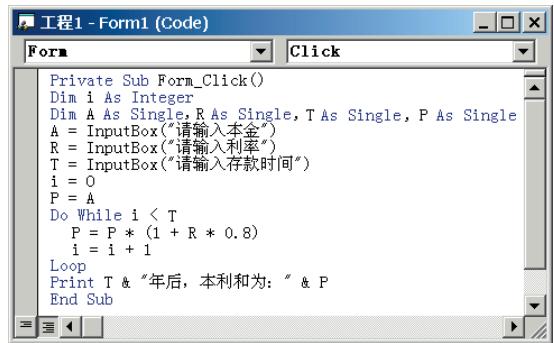
1. 浏览源程序。

(1) 运行配套光盘“个人理财问题”的可执行程序“个人理财.exe”。在本金 $A = 20$ 万,银行一年定期储蓄年利率 $R = 2.25\%$,存款时间 $T = 3$ 年时,扣除利息税后的本利和 $P = \underline{\hspace{2cm}}$ 。

(2) 打开“个人理财问题”的源文件“个人理财.vbp”,在“代码窗口”可浏览源程序如下:

提 示 板

计算机之所以在接受了三个数据后能计算出三年后的税后本利和,是因为计算机执行了个人理财问题的这段程序。



```

Private Sub Form_Click()
    Dim i As Integer
    Dim A As Single, R As Single, T As Single, P As Single
    A = InputBox("请输入本金")
    R = InputBox("请输入利率")
    T = InputBox("请输入存款时间")
    i = 0
    P = A
    Do While i < T
        P = P * (1 + R * 0.8)
        i = i + 1
    Loop
    Print T & "年后, 本利和为: " & P
End Sub

```

图 1-4 “个人理财问题”源程序

2. 探寻算法与程序间的对应关系。

(1) 分析比较算法与程序的对应关系。

个人理财问题算法：

个人理财问题程序：

提 示 板

个人理财问题程序是用计算机能接受的 VB 语言对解决该问题的算法的描述。对于计算机而言,实际上是一组解决该问题的操作的指令(或语句)序列。

程序的概念及其具有的特征,详见学习指引。

① 输入 A、R、T 的值;

② 令 $i=0$;
③ 令 $P=A$;④ 如果 $i < T$, 则执行⑤,
否则执行⑦;
⑤ $P=P \times (1+R \times 80\%)$;
⑥ $i=i+1$, 转④执行;⑦ 输出结果 P ;

⑧ 结束。

```

A=InputBox("请输入本金")
R=InputBox("请输入年利率")
T=InputBox("请输入存款时间")

```

(2) 试分析“个人理财”问题程序中每条指令(或语句)的语义。



讨论

选择其他实例,分析比较算法与程序的对应关系。

活动二 了解程序的结构

1. 认识顺序结构。

(1) 打开“求点到直线的距离”的源文件“求点到直线的距离.vbp”, 观察该程序有几条语句。

(2) 按 **F8** 键单步运行程序, 观察程序执行的顺序, 记录观察的结果。

```

Private Sub Form_Click()
    Dim A As Integer, B As Integer, C As Integer
    Dim X As Integer, Y As Integer
    Dim S As Single
    A = Val(InputBox("请输入直线AX+BY+C=0的A的值"))
    B = Val(InputBox("请输入直线AX+BY+C=0的B的值"))
    C = Val(InputBox("请输入直线AX+BY+C=0的C的值"))
    X = Val(InputBox("请输入点的横坐标"))
    Y = Val(InputBox("请输入点的纵坐标"))
    S = Abs(A * X + B * Y + C) / Sqr(A * A + B * B)
    Print "点到直线的距离为"; S
End Sub

```

提 示 板

程序有三种基本的结构：
顺序结构、选择结构和循环结构。一个基本结构可以包含一条或若干条语句。

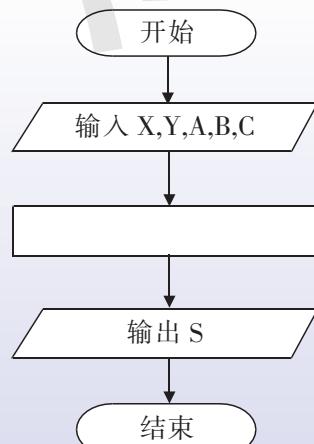
图 1-5 “求点到直线的距离”程序的执行顺序

程序有 _____ 条语句。

程序执行的顺序是：

_____;
_____;
_____;
.....
_____。

(3) 用流程图描述如下。



提 示 板

各语句依次按顺序执行的结构称为**顺序结构**。

2. 认识选择结构。

(1) 打开“求绝对值问题”的源文件“求绝对值.vbp”，观察该程序有几条语句。

(2) 按 **F8** 键单步运行程序，观察程序执行的顺序，记录观察的结果。

图 1-6 “求绝对值问题”的程序执行顺序

提 示 板

通过判断条件是否成立来选择执行程序的结构称为**选择结构**。

程序有_____条语句。

程序执行的顺序是：

_____；
_____；
_____；
.....
_____。

其流程图参见第 14 页流程图。

3. 认识循环结构。

(1) 打开“个人理财问题”的源文件“个人理财.vbp”。

(2) 按 **F8** 键单步运行程序，观察程序执行的顺序，记录观察的结果。

提 示 板

按照一定条件，反复执行程序中某一部分代码的结构称为**循环结构**。

关于程序的基本结构详见**学习指引**。

程序执行的顺序是：

_____；
_____；
_____；
.....
_____。

其流程图参见第 13 页流程图。



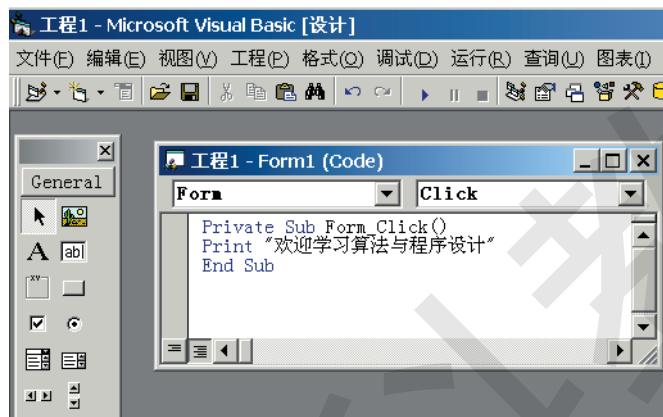
讨论

解决问题的算法和程序千差万别,但都是由顺序结构、选择结构和循环结构这三种基本结构构成。请举例说明你的认识。

活动三 了解程序的生成和执行

1. 了解程序的生成。

(1) 启动 Visual Basic 程序设计环境,在代码窗口中输入“Print”欢迎学习算法与程序设计”,如图 1-7 所示。



在代码窗口输入程序并加以保存等工作过程,称为**程序的编辑**。详见学习指引。

图 1-7 输入程序代码

(2) 单击“保存”按钮,生成一个 VB 程序。

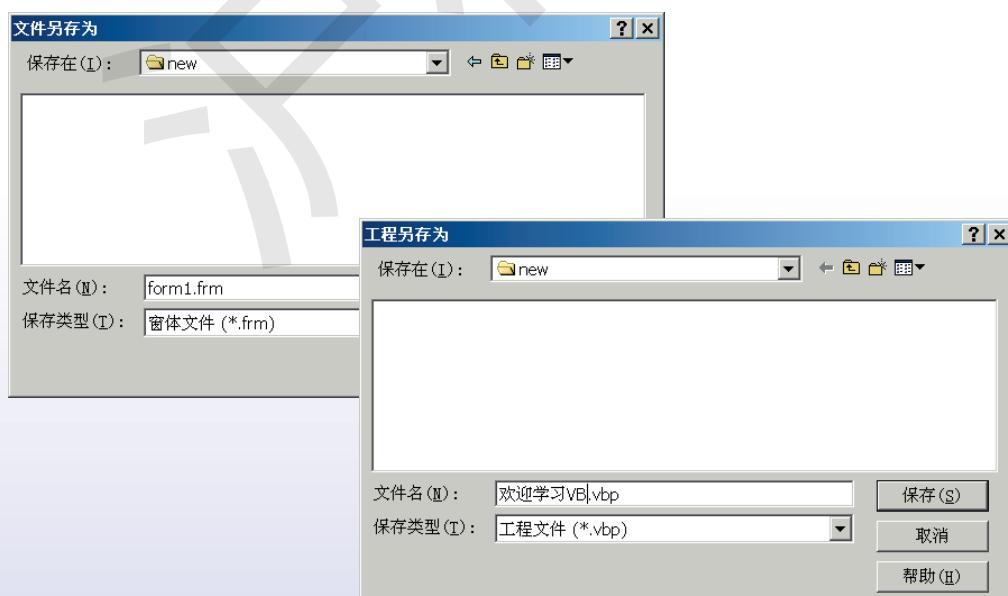


图 1-8 保存程序

提 示 板

计算机只能识别和执行由0和1组成的二进制的机器语言。对于高级语言源程序,计算机必须通过翻译程序或解释程序,将其转化为计算机能识别的机器语言,才能执行。详见学习指引。

2. 了解程序执行。

(1) 单击启动按钮▶,可以看到程序运行的结果。



图 1-9 运行结果

(2) 经编译生成可执行文件。



图 1-10 编译生成可执行文件

(3) 在资源管理器中找到并双击生成的文件“生成欢迎学习 VB.exe”。



在前两节的许多活动中,我们可以体会到算法设计对于问题的解决有着极其重要的作用。然而,要使计算机确实有解决问题的本领,还需依赖与算法相对应的程序。当然,程序的编制依赖于算法的设计,程序的效率主要取决于算法的效率。

■ 程序及其特征

如前所述,程序(program)实际上是一组操作的指令或语句的序列,是计算机执行算法

的操作步骤。计算机是通过运行该组指令,完成预期的任务的。

从对“个人理财问题”及其他实例的算法及与相应程序对应关系的分析中,可以感受到用VB语言编写的程序是如何通过操作来实现该问题解决的算法的。一般来说,程序具有以下一些特征:

- (1) 解决问题的每一个算法的步骤对应着程序中的一个或多个语句,每个语句实现一个或多个操作,每一个操作都有执行对象;
- (2) 在程序中被施行操作的对象称为数据。数据有属性和取值范围,即类型说明(见第2章);
- (3) 除了有特殊的程序转向控制语句之外,程序都是从第一条语句开始顺序执行。有时语句要求执行时作出判断,在某种情况下执行一条或一组语句,否则执行另一条或另一组语句;有时一组或一条语句可能需要执行一次以上,此时必须给出重复的次数或条件。

■ 程序的基本结构

在数学课中,我们已了解程序有三种最基本的结构:顺序结构、选择结构和循环结构。程序是由若干个基本结构组成的。而一个基本结构可以包含一条语句或若干条语句。

前面活动二中“求点到直线的距离”的程序,有10条语句。各语句是依次按顺序执行的,这就是一个**顺序结构**(sequence structure)。

“求绝对值问题”的程序中,语句“Print”该数的绝对值是”;Y”,要通过检验 $X < 0$ 成立与否决定是否输出 $Y = -X$;及通过检验 $X = 0$ 成立与否或非前两种情况下,决定是输出 $Y = 0$ 还是输出 $Y = X$,这就是一个**选择结构**(selection structure)。

在“个人理财问题”程序中,有三个语句“Do While $i < T$ ”、“ $P = P * (1 + R \times 0.8)$ ”和“ $i = i + 1$ ”需要重复执行,直到 $i = 3$ 为止。这称为**循环结构**(loop structure)。

任何程序都可用顺序、选择、循环三种基本逻辑结构,经过不同组合和反复嵌套予以实现。因此。这三种基本结构也称为程序的三种基本控制结构。事实上任何一种计算机程序设计语言都离不开这些控制结构,甚至可毫不夸张地说,能否真正掌握好一门计算机语言,归根到底是能否灵活运用这些控制结构。

程序的三种基本控制结构的一般形式如图1-11所示:

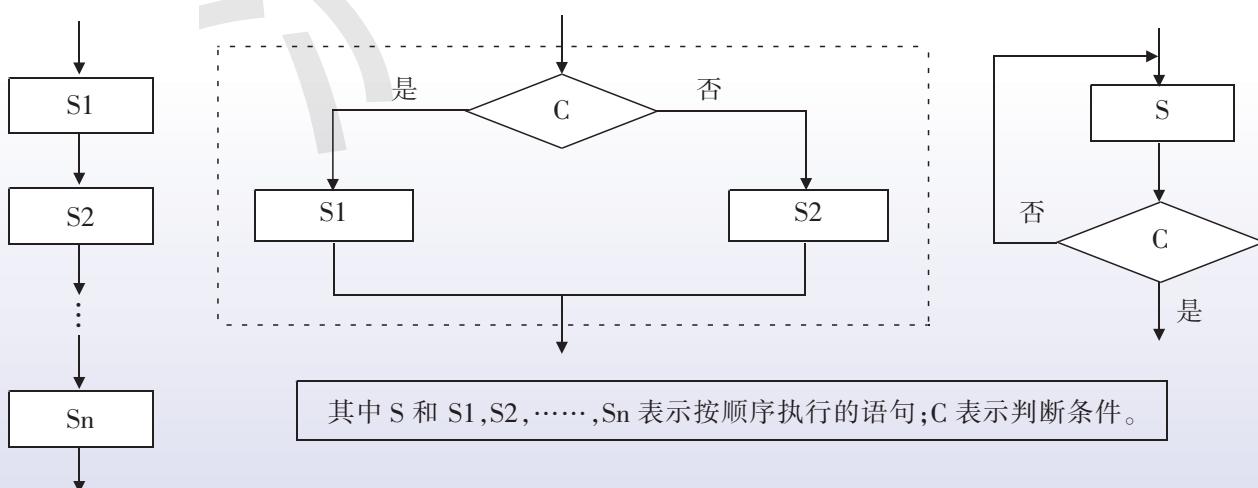


图1-11 程序的三种基本结构

从图示可以看出,三种基本控制结构都是一个入口,一个出口。用它们的组合来表现程序,结构清晰,容易理解,可读性强。

■ 程序设计

计算机不会自动地解决问题。计算机的所有操作都是按照人们预先编制好的程序进行的。因此若需运用计算机帮助解决问题,必须把具体问题转化为计算机可以执行的程序。而在问题提出之后,从分析问题、建立模型、设计算法、编写修改程序,一直到程序正确运行的整个过程就称作**程序设计**(programming)。



图 1-12 程序设计

一个程序质量的高低影响着解决问题的效率。一个高质量的程序除了依赖于第二节所述的一个“好”的算法的设计外,还需具备:

- (1) 程序具有良好的结构,逻辑清晰、可读性强;
- (2) 运行结果正确,满足问题解决的精度等要求;
- (3) 程序运行时间合适;
- (4) 便于检查、修正、移植和维护。

■ 程序设计语言的产生与发展

程序设计是为了达到某一计算和数据处理的目的,设计实现算法的命令和组织相关数据的过程。这种设计、组织和安排必须能被计算机理解并执行,这就需要有计算机的程序设计语言。

程序设计语言(programming language)是指人们编制程序所使用的计算机语言。随着计算机的发展,程序设计语言经历了从机器语言到高级语言的发展历程。

1. 机器语言

计算机主要由电子元器件组成的电路构成。由于电子元器件的特性,使得计算机只能识别“0”和“1”组成的机器代码。早期的程序设计语言就是由“0”和“1”这样的二进制代码指令组表示的,称为**机器语言**(machine language)。通常,不同的计算机,其指令系统会有所不同。每一条机器指令一般包含两个主要部分:操作码(规定了指令的功能)和操作数(规定了被操作的对象)。有的指令没有操作数,如停止指令。在这些指令的控制下,计算机可以实现最基本的算术运算和逻



图 1-13 程序设计语言的发展

辑运算。以 $5+12$ 运算为例,若某型号计算机指令采用 16 位二进制表示,用机器语言编写一个完成 $5+12$ 运算的程序可见下表:

指令序号	机器指令	指令说明	指令功能
1	1011 0000 0000 0101	操作码:表示数据传输功能。 左操作数(4位)表示0号寄存器, 右操作数(4位)是5。	把加数5送到0号寄存器中。
2	0000 0100 0000 1100	操作码:表示加法功能。 左操作数表示0号寄存器,右操作 数是12。	把0号寄存器中的内容与另一 数相加,结果存在0号寄存器中 (即完成 $5+12$ 的运算)。
3	1111 0100	停止功能	停止操作

用机器语言编写程序,有着诸多的不便,例如,必须时时注意到机器硬件的性质,如上述程序中要关心使用哪个寄存器;二进制数表示的语句冗长费解,不易使用推广;不相兼容的计算机其机器语言不同,用机器语言编写的程序在不同型号的计算机上不能通用。

当然,由于指令是计算机的基本操作,因此用机器语言编写的程序占用的存储单元较少,结构紧凑,执行速度快。

2. 汇编语言

用机器语言编写程序,对大多数使用者来说是一种灾难。20世纪50年代中期,相关人员开始采用一种类似英语缩略词并带有助忆符号的语言,以此替代复杂的二进制代码指令和操作数来编写程序,这就是**汇编语言**(assembly language)。用汇编语言编写的程序,计算机不能直接接受,需用汇编语言的翻译程序将程序中的每条语句翻译成机器语言,计算机才能执行。仍以上述完成 $5+12$ 运算的程序为例,改用汇编语言编写,如下表所示:

指令序号	汇编语言指令	指令功能
1	MOV AL,5	把加数5送到累加器AL中。
2	ADD AL,12	把累加器AL中的内容与另一数相加,结果存在累加器AL中 (即完成 $5+12$ 的运算)。
3	HLT	停止操作

用汇编语言编写程序比机器语言显然要容易。但由于汇编语言与机器语言之间大多存在一一对应关系,因此除了用符号替换二进制数的好处外,汇编语言保留了机器语言的许多特点。

3. 高级语言

20世纪60年代中期,接近于人的自然语言的**高级语言**(high-level language)问世。用高级语言编写程序简洁易懂。如,仍以上述程序为例,用VB语言编写即为简单易懂的语句:

Print 5+12

用高级语言编写的程序同样必须由翻译程序将其翻译成机器语言,才能在计算机上执

行。高级语言与机器语言之间是一对多的关系,即一条高级语言语句对应多条机器语言语句。

高级语言独立于机器,用其编写程序无须顾及与机器相关的实现细节,其可移植性比汇编语言要好得多。

高级语言的诞生是计算机技术发展的一个重要里程碑。它的出现为计算机的应用开辟了广阔的前景。

■ 程序的编辑和翻译

1. 程序的编辑

用汇编语言或高级语言编写的程序称为“源代码”。这些代码需要逐一地输入计算机中,并把它们以文件的形式保存起来,这一过程称为**程序的编辑**。被保存的文件称为源文件。本章活动中,多次提到的后缀名为.vbp 的源文件都是以文件形式保存起来的各问题求解程序。源文件一般都以文本形式存储,和我们常见的.txt 文件没有区别,可以用记事本等简单的文本编辑器对它们进行创建或编辑。很多高级语言有自带的编辑器,提供了较强的对源程序的修改、追加、删除等编辑功能。本节活动三就是用 VB 自带的编辑器进行创建和编辑工作的。(本章活动中后缀名.vbp 的文件是在 VB 中创建应用程序必须要有的工程文件。详见第 2 章)。

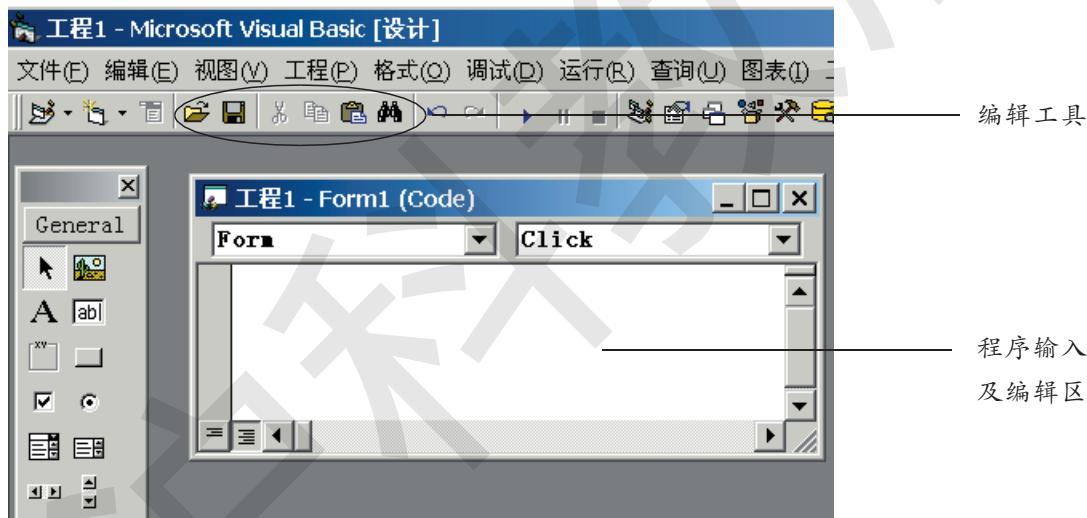


图 1-14 程序编辑区与编辑工具

2. 程序的翻译

从前面的学习中已经了解,人们乐于使用汇编语言和高级语言来编写程序,但计算机不可能直接执行这样的源程序。计算机只能识别和执行二进制机器语言代码。因此,所有源程序必须翻译成机器语言程序,才能为计算机识别并执行。

如前所述,汇编语言编写的程序,须用一种语言翻译器将程序中的每条语句翻译成机器语言,然后被保存在一个由它生成的程序文件中,如此便得到了计算机所需要的可执行程序。

高级语言的翻译程序有两种类型:编译程序和解释程序。

(1) 编译程序。

编译程序(compiler)的工作是将高级语言编写的源程序在执行前将其翻译成等效的机器语言程序,继而再连接生成可执行文件。其编译过程如图 1-15 所示。

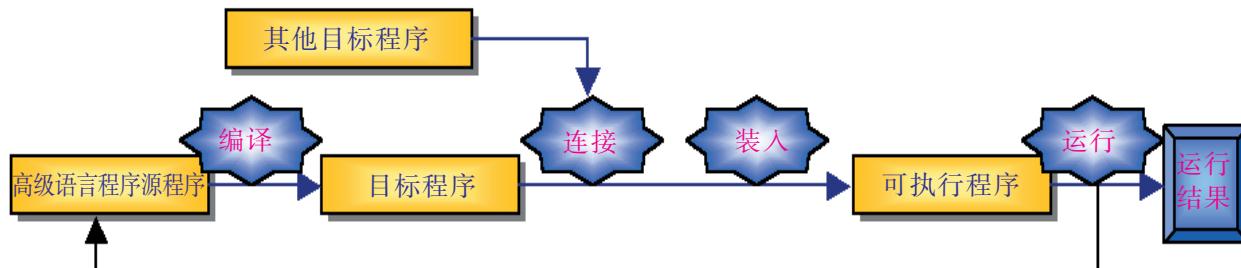


图 1-15 编译过程

- 其中：
- ① **编译**(compile)：将源程序编译成等效的汇编代码，然后由汇编程序将汇编代码翻译成目标程序(或称目标代码，文件后缀名为.obj)。目标程序一般不能直接在计算机上执行。
 - ② **连接**(link)：将若干目标代码连接在一起，构成一个完整的目标代码程序。
 - ③ **装入**(load)：将完整的目标代码程序装入主存储器，成为可在机器上直接执行的机器语言程序。

经编译连接生成后的可执行程序，可脱离高级语言环境独立执行，但是一旦程序需要修改，必须返回到编辑状态，经修改后，再重新编译连接，最终生成可执行程序继而执行。

(2) 解释程序。

解释程序(interpreter)的执行方式如同日常生活中的“同声翻译”。源程序由解释程序逐句转换为可解释执行的目标代码，然后执行。其转换方式是对源程序的每一可能的行为都用机器语言编写一个子程序，给予模拟。因此，对高级语言程序的解释是调用一系列子程序来完成的。解释过程如图 1-16 所示。由于是逐句重复图示过程，因此执行效率较低。但程序修改方便，可以动态地修改。

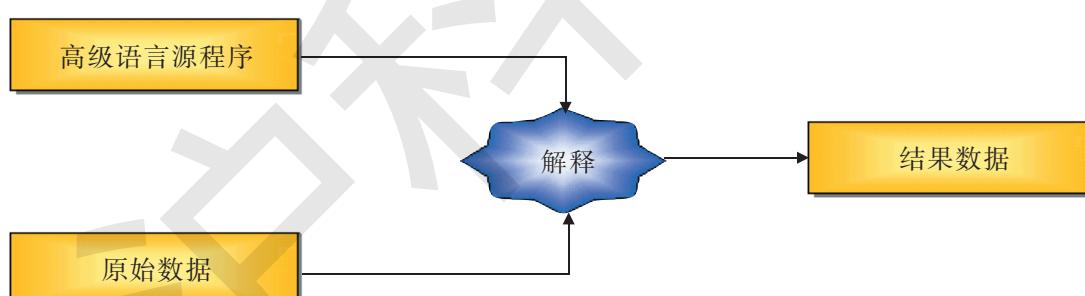


图 1-16 解释过程

现在，大多数程序设计语言都是编译型的，如 C/C++、Pascal、Fortran、JAVA 等，而早期的 Basic 语言却是解释型的。



体验活动

1. 讨论与交流：分析编译型语言和解释型语言的特点，比较各自的优势与不足。

下页续

> 接上页

语言种类	优 势	不 足
编译型语言		
解释型语言		

2. 打开配套光盘中的“求平均数.vbp”，按 **F8** 键单步运行程序，观察程序的执行顺序，分析程序的基本结构。

3. 用高级语言编写的程序，是怎样被计算机识别并执行的？试描述这一过程。



后续活动

1. 了解一种程序设计语言的发展。

选择一种程序设计语言，通过上网查询或图书馆查阅资料，了解这种程序设计语言产生、发展的历史与过程，能够解释其意义，试回答下列问题：

- (1) 这种语言是由谁创建的？
- (2) 这种语言有怎样的原创建背景？
- (3) 这种语言开发的主要应用场合是什么？
- (4) 这种语言是编译执行的还是解释执行的？
- (5) 这种语言的主要特点有哪些？

2. 收集有关如何学习程序设计的经验和体会。

到相关网站的讨论区和初学者及有经验的编程高手交流，找找从编程入门到自己开发简单程序的“捷径”，可以一起探讨以下问题：

- (1) 作为新手，如何克服畏难情绪，快速入门？
- (2) 同为初学者，在学习程序时，别人是怎么做的？有什么高招吗？
- (3) 初学程序设计，可能会走哪些弯路？怎样提高学习效率？
- (4) 有没有适合初学者的学习资源推荐？哪里可以看到简单、有趣、实用的编程例子？
- (5) 要成为一名好的程序员，要具备哪些品质？



评价

一级指标	评价内容	评价(★★★、★★、★)		
		自评	互评	教师评
信息技术知识	了解什么是算法,知道自然语言、流程图、伪代码三种算法的描述方法。			
	了解什么是程序及程序设计。			
	了解程序与算法的关系。			
	了解程序的三个基本结构。			
信息技术能力	了解人使用计算机解决问题需经历分析问题、建立模型、设计算法、编写程序、调试程序、运行程序的过程,以及过程中各环节的主要工作。			
	学会分析问题,根据已知条件建立模型,设计算法。			
	能够使用自然语言、流程图描述算法。			
行为	能运行已有程序,并验证结果是否正确。			
	能积极参与小组讨论,积极发表自己的意见并听取同学的意见,合作完成问题分析、建立模型、设计算法等环节工作。			
	在阅读算法和程序的过程中,主动探询算法与程序的关系,关注算法和程序在解决问题中的重要地位。			
体会:	在了解算法的同时,能关注古人解决问题时所用方法的精妙。			
	体会:			



参考资料

程序设计语言的发展

人与人之间的相互沟通和交往需要通信,通信首先需要的是语言。计算机出现后,人与计算机之间需要交互,其首要问题同样是语言问题。程序设计语言就是人与计算机之间的通信语言。

计算机的中央处理器是在二进制的指令控制下进行运算和执行操作的。这种二进制指令称为机器语言。计算机能直接理解的也就是这种语言。人类使用的是汉语、英语等各种自然语言,计算机理解不了。然而,要使计算机能按照人的意志去操作,又必须让计算机能理解人的意图。为此早期的计算机程序设计工

程序员不得不用二进制码去编写程序,然后用穿孔纸带或穿孔卡片把二进制指令码与数据送到计算机的存储器中,让计算机去执行。这是一种让人使用计算机的语言,向计算机发出指令的工作,实在是一件十分繁琐而且低效的工作。因此程序设计语言的研究是计算机科学中的一个重要课题。

人们希望计算机程序设计语言能尽量接近人类使用的自然语言和数学表达方式,这样不仅方便编写程序,也便于人们理解。最早出现的摆脱二进制编码的程序设计语言是汇编语言。汇编语言是一种把机器指令代码化的语言,是用英文字母来描述机器指令的语言。汇编语言命令与机器指令基本是一一对应的关系。由于可以用字母和数字来替代二进制码,程序的编写也就方便了许多。程序的可读性也比二进制码强。至今,汇编语言仍在操作系统核心部分和小型的控制设备中被使用。

随着计算机技术的发展,各种各样的计算机语言也相继出现。1952年出现了第一个高级语言 Short Code。它比较接近英语和数学的表达方式,用它编写程序,效率提高了许多,使用也更加方便。至今,不同类型、不同版本的程序设计语言先后出现不下数百种,它们有的昙花一现,有的被吸收合并到其他语言中,也有一些语言不断地被完善、更新、发展,形成了适应不同需要的常用程序设计语言,有的还被整合成一种软件的开发环境。

第2章

程序设计基础

通

过前面的活动,我们一起经历了分析问题、确定算法、编写程序等用计算机解决问题的基本过程,领略到程序设计在其中的重要地位。现在,我们将开始学习使用VB程序设计语言编写程序。在运用计算机来解决一些现实生活中的问题的过程中,逐步熟悉VB程序设计语言,学会如何实现程序的基本控制结构,掌握程序设计的基本思想和方法。

学习目标

- ★ 熟悉VB开发环境,理解并掌握数据类型、变量、常量、表达式、数组等概念,并能在程序编码中加以运用。
- ★ 学会运用赋值语句、输入语句和输出语句。
- ★ 学会运用选择语句实现各种选择流程控制。
- ★ 学会构建循环结构,并运用循环语句实现循环流程控制。
- ★ 了解知道并学习使用结构化程序的三种基本结构和自顶向下、逐步求精的方法。
- ★ 了解模块化程序设计的思想,并学会运用过程和函数实现模块。

第一节 VB 的可视化开发环境

启动VB,进入的是一个与Windows软件的风格几乎一致的图形用户界面,该用户界面上有工具栏和很多窗口。在这个功能强大的图形用户界面中,程序设计者能方便地完成应用程序的创建、运行、调试等工作,并通过类似绘制图形和安排图形的方式迅速设计用户想见到的窗口,不必再为界面的设计编写大量代码。

下面让我们一起在VB的可视化开发环境下创建一个简单的VB程序。



图 2-1 程序界面

活动一 创建一个新的应用程序

1. 创建一个工程。

(1) 启动VB,在出现的“新建工程”对话框中,单击“打开”按钮,建立一个标准EXE工程。

提 示 板

在VB的开发环境中,应用程序是以工程为单位组织的。一个**工程**(project)就是建立一个应用程序所需的所有文件的集合。

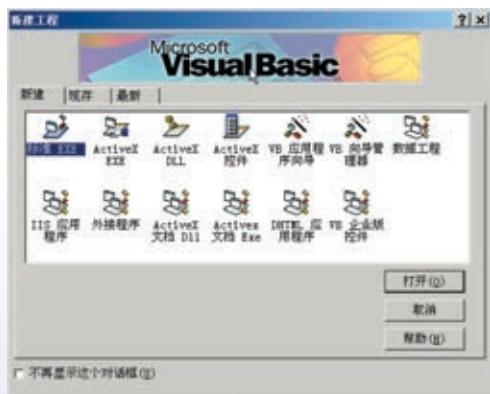


图 2-2 “新建工程”对话框

(2) 根据窗口标题的提示,在图2-3中找出“工程资源管理窗口”、“属性窗口”、“窗体布局窗口”、“窗体设计窗口”和“工具箱”。

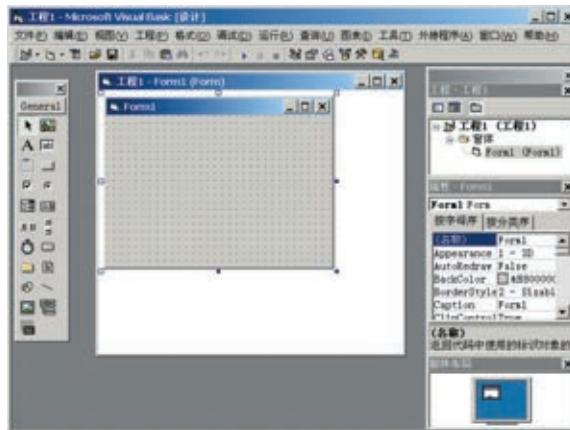


图 2-3 VB 主窗口

提 示 板

创建一个应用程序时,VB会在窗体设计窗口中打开一个名为 Form1 的窗体。窗体是一种预定义的结构化窗口,通过修改其属性可以建立各种不同的图形用户界面。

2. 认识 VB 的工作模式。

(1) 单击工具栏上的启动按钮  或者执行“运行”菜单中的“启动”命令,观察 VB 主窗口。

标题栏的标题是:_____。

(2) 单击工具栏上的结束按钮  或者执行“运行”菜单中的“结束”命令,观察 VB 主窗口。

标题栏的标题是:_____。

(3) 再次运行程序后,单击工具栏上的中断按钮  或者执行“运行”菜单中的“中断”命令,观察 VB 主窗口。

标题栏的标题是:_____。

活动二 了解属性窗口的使用

1. 选中窗体,在属性窗口的 Caption 属性中将窗体标题的显示文本设置为“Welcome”。

2. 分别在 Height 属性和 Width 属性中调整窗体的高度和宽度。

3. 在 StartUpPosition 属性中设置窗体的位置。

提 示 板

在 VB 中,每一个窗体都是一个对象。每一个对象都有相应的各种属性(property),用来描述它们的特征。这些属性值可以在属性窗口中修改。属性窗口的使用及对象的概念详见学习指引。

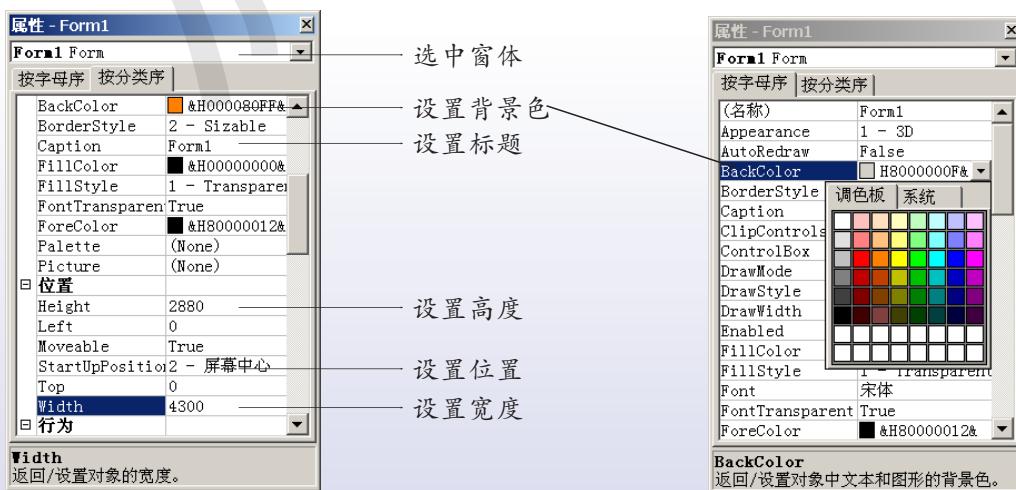


图 2-4 属性设置

4. 在 BackColor 属性中设置窗体的背景色。

活动三 了解标签控件的使用

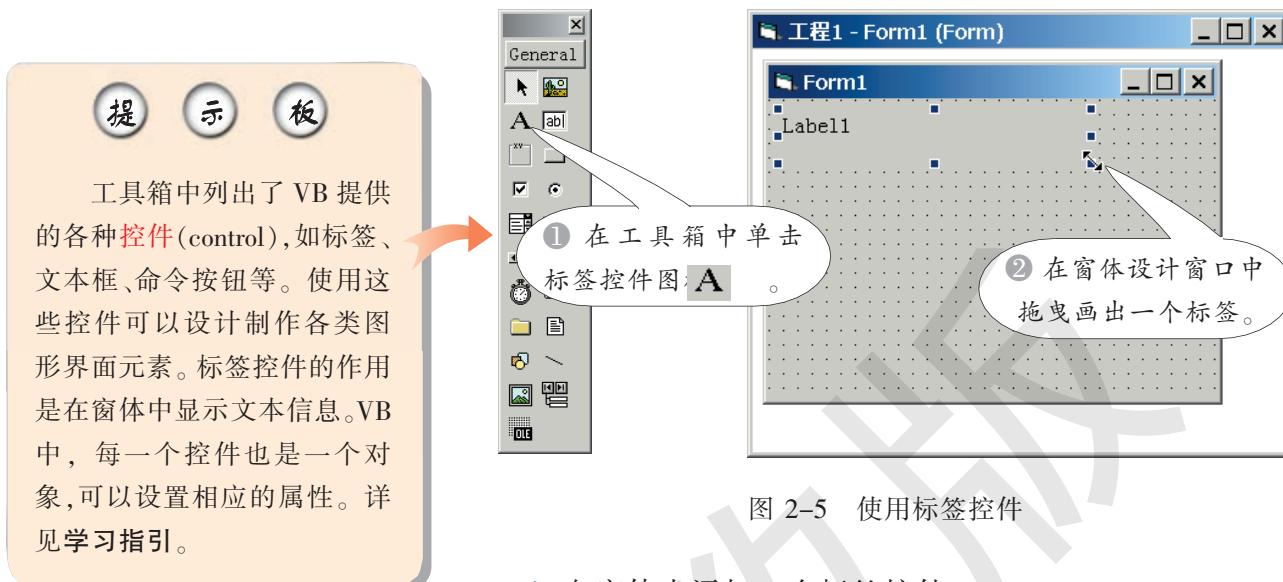


图 2-5 使用标签控件

1. 向窗体中添加一个标签控件。
2. 在窗体中单击该标签, 在属性窗口的 Caption 属性中将标签 Label1 的显示文本设置为“欢迎学习算法和程序设计”。
3. 在 Font 属性中设置标签显示文本的字体和大小。

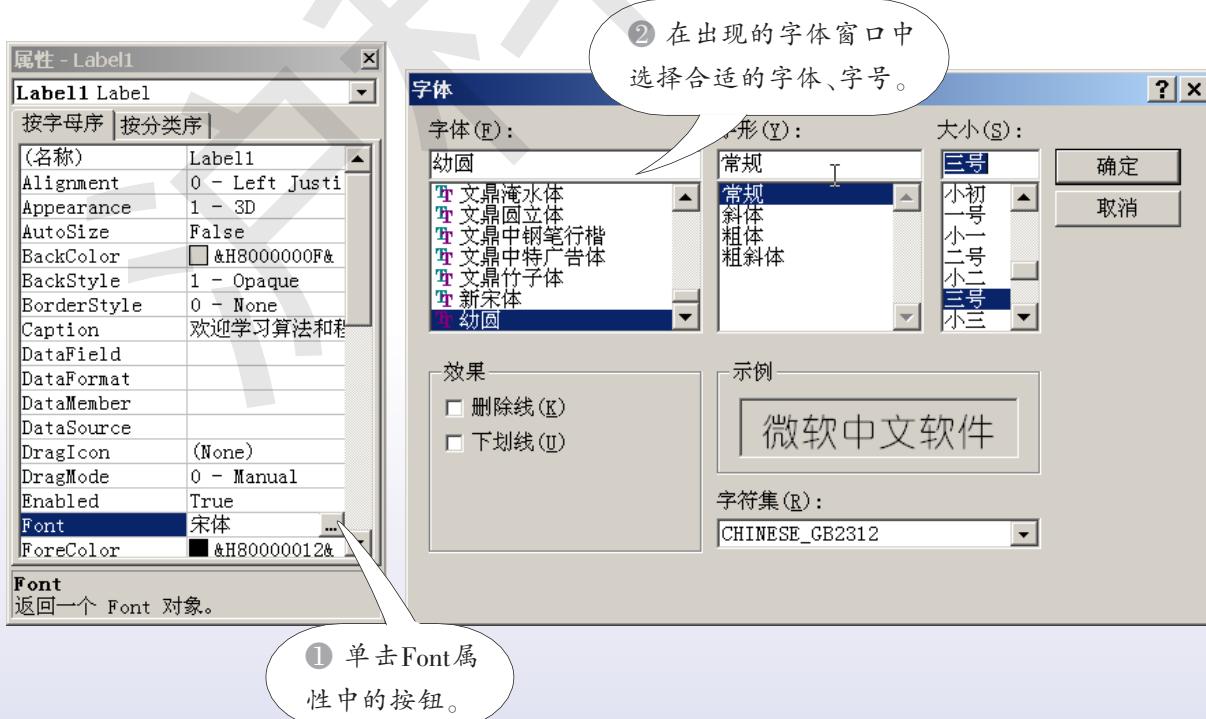


图 2-6 设置字体、字号

4. 在 BackStyle 属性的下拉框中选择“0-Transparent”，设置控件的背景样式为透明。

活动四 了解命令按钮控件的使用

1. 依照添加标签控件的方法在窗体中添加一个命令按钮控件。
2. 在属性窗口的 Caption 属性中设置按钮的显示文本为“退出”。
3. 双击命令按钮，在出现的代码窗口中添加一条实现程序退出功能的语句。

提 示 板

标签的颜色包括背景色和前景色。默认的背景色是白色，由 BackColor 属性设置；前景色表示字体的颜色，由 ForeColor 属性设置。如要让标签的背景色与窗体背景色一样，可以将 BackStyle 的属性值设置为“Transparent”（透明）。



图 2-7 输入语句

提 示 板

End 语句的作用是退出程序。程序运行时，一旦用户单击 Command1 按钮，就会执行 End 语句，这称为事件驱动。事件与事件驱动的概念详见学习指引。

4. 运行程序，并观察运行结果。

应用窗口的标题是：_____。

标签的背景颜色是：_____。

单击“退出”按钮时，出现的情况是：_____。

5. 保存工程。

- (1) 执行“文件”菜单下的“保存工程”命令，在弹出的“文件另存为”窗口中选择一个文件夹，将窗体文件以文件名 Welcome frm 进行保存。

提 示 板

保存工程需要保存一组文件，如窗体文件，用于保存设计制作好的窗体；工程文件，用于保存程序代码等。



图 2-8 保存窗体文件

(2) 在随后弹出的“工程另存为”窗口中将工程文件以文件名 FrmWelcome.vbp 进行保存。

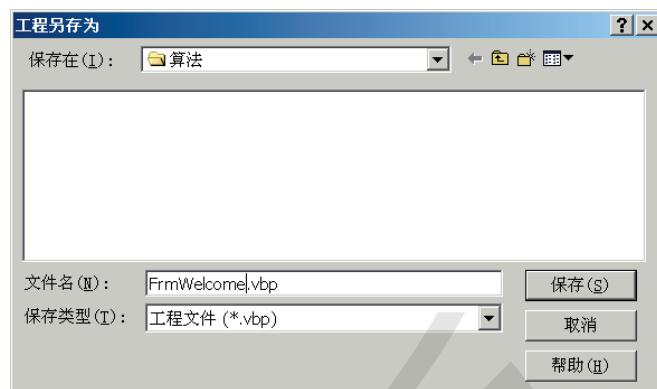


图 2-9 保存工程文件



■ VB 的可视化开发环境

VB 的可视化开发环境能够让用户方便地完成一个程序的创建、运行和调试。图 2-10 显示的是新建一个标准 EXE 工程后打开的界面，它本身也是一个标准的应用程序窗口，同样包括标题栏、菜单栏和工具栏，另外还包括窗体设计窗口、工具箱、工程资源管理窗口、属性窗口和窗体布局窗口等元素。

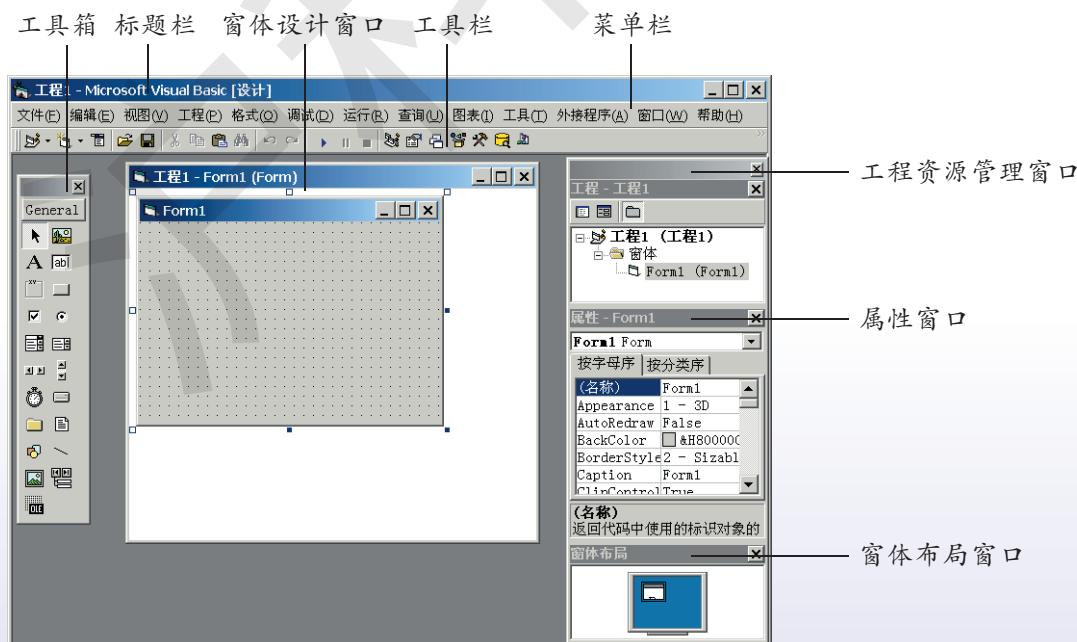


图 2-10 VB 主窗口

1. 窗体设计窗口

窗体设计窗口位于屏幕的主体位置,其中包含一个窗体(form)。应用程序的窗口就是在窗体的基础上制作完成的。

2. 工具箱

工具箱以可视化的方式列出了所有可供设计时使用的控件图标。只要在工具箱中单击选中一个控件图标,然后在窗体的合适位置拖曳鼠标就可画出一个控件。通过向窗体中添加控件,可以方便地设计出应用程序的窗口。图 2-11 给出了工具箱里各控件的名称。

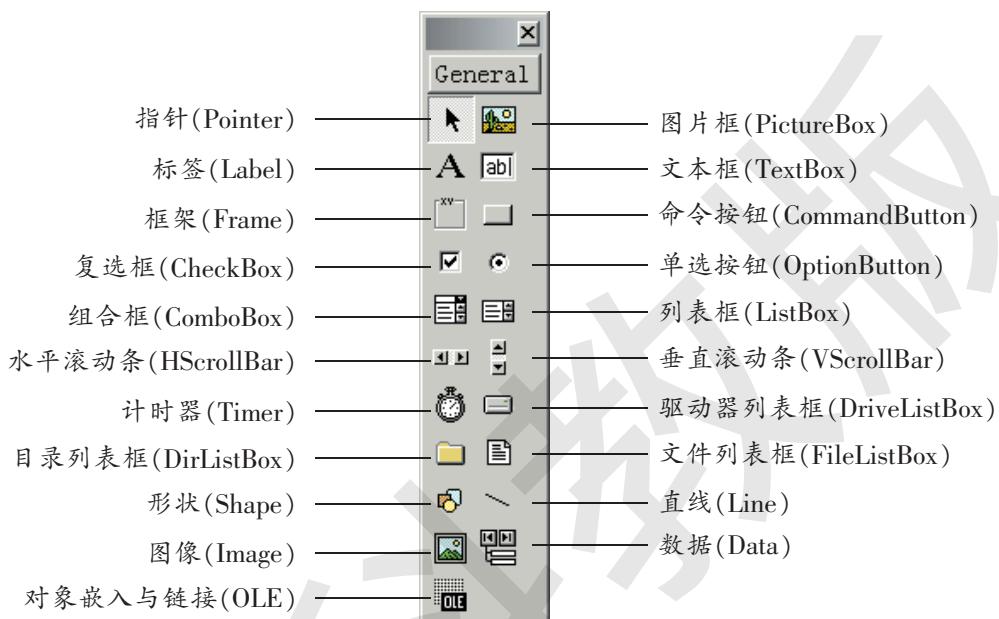


图 2-11 工具箱

3. 工程资源管理窗口

VB 使用工程来创建应用程序,工程是用来开发应用程序的文件的集合,如图 2-12 所示。当创建一个应用程序时,通常要创建一些新窗体,也可以调用其他应用程序设计好的窗体(其他模块或文件也是一样。图 2-12 中的标准模块和类模块将在以后章节介绍)。当工程的所有文件被汇集在一起之后,便可以编译工程,创建一个可执行文件。

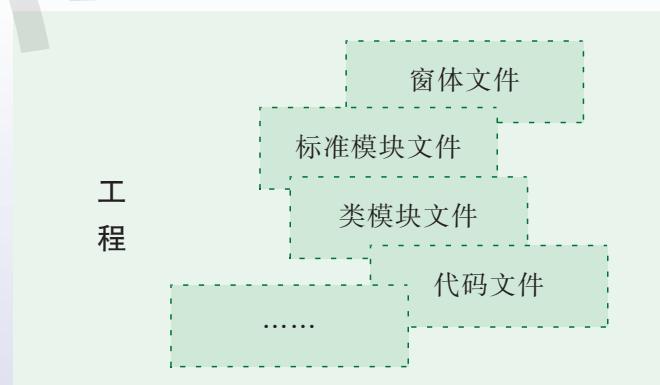


图 2-12 VB 工程包含的内容

工程资源管理窗口以树目录结构的形式显示了工程中的所有文件。如在活动一中的窗体文件和工程文件分别保存为 Welcome.frm 和 FrmWelcome.vbp 后,工程资源管理窗口中就显示这些文件的名称,如图 2-13 所示。

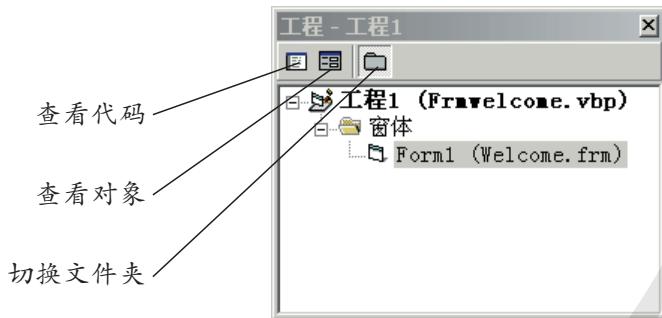


图 2-13 工程资源管理器窗口

工程资源管理窗口的标题栏下包含三个按钮。按下“查看代码”按钮将打开代码窗口;按下“查看对象”按钮将打开窗体设计窗口;按下“切换文件夹”按钮可在工程资源管理窗口下的列表框中展开树目录。

4. 属性窗口

每一个窗体和控件都有自己的属性。在属性窗口中列出了当前窗体或控件的属性及属性值,这些属性可以按字母序排列,也可按分类序排列,如图 2-14 所示。若要设置或改变某

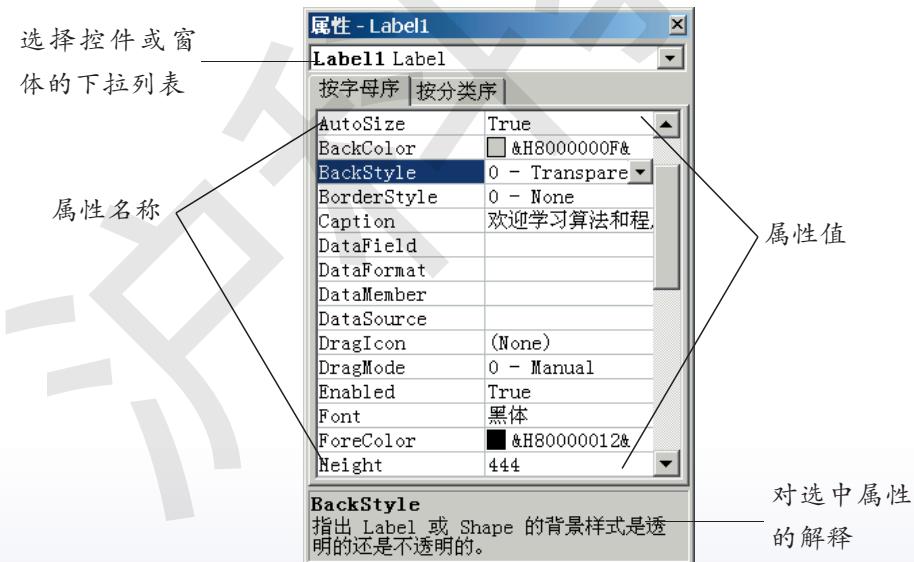


图 2-14 属性窗口

窗体或控件的属性,首先应选中该窗体或控件,有两种方法可以采用:一是在窗体设计窗口中直接单击窗体或控件,二是在属性窗口的下拉列表中选择需要进行属性设置的窗体或控件。选中后,单击属性窗口的列表中欲改变的某项属性,再对属性值进行修改。

5. 窗体布局窗口

窗体布局窗口用于设定程序运行时窗体在屏幕上的位置,默认位置为左上角。若需改

变,只要用鼠标在窗体布局窗口的小屏幕上将窗体拖曳至合适的位置即可,如图 2-15 所示。若要改变窗体的大小,则必须在窗体设计窗口中进行。

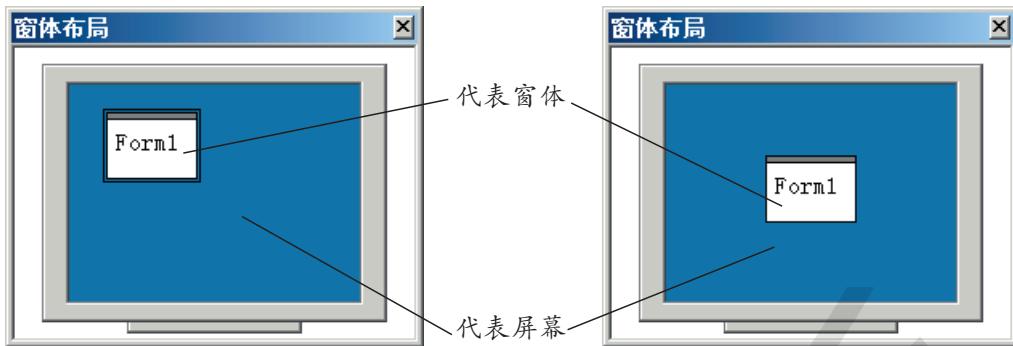


图 2-15 窗体布局窗口

■ VB 的三种工作模式

1. 设计模式

在设计模式下可以进行程序的界面设计、属性设置、代码编写等,标题栏上显示“设计”。

2. 运行模式

执行“运行”菜单中的“启动”命令或单击工具栏上的启动按钮或按 F5 键,即由设计模式进入运行模式,标题栏上显示“运行”。在此阶段可以查看程序代码,但不能修改。

3. 中断模式

当程序运行时单击“中断”按钮,或当程序出现运行错误时,都可以进入中断模式。在此模式下,运行的程序被挂起,可以查看代码、修改代码、检查数据。修改结束后,单击“继续”按钮(即“启动”按钮)可以继续程序的运行,单击“结束”按钮则结束程序的运行。

■ 可视化编程

在 VB 的集成开发环境中,程序员可以使用工具箱和属性窗口等方便地创建图形用户界面,而那些用于生成窗体的代码、用于设置窗体属性的代码、在窗体中创建和放置一个控件的代码等,都是由 VB 自动生成的,并作为工程的一个组成部分,最终编译入目标文件中,这就是可视化编程。VB 也是面向对象的编程环境。

VB 程序是由程序设计者编写的程序代码和 VB 自动生成的大量代码结合生成的,有时甚至可以不写一行代码,就得到一个完整的窗口应用程序。

1. 对象和属性

面向对象程序设计中,把现实世界中具有各种特性的实体称为 **对象** (object)(详见第 4 章第一节学习指引)。例如,在 VB 中,一个窗体是一个对象,一个控件也是一个对象。

窗体对象提供了程序运行时与用户进行交互的界面,而控件对象则具体实现了用户与应用程序的交互。用户可以使用控件对象进行输入数据、发布命令等操作,应用程序则利用控件对象显示运行信息,对用户的操作进行反馈等。

窗体中使用最频繁的控件对象是标签、文本框和命令按钮。

- 标签的作用是显示文本信息,在窗体中经常放在没有标题的控件之前,用于提示这些控件的作用。

- 文本框是让用户输入数据的最常用的控件。

- 命令按钮的作用是让用户选择程序选项,并控制程序的执行。例如,通常“退出”按钮的作用被设定为关闭一个对话框。

每一个对象都有自己的属性。例如,一个窗体对象有名称、标题、高度、位置、背景色等多个属性。

对象的属性可以在程序开发时直接在属性窗口中进行设置,也可以通过程序代码进行设置。访问对象属性的格式是:

对象名.属性名

例如:

Form1.Caption="Welcome" '设置窗体的标题为“Welcome”

Command1.Enabled=False '设置命令按钮 Command1 变灰,不能使用

在 VB 中,有很多属性是大多数对象所共有的。下表列出了一些常用的属性。

属性	功 能
Name(名称)	设置对象的名称,以标识该对象
BackColor	设置对象中显示文字和绘制图形的背景颜色
Font	设置对象中文字的字体、字号和特殊显示方式等
Caption	设置对象的显示文本,如命令按钮上的文字,复选框旁边显示的字符串等
Text	设置可接受输入的对象中输入或显示的字符串,如文本框控件中输入的文本
Height	设置对象的高度
Enable	设置对象是否可用,若设置为 False,则对象变灰,不会响应任何事件
Visible	设置对象是否可见,若设置为 False,则对象不显示

2. 方法

每一个对象都具有功能和操作,这些功能和操作被称为方法(method)。对象的方法对应于实体的行为,如老师有姓名、性别、年龄、教授的课程等属性,也有讲解、提问、辅导等行为(方法)。又如图书馆里的一本图书,它有书名、编号、作者、出版社、存放位置等属性,也能对它进行新书入库、借书、续借、还书等操作(方法)。

对窗体对象而言,窗体的大小、名称、标题及在屏幕中的位置等都是它的属性,在窗体中显示文字、清除文字、画线或关闭该窗体等操作都称为方法。例如,显示窗体 Form1 可调用 Show 方法,隐藏窗体则可调用 Hide 方法。对象方法的调用都是在程序中实现的,调用对象方法的格式是:

对象名.方法名

例如：

```
Form1.Show    '显示窗体 Form1
Form1.Hide    '隐藏窗体 Form1
```

3. 事件和事件驱动机制

在运行本节活动中的程序时,当用户单击“退出”按钮后,应用程序的执行就结束了,这是应用程序的事件驱动机制在工作。因为用户在做“单击”的动作时,产生了一个“单击”事件,该事件能被命令按钮对象所识别,并传递一条描述这个事件的消息到应用程序。应用程序接收到该事件发生的消息后,调用执行相应的处理该事件的程序代码,即事件过程(本节活动中为 Command1_Click 过程),而这个过程中只有一条 End 语句,其作用就是关闭应用程序。

事件(event)是用户与对象或对象与对象之间的交互动作,消息(message)则是描述事件发生的信息。例如,用户单击鼠标时,系统就会产生一条特定的消息,标识鼠标单击事件的发生。

在 VB 中,系统为每个对象预先定义好了一系列的事件,如单击(Click)事件、双击(Dblclick)事件、装入(Load)事件、改变(Change)事件、鼠标移动(MouseMove)事件、获取焦点(GotFocus)事件等。

VB 中的每一个窗体和控件都有一个预定义的事件集合。例如,针对命令按钮,VB 在代码窗口中提供了各种事件,如图 2-16 所示。

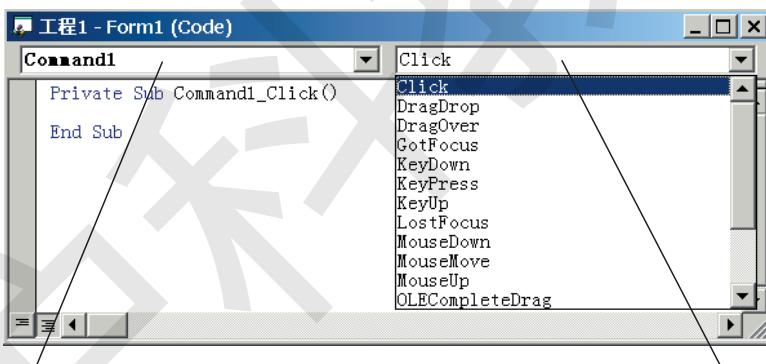


图 2-16 VB 事件

事件驱动(event-driven)是指应用程序系统接收到消息后,自动调用相应的事件过程进行响应。事件驱动机制是应用程序的运行机制。打开一个应用程序后,该程序即处于等待状态,当用户或系统触发某个事件时,应用程序会接收到事件发生的消息,并自动查找与消息对应的事件过程,如果预先定义了这样的事件过程,系统就会自动调用执行事件过程中的程序代码。

事件名称	激活方式
Load	使用 Load 语句启动应用程序,或引用未装载的窗体或控件
KeyPress	用户单击键盘上的一个键

(续表)

事件名称	激活方式
Click	用户在一个对象上单击鼠标按钮
GotFocus	通过 TAB 键切换或单击对象之类动作使对象获得焦点, 也可以在代码中用 SetFocus 的方法改变焦点
Activate	激活某一对象

在 VB 中, 各对象能响应的事件种类很多, 上表列出了一些常用的事件。

■ VB 可视化编程的一般步骤

在编写 VB 程序的时候, 一般需要执行的基本步骤是:

- (1) 创建一个新的工程。
- (2) 通过在窗体中添加各种控件, 创建图形用户界面, 如向窗体中添加标签、文本框、命令按钮等。
- (3) 在属性窗口中设置各对象的属性, 如修改窗体的背景颜色, 设置窗体和标签的显示文本以及调整显示文本的字体和大小等。
- (4) 编写程序代码, 并在代码窗口中进行输入。
- (5) 调试、修改程序, 直至程序能够正确运行。
- (6) 保存工程及相应的文件。

体验活动

1. 对本节活动中的窗体作进一步修改。

- (1) 添加一幅图片, 将标签文本分为两行, 移至合适的位置, 再添加另一个命令按钮“进入”, 并改变两个按钮及标签显示文本的颜色(如图 2-17 所示)。

提示: 需要使用工具箱中的图片框来添加图片。



图 2-17 程序运行界面

下页续

接上页

(2) 如果由你来设计活动中的窗体,你会作哪些添加和修改?

		窗体	标签	命令按钮
修改	1			
	2			
修改的方法	1			
	2			
添加	1			
	2			

2. 创建如图 2-18 所示的图形用户界面,打开代码窗口,逐一观察窗体、标签、文本框、命令按钮的主要事件过程,根据帮助菜单,写出下表中各事件过程的含义。



图 2-18 用户登录界面

事件过程	事件描述
Form_Load()	
Label1_Click()	
Text1_Change()	
Text1_Gotfocus()	
Command1_Click()	

提示:窗体上有三个标签、两个文本框和两个命令按钮。文本框的 Text 属性用于设置显示文本,需要作修改。

第二节 VB语言概述

VB提供的是可视化的开发环境,程序设计者可以像“画画”一样直接在窗体中构建出应用程序的界面,所以能把更多的精力放在编写实现程序功能的代码上,这也是整个程序设计的关键。VB语言的语法是最容易被初学者所接受的,即使从没学过任何一种编程语言,也能快速上手。

活动一 观摩、分析程序语句

以下是小明利用计算机解决某问题的过程。

提 示 板

程序由8行代码组成,每行均为VB的一个语句。以单引号'开头的语句为**注释语句**,供阅读理解程序用。计算机在执行程序时会自动跳过。

问题

已知某段时间美元兑人民币的汇率是8.2765,编写一个将美元兑换成人民币的汇率转换程序。

分析问题

设用usd代表美元金额,用rmb代表人民币金额。
美元兑人民币的公式为:rmb=usd×8.2765。

设计算法

① 输入usd的值;
② $rmb = usd \times 8.2765$;
③ 输出rmb的值。

编写程序

```

① Private Sub Form_Click() '定义单击窗体事件过程
② Dim usd As Single '定义变量usd,代表美元金额
③ Dim rmb As Single '定义变量rmb,代表人民币金额
④ Const RA AS Single= 8.2765 '定义常量RA,代表汇率
⑤ usd = Val(InputBox("输入美元")) '在输入对话框中输入美元金额并赋给变量usd
⑥ rmb = usd * RA '计算usd与RA的乘积,将值赋给变量rmb
⑦ Print "可兑换人民币" & rmb & "元" '把rmb的值输出到窗体上
⑧ End Sub '过程结束

```

提 示 板

程序第②、③、④句中有Single字样,表示的是单精度实型数据类型。关于数据类型详见**学习指引**。



1. 仔细阅读以上代码,参考其中的注释语句,了解各语句的含义。

2. 新建一个工程,执行“视图”菜单中的“代码窗口”命令,打开代码窗口,输入以上代码。

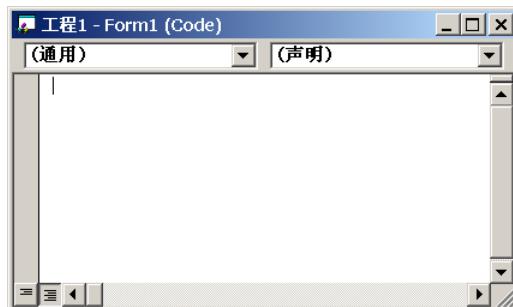


图 2-19 代码窗口

3. 调试程序,观察程序是否顺利运行。若出现错误,请检查代码的输入是否有误。

4. 依次将窗体文件、工程文件以文件名 change.frm、change.vbp 进行保存。

5. 了解数据类型的作用。

- (1) 运行程序,输入不同的数据,观察运行结果。

输入美元值为 1238 时,运行结果为 _____;

输入美元值为 5000 时,运行结果为 _____。

- (2) 将变量 rmb 的数据类型改为整型(Integer),即将代码窗口中的“Dim rmb As Single”改为“Dim rmb As Integer”,再次输入数据,观察运行结果。

输入美元值为 1238 时,运行结果为 _____;

输入美元值为 5000 时,运行结果为 _____。



讨论

为什么修改了变量 rmb 的数据类型后,运行结果会有差异?

6. 了解数据输入、处理和输出语句。

- (1) 按 F8 键或执行“调试”菜单下的“逐语句”命令进入调试状态。观察各语句的执行情况,加深对各语句功能的理解。

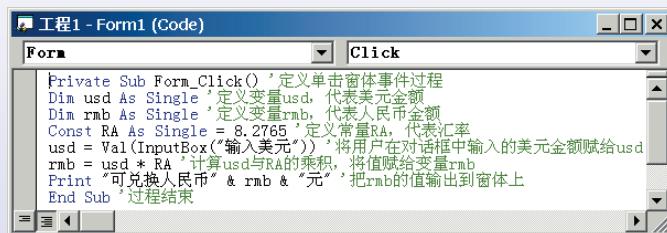


图 2-20 观察语句执行顺序

提 示 板

这些语句按出现的顺序逐句执行,这种程序结构便是顺序结构。顺序结构的语句实现详见学习指引。

(2) 向监视窗口中添加变量 usd、rmb, 按 F8 键逐语句调试程序, 观察监视窗口中变量 rmb、usd 值的变化。

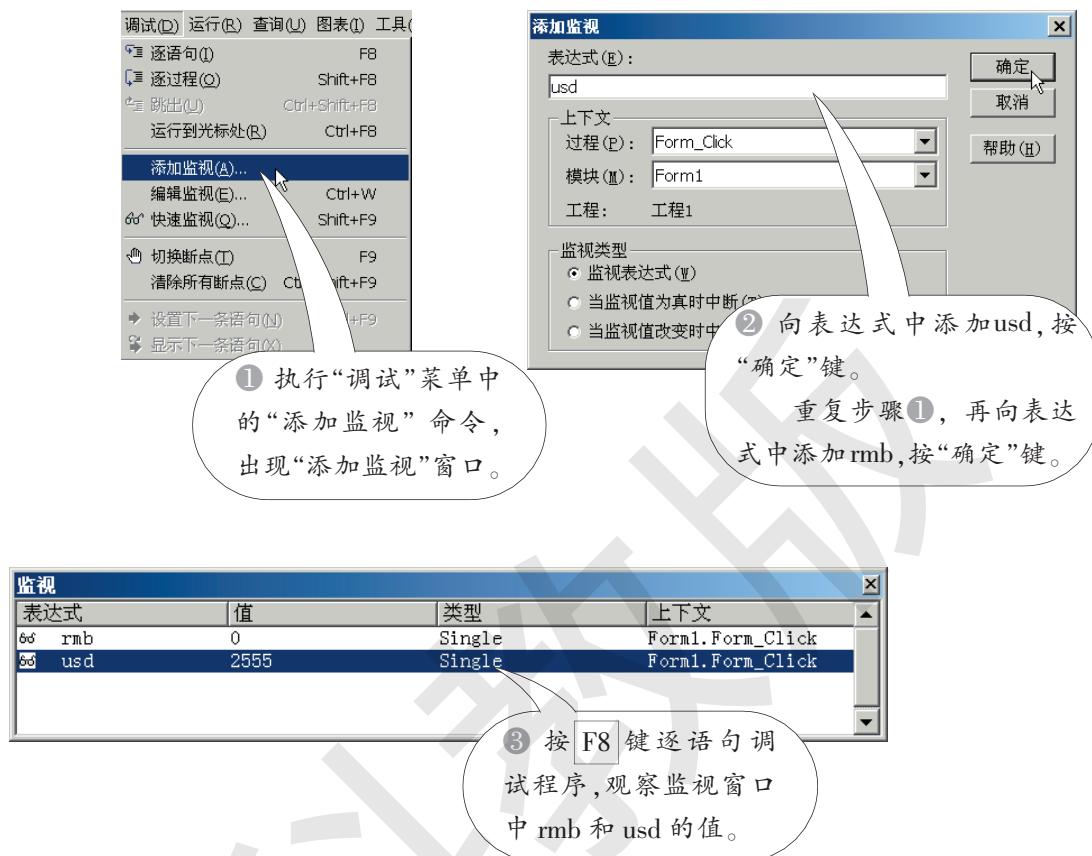


图 2-21 添加监视, 观察变量值的变化

提 示 板

InputBox()是提供从键盘输入数据的系统函数, 程序运行时将出现输入对话框。“=”为赋值运算符。InputBox()返回的数据为字符串型, Val()函数能将数字字符串型数据转换为实型数据, 或将非数字字符串型数据转换为数值 0。详见学习指引。

若输入美元值为 2555, 变量 usd 在执行 _____ 语句时, 其值由 0 变化为 _____, 变量 rmb 在执行 _____ 语句, 其值由 0 变化为 _____。

→ (3) 运行程序, 分别向输入对话框中输入“2000”和“2000 元”, 观察运行结果; 将“usd = Val(InputBox("输入美元"))”修改为“usd = InputBox("输入美元")”, 再次运行程序, 分别向输入对话框中输入“2000”和“2000 元”, 将运行结果填入下表。

	输入“2000”	输入“2000 元”
原程序		
修改后的程序		

(4) 将“Print"可兑换人民币" & rmb & "元"”修改为“Print "可兑换人民币" & Format (rmb,"0.00") & "元" ”,上机调试程序,观察运行结果,分析添加这条语句的好处。

(5) 模仿“Print "可兑换人民币" & rmb & "元"”语句,写出输出格式为“××美元可兑换××元人民币”的语句,并上机验证。

活动二 修改、完善程序

活动一中将美元兑人民币的汇率设为常量,但实际上汇率是一直波动的,汇率的值应改为程序运行时由用户输入。小明打算对程序进行修改。

1. 分析问题。

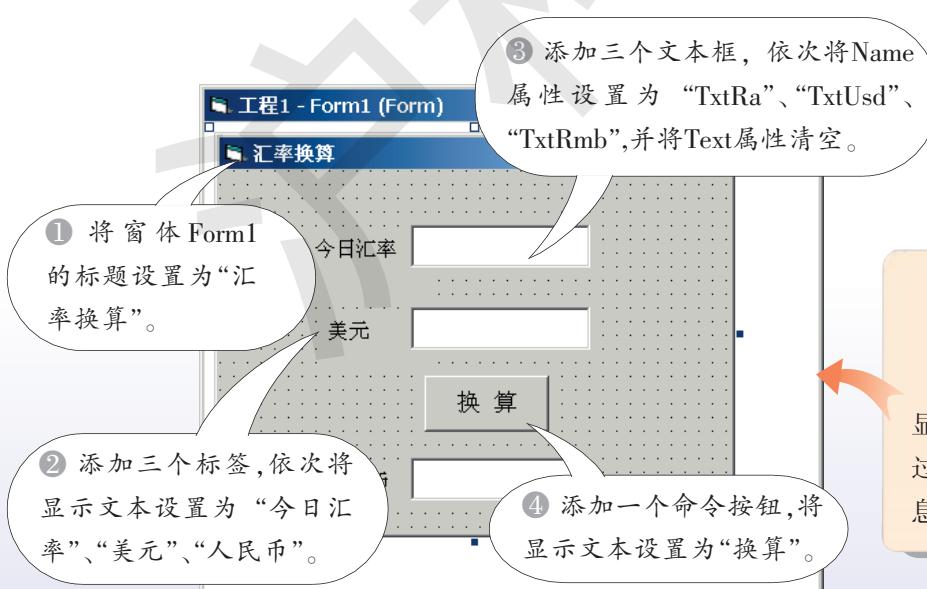
(1) 因为汇率的值不固定,在程序运行时由用户输入,所以应将活动一中定义为常量的汇率RA改为变量,设用ra表示。

(2) 为方便用户输入汇率和美元值,以及观察换算结果,考虑使输入与输出在同一窗体中实现。

2. 实现应用程序。

(1) 打开活动一中的工程文件change.vbp。

(2) 设计图形用户界面。向窗体中添加控件,并设置窗体及控件的属性。



提示板

Format()函数可以设置常量、变量或表达式值的输出格式,它的使用方法是:

Format(常量或变量,格式控制串)

提示板

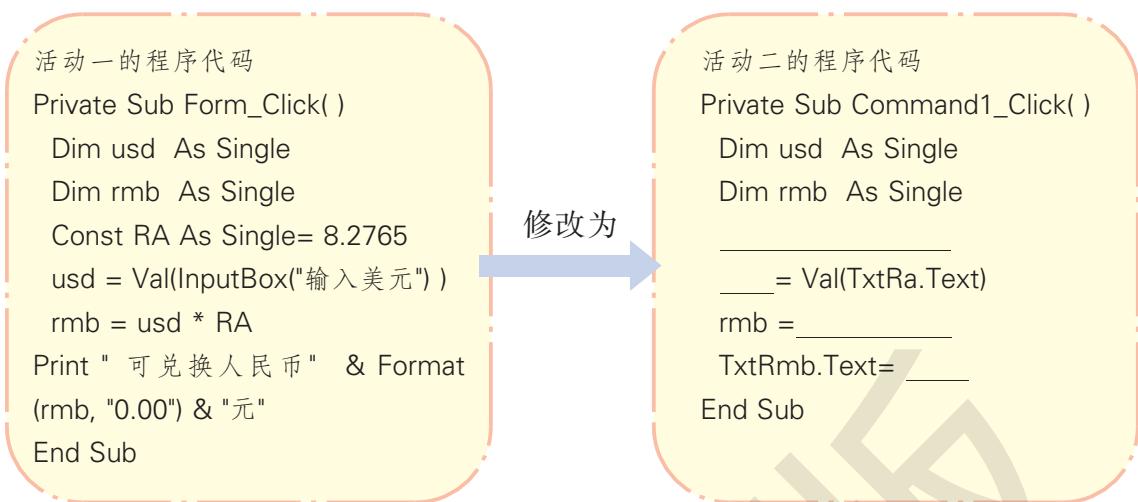
Print是输出数据、文本的一个重要方法。详见学习指引。

提示板

文本框控件 `TextBox` 被用来显示用户输入的信息,或者通过给 Text 属性赋值来输出信息。

图 2-22 制作界面

(3) 修改程序代码,并与活动一的程序代码进行对比。



(4) 打开代码窗口,删除Form_Click()过程中的代码。在左边的下拉框中选择Command1,在右边的下拉框中选择Click(如图2-23所示),在编辑区中输入新的程序代码。



图 2-23 代码窗口

3. 调试、运行程序。



图 2-24 程序运行界面

4. 保存工程。



本节活动一的美元兑换人民币程序有8行代码,每行代码均为VB的一个语句,每一个语句都须按照一定的规则书写。VB的程序就是由这样的一个个语句组成的。程序包括常量、变量说明语句,输入语句,赋值语句和输出语句。

综上所述,VB的程序一般包含如下部分:

- (1) 数据类型说明部分;
- (2) 数据的输入部分;
- (3) 数据的处理部分;
- (4) 数据的输出部分。

■ 数据类型和常量、变量

1. 数据类型

在计算机程序设计中,数据是一个非常广义的概念,任何符号(如数字、英文字母、汉字或某些特殊符号等)只要将它赋予一定含义,就是数据。

为了更好地处理各种数据,VB系统预先定义了多种基本数据类型。VB中的基本数据类型一般可分为三大类:数值型、字符串型,以及不属于以上两类的其他类型,如下表所示。活动一的程序中就用到了单精度型的数据。

数据类型		类型标识符	说 明	举 例
数 值 型	整 型	Integer	用2个字节存储,可以表示-32768至32767之间的整数	56,-145
	长整型	Long	用4个字节存储,可以表示-2147483648至2147483647之间的整数	1874738,-82273
	单精度型	Single	用4个字节存储,可以表示-3.37E+38至3.37E+38之间的实数	3.67,-76.88
	双精度型	Double	用8个字节存储,可以表示-1.67E+308至1.67E+308之间的实数	8.7E+100
字符串型		String	每个字符占一个字节,用双引号括起来,可以储存0至65535个字符	"animal", "It's time to study"
其他	布尔型	Bool	又称逻辑型,用2个字节存储,其值只有True和False	True, False
	日期型	Date	用8个字节存储,存放日期数据	#1982-7-18#

不同类型数据的运算特性是不相同的。例如,数值型数据通常进行加减乘除四则运算操作,而字符串型数据通常进行串连接、子串查找等操作,但不能进行四则运算。

2. 常量和变量

程序运行时,输入的数据、参加运算的数据和运行的结果包括中间结果等数据都需要存

储在计算机的内存中。**常量**(constant)是一个内存单元,在程序运行的过程中,其中的数据不会改变。例如,活动一程序中的 8.2765、“输入美元”等均为常量,而 RA 被定义为常量,其值是 8.2765。**变量**(variable)是一个命名的内存单元,在程序运行的过程中,其中的数据可以改变。例如,活动一程序中的变量 usd 存放的是用户输入的美元数据,rmr 存放的是换算成的人民币数据。

在程序中,常量和变量都需要命名,VB 的命名规则如下:

- (1) 必须以字母开头,由字母、数字或下划线组成,长度不超过 255 个字符;
- (2) 不能使用 VB 中的关键字(系统已使用的专用名称,如 Dim、Single、As 等),并尽量不与 VB 中标准函数名同名;
- (3) VB 中不区分常量、变量的大小写,但常量一般用大写字母表示。如,本节的汇率转换程序中常量名为 RA,变量名为 rmb 等。

3. 常量的表示

常量包括直接常量和符号常量。其中直接常量是指在程序中直接给出的数据,如活动一中的 8.2765、“输入美元”。符号常量用一个标识符表示,如活动一中的 RA,符号常量说明语句的格式如下:

Const 常量名 [AS 数据类型]=表达式

其中,表达式可以是数值常量、字符串常量或以这些常量及运算符组成的式子,数据类型的定义可以省略,系统会根据表达式的值来确定常量的类型。

例如,进行数学运算时经常要用到 π 的值,可以先定义一个符号常量 Pi,在每次都需要输入 3.1415926 的地方用常量名 Pi 代替,如求圆的面积可以用代码表示为:

Const Pi AS Double =3.1415926

Area=Pi*Radius* Radius

很明显,使用符号常量的语句比不使用符号常量的语句 Area=3.1415926*Radius* Radius 更容易理解,输入更简洁,且不易出差错。

各种数据类型的常量表示如下页表。

4. 变量的定义

使用变量前,一般应该先对变量的数据类型和变量名进行定义,以使系统为其分配存储单元。变量说明语句的格式为:

Dim 变量名 [As 数据类型]

例如: Dim n As Integer '定义一个整型变量 n

Dim height As Single '定义一个单精度型变量 height

Dim StuName As String '定义一个字符串型变量 StuName

Dim i As Integer, j As Integer '定义两个整型变量 i 和 j

Dim 语句可以同时定义多个变量,但各个变量不可共享数据类型。例如,上述语句 Dim i As Integer, j As Integer 定义的是两个变量 i 和 j,且分别被定义为整型变量。再如,语句 Dim i, j As Integer 定义的变量也有两个:i 和 j,但由于只给出一个数据类型 Integer,因此该语句实际为:定义 j 为整型,定义 i 为 Variant 型(Variant 是 VB 的缺省数据类型,可存储各种数据,但所占存储单元比其他类型都多)。

数据类型		说 明	举 例
整数型	实数型实数	只能包含数字0~9和正负号(正号可以省略)	85,-5675
	八进制	由数字0~7组成,并以“&O”或“&”为引导,其后面的数据位数小于等于6,范围为&00~&0177777	&0746
	十六进制	由数字0~9、字母A~F组成,并以“&H”为引导,其后面的数据位数小于等于4,范围为&H0~&HFFFF	&H7AF8
实数型	十进制形式	由小数部分和整数部分构成	3.14,-23.4567
	指数形式	以“E”表示底数10,E前面是小数部分,后面是指数部分,都不能省略,且E的前后不能有空格	1.568E-46,120.098E5
字符串型	用双引号括起来的一行字符,其中不包括任何字符的字符串称为空串		"China","华东师范大学",""
布尔型	只有True和False两个值		True, False
日期型	表示日期和时间,用一对#括起来		#2/14/2004 6:30:56 PM#, #10/1/2005#

另外,在为一个变量选择数据类型时,必须考虑它的可能大小。不同的数据类型可以表示的数值范围是不同的。例如,整数类型数据的最高位表示符号,其余各位表示数值。其中整型用2个字节来存储一个整数,所以它可以表示的数值范围是 -2^{15} 到 $2^{15}-1$;长整型用4个字节来存储一个整数,所以它可以表示的数值范围是 -2^{31} 到 $2^{31}-1$ 。又如,实数类型数据按科学计数法分成三部分:符号位、小数部分和指数部分,所以虽然单精度型与长整型同样是用4个字节存储,但单精度型可以表示的数值范围要比长整型大得多。本节活动一中,当变量rmb的数据类型改为整型后,在输入5000时,运行结果会显示溢出,即为存储空间不够所致。

■ 运算符、函数和表达式

1. 运算符

运算是对数据进行加工处理的过程,描述各种不同运算的符号称为**运算符**(operator)。程序设计中最常用的运算符主要包括算术运算符、关系运算符、逻辑运算符、字符串运算符等。本节主要介绍算术运算符和字符串运算符,其他的留待后面学习。

(1) 算术运算符。

算术运算符是最常用的运算符。8种主要的算术运算符如下表所示。

需要说明的是:

- ① 当指数运算中的指数是一个算式时,必须加上括号,例如 $2^(1/3)$ 。
- ② 注意区分除法运算符(/)和整除运算符(\),其中整除是指两个整数相除后得到的值的整数部分,例如1/2的值是0.5,而1\2的值是0。
- ③ 取模(即取余数)的运算只能在两个整数之间进行。
- ④ 算术运算符的运算规则与数学中相同。表中算术运算符的优先级由上到下递减,但

运算符	含义	优先级	示例	结果
$^$	指数	1	3^2	9
$-$	取负	2	-3	-3
$*$	乘	3	$3*3*3$	27
$/$	除	3	$10/3$	3.33333333333333
\backslash	整除	4	$10\backslash3$	3
Mod	取模	5	$10\text{Mod}3$	1
$+$	加	6	$10+3$	13
$-$	减	6	$3-10$	-7

括号的优先级最高,在多种运算符的混合运算中,应该更多地使用括号来明确运算的顺序。VB 中只用一种括号“()”,它可以多次嵌套使用。

(2) 字符串运算符。

字符串运算符可以对字符串型数据进行串连接操作,即将两个或多个字符串连接成一个字符串,主要有“&”和“+”两种。

“&”两旁的数据不管是字符串型还是数值型,系统都将其作为字符串型,然后进行连接,如“祝贺” & “中国” & “申奥成功”,结果为“祝贺中国申奥成功”;“123” & 45,结果为“12345”;“abc” & 12,结果为“abc12”。

“+”的情况比较复杂。如果两旁的数据均为字符串型,则将两个字符串进行连接,如“中国”+“上海”,结果为“中国上海”;如果两旁的数据均为数值型,则进行算术加法运算,如123+45,结果为168;如果两旁的数据一个为数字字符串型,一个为数值型,系统自动将数字字符串转换为数值,然后进行算术加法运算,如“123”+45,结果为168;如果两旁的数据一个为非数字字符串型,一个为数值型,如“abc”+12,则出错。

本节活动中显示美元兑换人民币的计算结果时,代码表示为:

“可兑换人民币” & rmb & “元”

其中 rmb 是一个数值型变量,“可兑换人民币”和“元”是两个非数字字符串型的字符串。用“&”运算符连接时,系统会先将 rmb 的值转化为字符串型,再连接前后两个字符串;但如果将“&”换为“+”,就会产生编译错误。

2. 系统函数

在 VB 中有两类函数,分别是系统函数和自定义函数。**系统函数**又称为内部函数,是系统预先定义、由程序系统内部提供的一些程序段,存放在函数库中,供程序员直接通过函数名调用。自定义函数将在第 3 章中介绍。

系统函数主要包括数学函数、字符串函数、日期函数、转换函数和测试函数等,每个系统函数完成一个独立的功能。调用函数的格式为:

函数名(参数值列表)

其中,参数是指在调用函数时交给函数处理的数据。

VB 提供的常用系统函数如下页表所示。

函数类型	函数名	功 能
数 学 函 数	Sqr(x)	求平方根
	Abs(x)	求绝对值
	Log(x)	求以 e 为底的对数
	Sin(x)	求正弦值
	Cos(x)	求余弦值
	Tan(x)	求正切值
	Atn(x)	求反正切值
字 符 串 函 数	Len(x)	求字符串的字符长度
	LenB(x)	求字符串的字节数
	Left(x,n)	从字符串左边取 n 个字符
	Right(x,n)	从字符串右边取 n 个字符
	Ucase(x)	将字符串中所有小写字母改为大写
	Lcase(x)	将字符串中所有大写字母改为小写
日 期 函 数	Date()	返回系统日期
	Time()	返回系统时间
	Month(C)	返回月份(1~12)
	Year(C)	返回年份(1752~2078)
	Day(C)	返回日期(1~31)
转 换 函 数	Str(x)	将数值数据转换成字符串
	Val(x)	将字符串中的数字转换成数值
	Chr(x)	返回以 x 为 ASCII 码的字符
	Asc(x)	给出字符 x 的 ASCII 码
	Cint(x)	将数值型数据 x 的小数部分四舍五入取整
	Int(x)	取小于等于 x 的最大整数
测 试 函 数	Err()	发生错误时,返回一个错误号
	Err1()	发生错误时,返回发生错误的行号
	Error, error\$	返回与错误号相对应的错误信息

3. 表达式

表达式(expression)指由运算符及括号连接各种运算元素组成的有意义的式子。运算元素可以是常量、变量及函数。根据运算元素和运算符号的不同,表达式有数值表达式、字符表达式等。数值表达式由数值型数和算术运算符组成,表达式的值为数值型数,如活动一程序中的 `usd * RA`,当 `usd` 的值为 2000 时结果为 1 6553; `100\10 Mod(3*2)` 的结果为 4。字符表

达式由字符串运算符和字符型数据组成,表达式的值为字符型,如"祝贺" & "中国" & "申奥成功",结果为“祝贺中国申奥成功”。

■ 赋值语句与输入输出语句

1. 赋值语句

赋值语句是程序设计语言中最基本的语句,它的作用就是把某一值赋给某一变量。VB 中赋值语句的格式为:

变量名=表达式

其中“=”是赋值运算符,它将右边表达式的值赋给左边的变量。

本节活动程序中的 $rmr = usd * RA$,就是将表达式 $usd * RA$ 的值求出,然后赋给变量 rmr 。值得一提的是,赋值语句中,赋值运算符右边的值赋给左边的变量后,原右边的值仍保持不变。例如:

- (1) $a=15$,表示将数值 15 赋给变量 a。
- (2) $b=a$,表示将变量 a 的值赋给变量 b,但 a 的值不改变。
- (3) $b=b+1$,表示将变量 b 的值加 1 后,赋给变量 b。

如果原来 b 的值为 10,则执行此语句后 b 的值为 11,如图 2-25 所示。

下面一组代码完成的是变量 i 和 j 的赋值、计算及交换操作,各语句的作用如图 2-26 所示:



图 2-25

```
'为变量 i,j,t 分配三个整型的存储单元
Dim i AS Integer, j AS Integer, t AS Integer
```

① $j=5$

② $i=j$

③ $i=i+5$

'④~⑥句执行交换操作,交换变量 i 和变量 j 的值

④ $t=j$

⑤ $j=i$

⑥ $i=t$

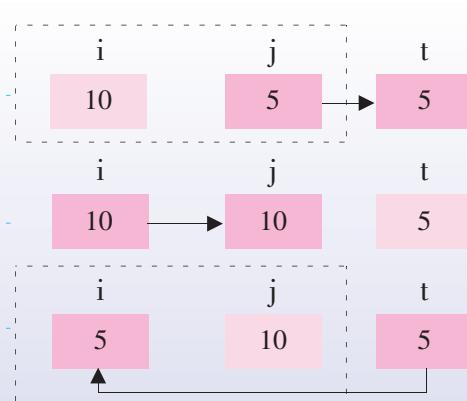
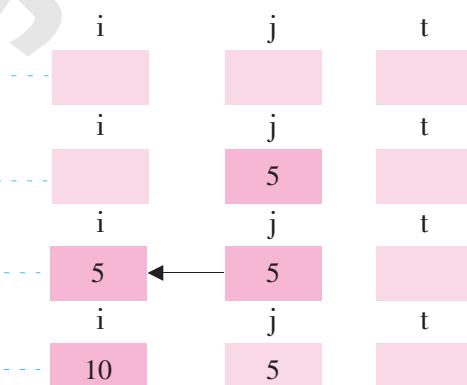


图 2-26 语句作用示意

使用赋值语句要注意的是：

(1) 在 VB 中规定：如果数值变量没有赋值，其值为 0；如果字符串变量没有赋值，其值为空串。

(2) 赋值运算符两边的数据类型要一致，否则可能发生“类型不匹配”的错误，例如：

```
Dim x As Integer
x="string"
```

第 2 句在运行的时候就会发生“类型不匹配”的错误，程序的运行将中断。

2. 输入语句

InputBox()是提供从键盘输入数据的系统函数。InputBox()函数执行时将产生一个对话框，用户可以在对话框中输入数据，该函数的返回值就是用户输入的字符串。

InputBox 函数的语法格式为：

```
InputBox(提示信息, 标题, 默认值, 位置)
```

其中，提示信息是指显示在对话框中的提示语句，不能省略；标题是指弹出的对话框的标题；默认值是指默认的输入数据值；位置是指弹出的对话框的左上角在屏幕上的位置。

如以下代码运行时，在屏幕坐标(100,30)处将出现如图 2-27 所示的输入对话框。



图 2-27 “输入”对话框

```
Dim username As String
username = InputBox("请输入用户名", "输入", "标准用户", 100, 30)
```

3. 输出语句

(1) Print 方法。

Print 是输出数据、文本的一个重要方法。Print 方法输出数据的基本格式是：

```
Print 输出项 1[;输出项 2;…输出项 n]
```

如果输出内容是数、变量或表达式，则输出数、变量或表达式的值；如果输出内容是用引号括起来的字符串，则按照原样输出该字符串。多个输出内容之间可以用逗号或分号分隔。如果用分号，则后面一个输出紧跟前一个输出；如果用逗号，则按照分区格式输出。

例如，Print "可兑换人民币" & rmb & "元" 就是将计算出来的变量 rmb 的值以“可兑换人民币××元”的形式输出到窗体上。

(2) MsgBox 函数。

MsgBox 函数的作用是弹出一个对话框，以显示字符串。例如，可以用 MsgBox 函数显示本节活动的结果。

```
MsgBox "可兑换人民币" & rmb & "元"
```

需要注意的是:使用 InputBox 和 Print 语句进行输入输出,并不能创建友好的用户交互界面。在 VB 可视化编程中,更多地是采用丰富多彩的控件来对界面进行设计。例如,活动二中使用了三个文本框,一个文本框用于输入当天的汇率,一个文本框用于输入美元值,一个文本框用于输出换算后的人民币值。使用文本框的 Text 属性可以方便地获取或改变文本框的值。常用的输入输出控件有:文本框、命令按钮、标签、列表框、单选按钮、复选框和组合框。

本节活动程序中的语句都是按其出现的顺序逐句执行的,执行完第一句后执行第二句,然后执行第三句,依次执行下去,直到执行完最后一句,这种结构便是顺序结构,通常由输入、输出、赋值等语句实现。

■ VB 程序的调试

程序编写好后,需要试运行,以便发现错误和修改错误,这种边运行边修改直到程序运行正确的过程就是调试程序。

1. 设计时的常见错误

在用 VB 进行程序设计时,常见的错误主要有以下几种:

(1) 编辑时的错误。

在代码窗口输入或编辑程序代码时,VB 会对程序进行语法检查,当发现语句没有输完、关键字输入错误等情况时,系统会弹出对话框,提示出错,并在错误处加亮显示,以便修改。

(2) 运行时的错误。

在运行程序时发生的错误,一般是由遗漏关键字或指令代码执行了非法操作等引起的,如缺少 End If 语句、数据类型不匹配、试图打开一个不存在的文件等。系统会报错并加亮显示,等候处理。

(3) 逻辑错误。

如果程序运行后得不到所希望的正确结果,则说明存在逻辑错误。逻辑错误产生的原因很多,如运算符使用不正确、语句的次序不对、循环语句中的初值或终值不正确等。对于逻辑错误,系统是检查不出来的,因此也不会有提示信息,需要用户自己分析阅读程序,并进行认真细致的调试。

2. VB 的调试工具栏

VB 提供了一些调试工具,其调试工具栏如图 2-28 所示。

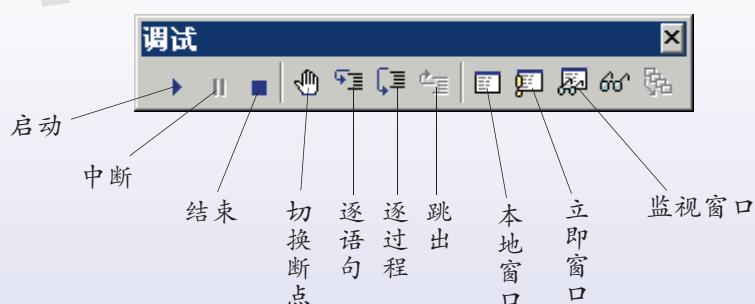


图 2-28 “调试”工具栏

利用该工具栏可以进行启动程序、中断运行、在程序中设置间断点、监视变量、单步调试、过程跟踪等操作,为程序调试提供方便。

3. 程序调试方法

(1) 进入/退出中断状态。

进入中断状态有四种方法:

- ① 程序运行时发生错误自动进入中断;
- ② 程序运行中用户按中断键强制进入中断;
- ③ 用户在程序中预先设置断点,程序执行到断点处即进入中断状态;
- ④ 采用单步调试方式,每运行一句可执行代码后,即进入中断状态。

(2) 利用调试窗口。

① 立即窗口。这是调试窗口中使用最方便、也是最常用的窗口。可以在程序中用 Debug.Print 方法,把输出送到立即窗口,也可以在该窗口中直接使用 Print 语句或“?”显示变量的值。

② 本地窗口。该窗口显示当前过程中所有变量的值,当程序的执行从一个过程切换到另一个过程时,该窗口的内容会发生改变。本地窗口只反映当前过程中可用的变量。

③ 监视窗口。该窗口可显示当前的监视表达式的值,但在此之前,必须在设计阶段利用“调试”菜单中的“添加监视”命令或“快速监视”命令添加监视表达式。

(3) 插入断点和逐句跟踪。在调试程序时,通常会设置断点来中断程序的运行,然后逐句跟踪检查相关变量、表达式的值是否在预期的范围内。

在中断模式或设计模式下可以设置或删除断点。在代码窗口中选择怀疑存在问题的语句,按 F9 键将该语句设置为断点,则程序运行到断点处即停下,进入中断模式。

体验活动

1. 观察下列程序的运行结果,体会各语句的作用。

(1) 比较程序中两条输入语句的执行效果。

```
Private Sub Form_Click()
    Dim a As Double
    Dim b As String
    a = Val(InputBox("a=", "输入数字", 100))
    b = InputBox("b=")
End Sub
```

(2) 分析 Print 语句中各符号对显示的影响。

```
Private Sub Form_Click()
    Const a As Integer = 5
    Const b As Integer = 8
    Print "a="; a; "b="; b
End Sub
```

下页续

接上页

```
Print "a="; a, "b="; b
Print "a="; a,
Print "b="; b
Print "a="; a
Print "b="; b
End Sub
```

2. 阅读以下程序,写出运行结果,说明程序的功能,并上机验证。

```
Private Sub Form_Click()
    Dim a As Double
    Dim b As Double
    Dim Max As Double
    Dim Min As Double
    a = Val(InputBox("a=", "输入"))
    b = Val(InputBox("b=", "输入"))
    Print "a="; a; "b="; b
    Max = (a + b) / 2 + Abs(a - b) / 2
    Min = (a + b) / 2 - Abs(a - b) / 2
    Print "max="; Max; "min="; Min
End Sub
```

3. 阅读以下程序,写出运行结果,说明程序的功能。

```
Private Sub Form_Click()
    Dim a As Double
    Dim b As Double
    Dim c As Double
    a = Val(InputBox("a=", "输入数字", 100))
    b = Val(InputBox("b=", "输入数字", 200))
    Print "a=" & a & " b=" & b
    c = a
    a = b
    b = c
    Print "a=" & a & " b=" & b
End Sub
```

4. 试编写一个程序,当输入正方形的边长时,输出正方形的周长和面积。

5. 已知三种水果糖每500克价格分别为10元、12元、16元,为糖果店的营业员设计一个程序,计算顾客购买各种水果糖后一共要付多少钱。

6. 从键盘上输入小时、分和秒,把它们转换为秒数并输出。

第三节 选择结构

在日常生活中,常常会面临这样、那样的选择。例如,如果星期天不下雨,就去植物园看花展,否则就在家看书。又如,运动会上,如果获得一等奖,就奖励一套图书;如果获得二等奖,就奖励一个足球;如果获得三等奖,就奖励一套文具。

计算机在解决这类两者选一或者具有多种条件选择的问题时,需要用到选择结构。采用选择结构的程序通过对具体条件的判断,选择程序的流程,实现根据不同的情况执行不同的操作。

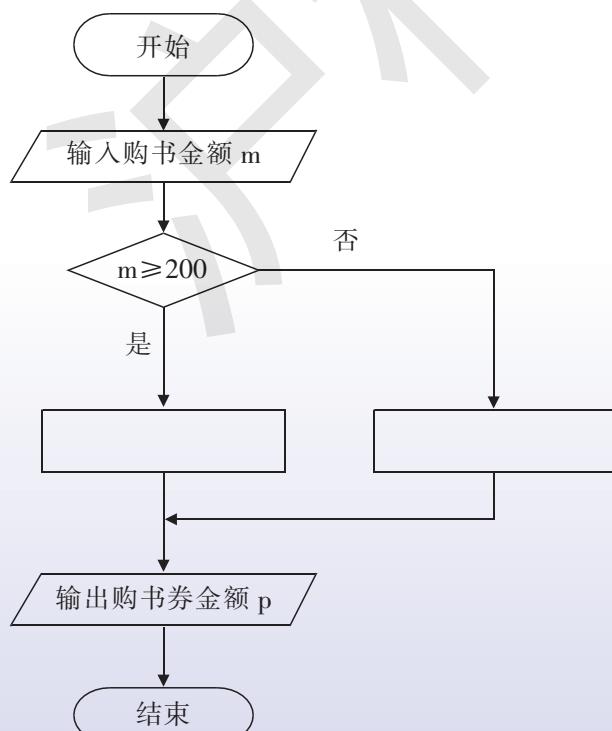
活动一 用 If 语句实现简单选择结构

问题:某书城在进行促销活动,凡购书满 200 元,可以去服务台领取相当于购书金额 10% 的购书券,供下次购书使用,购书券最小面值 1 元。现需要为服务台的工作人员编写一个程序,计算每次发放购书券的金额。

1. 分析问题。

设 m 表示某人的购书金额, p 表示需要发放的购书券金额。求解的条件是如果 _____, 则 _____; 否则 _____。

2. 设计算法,依据问题分析,将流程图补充完整。



3. 实现应用程序。

(1) 新建工程, 参照图 2-29, 创建窗体。

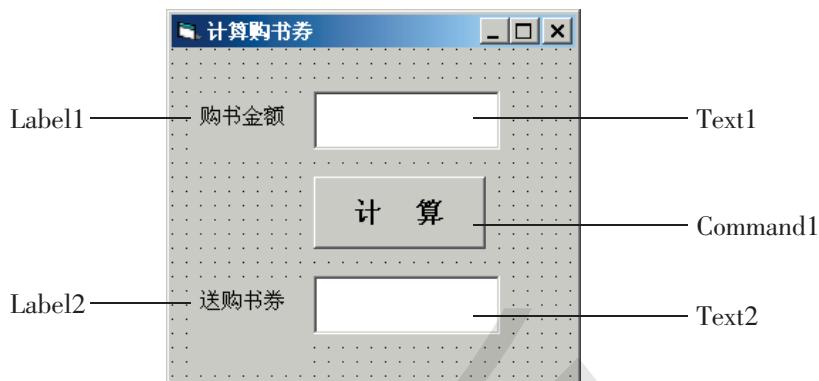


图 2-29 制作窗体

If语句的标准格式如下:

If 条件 Then

语句块1

[Else

语句块2]

End If

当条件成立时, 执行Then分支下的语句块1; 当条件不成立时, 执行Else分支下的语句块2。

本代码中的“ $m \geq 200$ ”是一个关系表达式, 用于对条件进行判断; “ \geq ”是一个关系运算符, 相当于数学中的“ \geq ”。
详见学习指引。

(2) 编写程序代码。

```
Private Sub Command1_Click()
    Dim m As _____
    Dim p As _____
    m=Val(Text1.Text)
    If m >=200 Then
        p= _____
    Else
        p= _____
    End If
End Sub
```

(3) 打开代码窗口, 输入代码。

4. 运行、调试程序, 如图 2-30 所示。

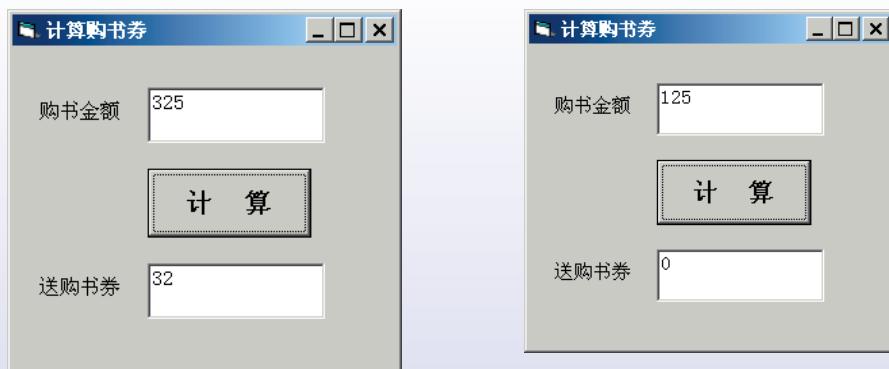


图 2-30 程序运行界面

活动二 用 If 语句的嵌套实现多重选择结构

问题：书城的促销活动方案进行了如下修改：购书未满 200 元，赠送 5 元购书券；购书满 200 元但未满 500 元，赠送购书券金额为消费金额的 5%；购书满 500 元但未满 1000 元，赠送购书券金额为消费金额的 10%；购书满 1000 元，赠送购书券金额为消费金额的 15%。购书券最小面值 1 元。请根据新的促销方案修改活动一的程序。

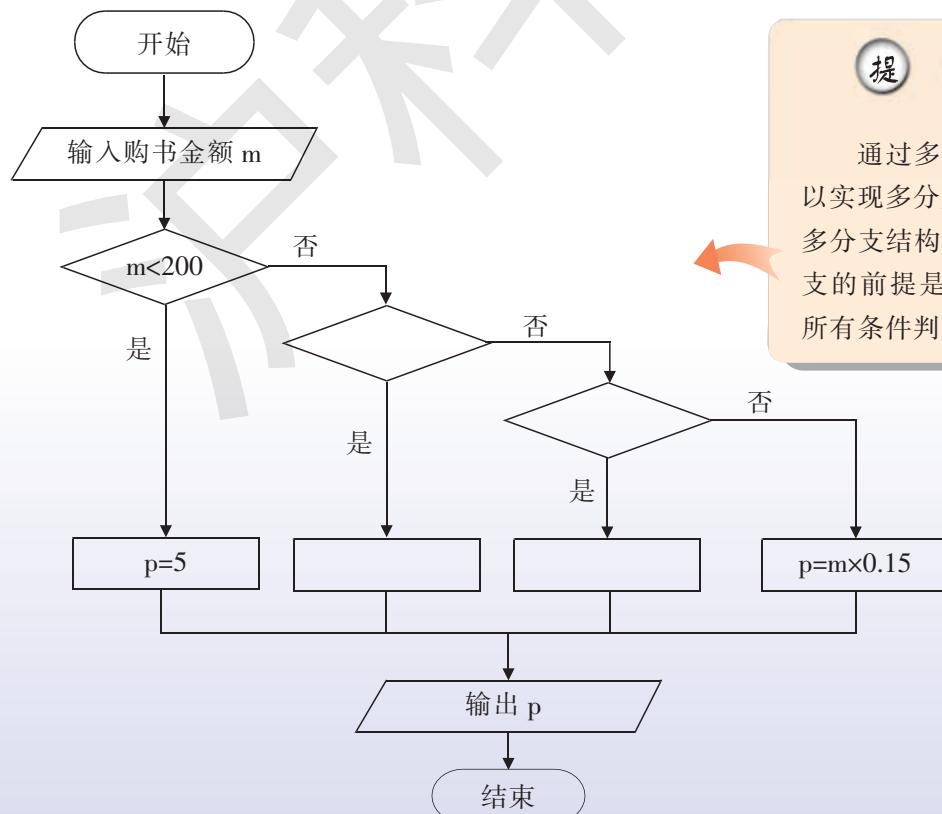
1. 分析问题。

与活动一的问题相比，问题的求解条件和相应的处理更加复杂，写出不同的求解条件和相应的处理，完成下表。

	消费金额范围	购书券金额计算
情况 1	$m < 200$	$p = 5$
情况 2		
情况 3		
情况 4	$M \geq 1000$	$p = m \times 0.15$

本活动的关键是判断 _____ 的值属于哪一个范围，再进行相应的处理。

2. 设计算法，依据问题分析，将流程图补充完整。



提 示 板

通过多次的选择判断，可以实现多分支的选择结构。在多分支结构中，进入某一个分支的前提是满足在其之前的所有条件判断。

3. 修改活动一的程序代码。
4. 运行、调试程序, 观察运行结果。

购书金额达 150 元, 送购书券 _____;
 购书金额达 200 元, 送购书券 _____;
 购书金额达 355 元, 送购书券 _____;
 购书金额达 500 元, 送购书券 _____;
 购书金额达 783 元, 送购书券 _____;
 购书金额达 1000 元, 送购书券 _____;
 购书金额达 1748 元, 送购书券 _____。

活动三 使用多重选择语句实现活动二的程序

活动二的程序中使用了三个 If 语句的嵌套来处理四种情况, 如果再多几种情况, 程序的可读性和效率都会比较差。VB 中还提供了一种多重选择语句, 利用它也可以方便地实现上一程序。

1. 阅读程序代码, 理解该多重选择语句的含义。

提 示 板

在VB中, 设置了一个处理多重情况的语句, 即多重选择语句。多重选择语句的执行过程是: 当表达式的值满足某 Case 语句的表达式范围, 则执行该语句后的语句组。例如, 本程序中当 $m < 200$ 时, 执行 $p = 5$; 当 $200 \leq m \leq 499.99$ 时, 执行 $p = m * 0.05$ 。

```
Private Sub Command1_Click( )
  Dim m As Single, p As Integer
  m = Val(Text1.Text)
  Select Case m
    Case Is < 200
      p=5
    Case 200 to 499.99
      p=m*0.05
    Case 500 to 999.99
      p=m*0.1
    Case Is >=1000
      p=m*0.15
  End Select
  Text2.Text = p
End Sub
```

2. 运行、调试程序,如图 2-31 所示。



图 2-31 程序运行界面



讨论

与活动二的 If 语句的嵌套进行比较,分析 If 语句的嵌套与多重选择语句的适用情况。



学习指引

在解决一个较复杂实际问题时,往往需要根据某些条件作出判断,或是有条件地执行某一操作。计算机程序中的选择结构能让程序根据不同的情况,执行不同的程序段,得到不同的结果。最基本的选择结构流程图如图 2-32 所示,当判断条件成立的时候,执行语句块 1;否则就执行语句块 2。语句块 1 和语句块 2 均表示一个语句组,可以包含 0 条或多条语句。

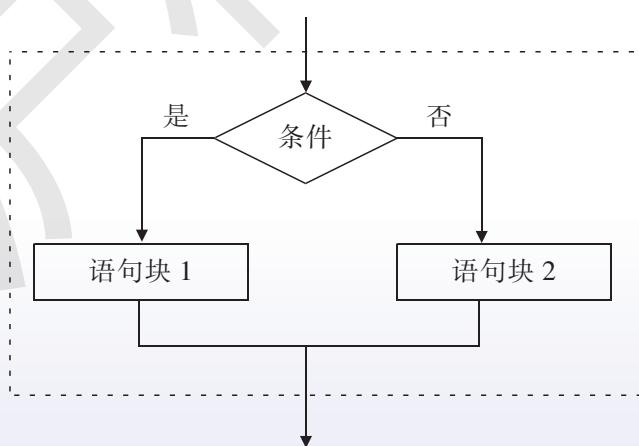


图 2-32 最基本的选择结构流程图

■ 控制条件

VB 支持的判断条件都是由关系运算符构成的关系表达式或由逻辑运算符构成的逻辑表达式。例如,在活动一的程序中,“ $m>=200$ ”就是一个关系表达式。

1. 关系运算符

关系运算符也称为比较运算符,它用于对两个数据的值进行比较,比较的结果是一个逻辑值。用关系运算符连接的表达式称为关系表达式。如果表达式成立,其逻辑值为真(True),如果表达式不成立,其逻辑值为假(False)。6种常用的关系运算符如下表所示。

关系运算符	含义	示例	结果
=	等于	"ABCDE"=="ABR"	False
>	大于	3>23	False
>=	大于等于	3>=23	False
<	小于	3<23	True
<=	小于等于	3<=3	True
<>或><	不等于	"abc"<>"ABC"	True

2. 逻辑运算符

在实际问题中,对一些复杂的条件,需要用几个关系表达式组合起来表示,这种组合操作就是逻辑运算,又称为布尔运算。常用的逻辑运算符包括与(And)、或(Or)、非(Not)。

对 And,只有当其两端连接的条件都成立的时候,逻辑运算结果才为 True;对 Or,只要其两端连接的条件中有任何一个成立,逻辑运算结果就为 True;Not 则是取反操作,若原条件为 True,逻辑运算结果就为 False,若原条件为 False,逻辑运算结果就为 True。

逻辑运算符的优先级关系如图 2-33 所示。

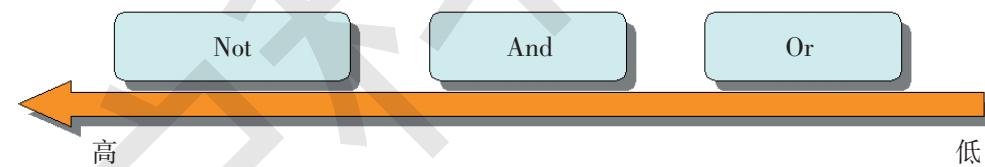


图 2-33 逻辑运算符的优先级

例如:

(1) 要表示 m 是大于 200 并且小于等于 500 的一个数,需要用 $m>200$ 和 $m<=500$ 这两个关系表达式通过与(And)运算来组合:

$(m>200) \text{ And } (m<=500)$

当 $m=800$ 时, $m>200$ 的值为 True, $m<=500$ 的值为 False, True And False 的值为 False,故运算结果为 False。

(2) 要表示 m 是大于 500 或者小于等于 200 的一个数,需要用 $m>500$ 和 $m<=200$ 这两个关系表达式通过或(Or)运算来组合:

$(m>500) \text{ Or } (m<=200)$

当 $m=100$ 时, $m>500$ 的值为 False, $m<=200$ 的值为 True, False Or True 的值为 True,故运算结果为 True。

(3) $a \text{ Or } b \text{ And } \text{Not } c$

此表达式首先执行 $\text{Not } c$, 再将其结果与 b 进行与运算, 最后将结果与 a 进行或运算, 其运算顺序如图 2-34。

当 $a=True, b=False, c=True$ 时, $\text{Not } c$ 的值为 False , $b \text{ And } \text{False}$ 的值为 False , $a \text{ Or } \text{False}$ 的值为 True , 此表达式的运算结果为 True 。

如表达式改为 $(a \text{ Or } b) \text{ And } \text{Not } c$, 则先进行括号中的运算, 最终运算结果为 False 。

3. 控制条件的表示

逻辑运算符和关系运算符常在控制语句中用于表示条件。例如, 要判断 c 是 a, b, c 中的最大值, 表达式可写为: $(c>b) \text{ And } (c>a)$ 。

在一个表达式中可以出现多种类型的运算符, 所以还需要规定不同类型运算符之间的优先关系。运算符的优先顺序如图 2-35 所示。

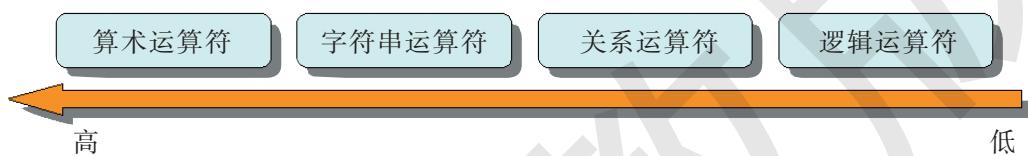


图 2-35 基本运算符的优先级

例如, 表达式 $a>b+5 \text{ And } a<=c-5$ 表示的是判断 a 是否既大于 $b+5$ 的值, 又小于等于 $c-5$ 的值。

又如, 表达式 $c+a>b+5 \text{ Or } a<c-5 \text{ And } c>0$ 的运算顺序如图 2-36。

当 $a=1, b=-8, c=3$ 时, 首先进行算术运算, 运算 $c+a$ (值为 4)、 $b+5$ (值为 -3)、 $c-5$ (值为 -2); 其次进行关系运算, 运算 $4>-3$ (值为 True)、 $a<-2$ (值为 False)、 $c>0$ (值为 True); 然后进行逻辑运算, 根据优先级, 先运算 False And True (值为 False), 再运算 True Or False (值为 True), 最后得出运算结果 True 。

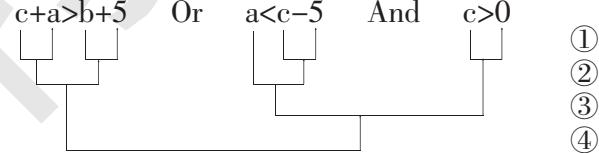


图 2-36 运算顺序示例

■ 实现选择结构的语句

1. If 语句

If 语句是几乎所有的高级编程语言中普遍使用的选择结构语句, 程序可以通过 If 语句来完成对条件的判断、选择。

(1) 简单 If 语句。

格式:

If 条件 Then

语句块

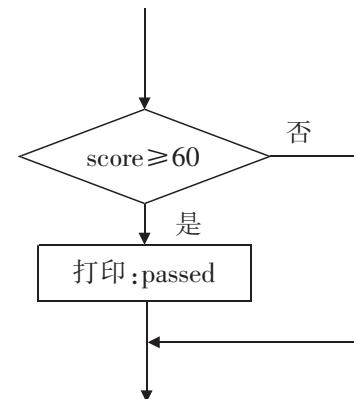
End If

简单 If 语句只包含一个分支, 如果条件成立, 则执行 Then 下面的语句块; 如果条件不成立, 就什么都不执行。

例如,考试成绩高于 60 分表示通过,流程图如图 2-37 所示,当条件成立时,执行输出语句,打印字符串“passed”;当条件不成立时,不做任何操作。

实现这一算法的代码为:

```
If score>=60 Then
    Print "passed"
End If
```



(2) If/Then/Else 语句。

格式:

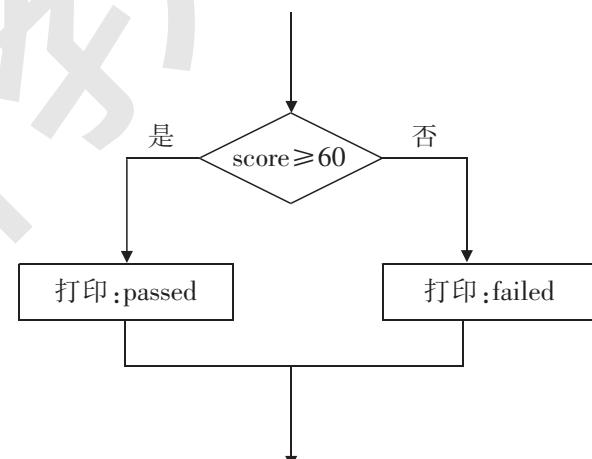
```
If 条件 Then
    语句块 1
Else
    语句块 2
End If
```

If/Then/Else 语句包含两个分支,当条件成立时,执行 Then 分支后面的语句块 1;当条件不成立时,执行 Else 分支后面的语句块 2。

使用 If/Then/Else 语句,可以根据条件的成立和不成立指定不同的操作。例如对上一例题进行补充,要求在成绩低于 60 分的时候打印“failed”,其流程图如图 2-38 所示。

实现这一算法的代码为:

```
If score>=60 Then
    Print "passed"
Else
    Print "failed"
End If
```



(3) If 语句的嵌套。

当程序处理面临多项选择的时候,可以通过 If 语句的嵌套实现多分支结构。If 语句的嵌套是指在一个 If 语句中又包含一个或多个 If 语句,其一般形式如下:

```
If 条件 Then
    If 条件 Then
        语句块 1
    Else
        语句块 2
    End If
}
```

内嵌 If 语句

```

Else
  If 条件 Then
    语句块 3
  Else
    语句块 4
  End If
End If

```

例如将百分制的分数转化为等级制,流程图如图 2-39 所示。

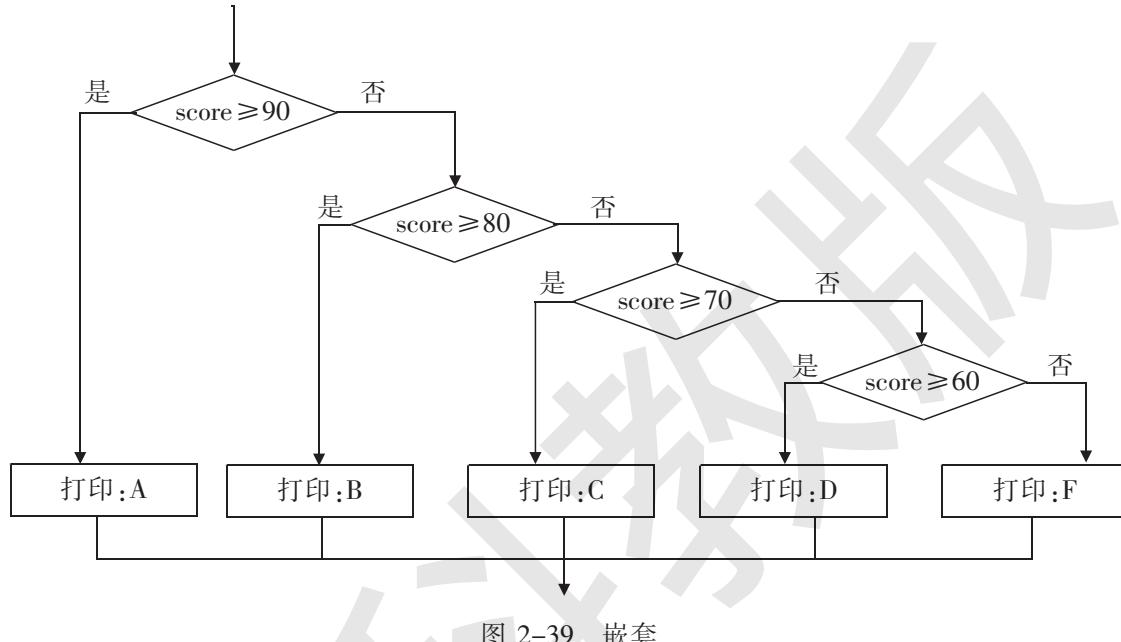


图 2-39 嵌套

实现这一算法的代码为:

```

If score>=90 Then
  Print "A"
Else
  If score>=80 Then
    Print "B"
  Else
    If score>=70 Then
      Print "C"
    Else
      If score>=60 Then
        Print "D"
      Else
        Print "F"
      End If
    End If
  End If
End If

```

2. Select Case 语句

虽然使用 If 语句的嵌套可以实现多分支结构,但当嵌套结构较多时,容易使程序变得混乱和难以调试,程序的运行效率也会比较差。一般高级编程语言都提供了多重选择语句,能够更方便地实现多分支结构。VB 提供的多重选择语句是 Select Case。

Select Case 语句的格式为:

```
Select Case 表达式
  Case 表达式范围 1
    语句块 1
  Case 表达式范围 2
    语句块 2
  .....
  Case 表达式范围 n
    语句块 n
  [Case Else
    语句块 n+1]
End Select
```

其中,表达式范围可以是以下几种:

- (1) 一个常量值,如 1。
- (2) 用逗号分隔的值序列,如 1,2,3,4,5,6,7。
- (3) 值的一段范围,例如 1 to 100 或 "A" to "Z",分别表示表达式的值在 1 到 100 或 "A" 到 "Z" 之间。
- (4) Is 关系表达式,如 Is >45 或 Is <=20。

使用 Select Case 语句实现将百分制的分数转化为等级制,比 If 语句要清晰得多,用代码表示为:

```
Select Case(score)
  Case 90 to 100
    Print "A"
  Case 80 to 89
    Print "B"
  Case 70 to 79
    Print "C"
  Case 60 to 69
    Print "D"
  Case 0 to 59
    Print "F"
  Case Else
    Print "Input error!"
End Select
```



体验活动

- 火车站寄存行李的收费标准是 50 千克以内(包括 50 千克)0.20 元/千克,超过部分的收费标准为 0.50 元/千克。编写程序,根据输入的任意重量,计算出应付的行李费。
- 根据以下的算法写出程序代码,并指出其功能。

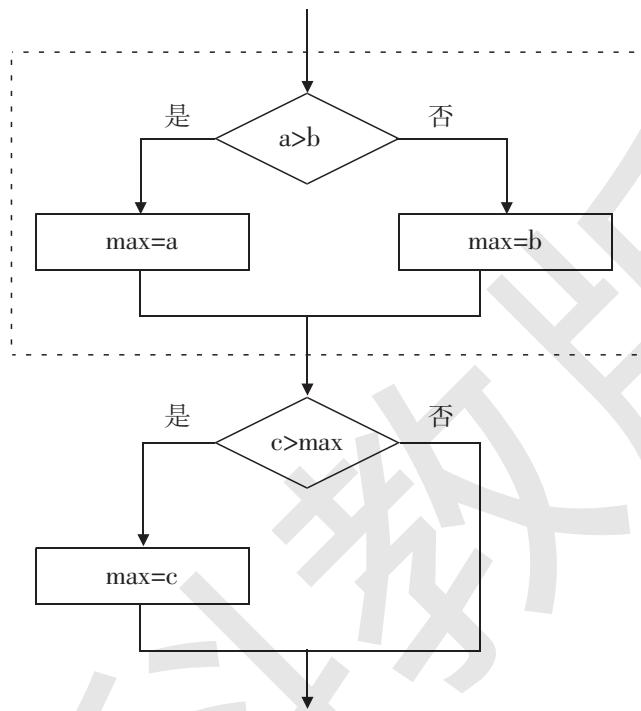


图 2-40 流程图

- 给定三角形的三条边长,计算三角形的面积。

提示:首先判断给出的三条边能否构成三角形(三角形的两边之和大于第三边),如可以,则计算并输出该三角形的面积;否则输出“不能构成三角形”。设三角形三边为

$$a, b, c, q = \frac{a+b+c}{2}, \text{则面积 } S = \sqrt{q \cdot (q-a) \cdot (q-b) \cdot (q-c)}.$$

- 编写一个数字到星期名称的转换程序,当输入 1 时,输出“星期一”;当输入 2 时,输出“星期二”;依此类推。如果输入 1 到 7 之外的数字,输出“输入错误”。

- 编写一个判断某年是否为闰年的程序。

提示:判断某年是否为闰年的方法是如果该年份的数值能被 4 整除但不能被 100 整除,或者能被 400 整除则该年份是闰年。

- 求某人的实足年龄,要能输入出生年月日后,输出实足年龄。求解实足年龄的方法是用当前年份减去其出生年份,并判断生日是否已过,如果尚未过生日,实足年龄还要减一。

提示:判断生日是否已过,首先应进行月份的比较,若月份相同,还需进行日的比较。需要用到的系统函数有 Date()、Year()、Month()、Day(),Date()表示由计算机提供的系统日期,Year()、Month()、Day()分别用来提取日期中的年、月、日。

第四节 循环结构

在日常生活中有许多需要重复进行的工作。例如,考试结束后,教师要进行阅卷工作,如果全班共有50个学生,则教师需要批阅50份试卷,工作十分繁重。出现标准化试题后,学生用2B铅笔将答案涂写在答题纸上,该部分的阅卷工作就可以由计算机来完成了。计算机进行阅卷工作,其过程是:

先批阅第1个学生的试卷,给出成绩;

如果没有批阅到第50位学生,

则继续批阅下一个学生的试卷,给出成绩;

如果批阅完50个学生的试卷,则结束工作。

用计算机完成阅卷工作,可以采用循环结构的程序。在阅卷工作中,“批阅一份试卷并给出成绩”是需要计算机重复进行的工作。

在VB中,有哪些语句可用以实现循环结构呢?

活动一 用For...Next语句实现循环结构

问题:周末,小明到爸爸的电脑城去帮忙。爸爸正忙着进行月底清点。爸爸所在的品牌电脑部经营着不同品牌和型号的35种电脑。他希望小明能编写一个程序,帮助计算每月电脑的销售总额。

1. 分析问题。

通常,本问题可用连加的方法求解,即月销售总额由各品牌和型号电脑的月销售额相加得到。

设 s 为电脑的月销售总额, x_i 为某种电脑的月销售额, $i=1, 2, 3, \dots, 35$, 则可列计算式: _____。

若改为采用累加的方法,设 $s_0=0$, x_i 为某种电脑的月销售额, $i=1, 2, \dots, 35$, 则

$$s_1=s_0+x_1$$

$$s_2=s_1+x_2$$

...

$$s_{35}=s_{34}+x_{35}$$

那么,解决本问题,需重复进行的工作是: _____; 重复计算的次数为: _____。

2. 设计算法。

(1) 阅读小明设计的算法。

提 示 板

需求解的问题是一个繁复的问题时,若能发现需重复进行的工作,就可采用循环结构的程序解决。此例是找到可重复进行计算的算式。

第1步：令 $s = 0$ ；
 第2步：令 $i = 1$ ；
 第3步：如果 $i \leq 35$ ，则执行第4步，否则执行第7步；
 第4步：输入第 i 种电脑的月销售额到 x ；
 第5步： $s = s + x$ ；
 第6步： $i = i + 1$ ，转第3步执行；
 第7步：结束。

(2) 分析算法。

- ① 哪几步在重复执行？
- ② 循环开始时 i 的值为多少？每执行一次循环操作， i 的值怎样变化？循环结束时 i 的值为多少？变量 i 的作用是什么？
- ③ 统计销售额的工作是由哪几步完成的？

(3) 图2-41为循环结构的标准流程图，参照此图，试将本活动算法用流程图描述出来。

提 示 板

构建一个循环结构，首先要分析需要重复执行的操作，提炼出循环操作内容，然后要确定如何控制循环。

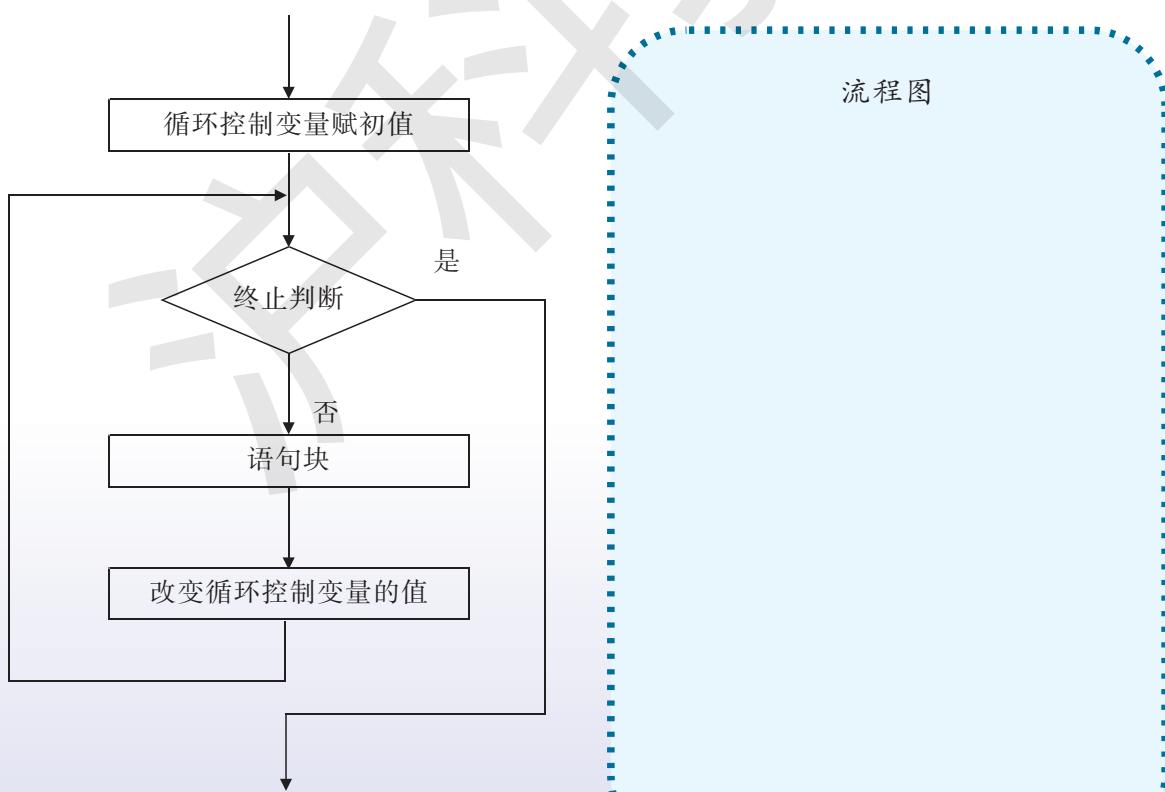


图 2-41 循环结构标准流程图

3. 实现应用程序。

- (1) 新建一个工程。
- (2) 完成以下代码，并将其输入代码窗口。

提 示 板

程序中第6~9句是循环结构语句，语句格式为：
 For 循环控制变量=初值
 To 终值 [Step 步长]
 循环体语句块
 Next [循环控制变量]

```
Private Sub Form_Click()
```

```
  Dim s As Single
  Dim x As Single
  Dim i As Integer
  s = 0
  For i = 1 To 35 Step 1
    x = Val(InputBox("输入第" & i & "种电脑的月销售额"))
    s = _____
  Next i
  Print "月销售总额为"; Format(s, "#,###.00")
End Sub
```

(3) 调试运行程序。



图 2-42 监视变量

(4) 依次保存窗体文件、工程文件为sum.frm和sum.vbp。

活动二 在循环结构中数组的应用

问题：计算完本月电脑的销售总额，小明的爸爸还希望能用先前输入的35种电脑的月销售额作些其他统计分析工作。但小明发现活动一的程序中输入的数据并没有被保存下来。于是，他决定修改程序，即在计算每月电脑的销售总额的同时能保存35种电脑的月销售额。

1. 分析问题。

活动一的程序在统计了月销售总额后没有保留下各品种电脑的月销售额数据,是因为程序采用同一个变量来存放这些输入的数据,当这些数据参与了累加计算后,又被下一个品种的相应数据覆盖了。

若欲保留这些输入数据,可以使用一种称为“数组”的数据结构。例如,可用数组x(35)来保存这35种电脑的月销售额,其中x(1)表示第1种电脑的月销售额,x(2)表示第2种电脑的月销售额,……,x(35)表示第35种电脑的月销售额。

2. 设计算法。

用数组替代变量完成计算月销售总额的算法。

第1步:令s=_____;

第2步:令i=1;

第3步:如果*i*≤35,则执行第4步,否则执行第7步;

第4步:输入x(*i*)的值;

第5步:s=s+_____;

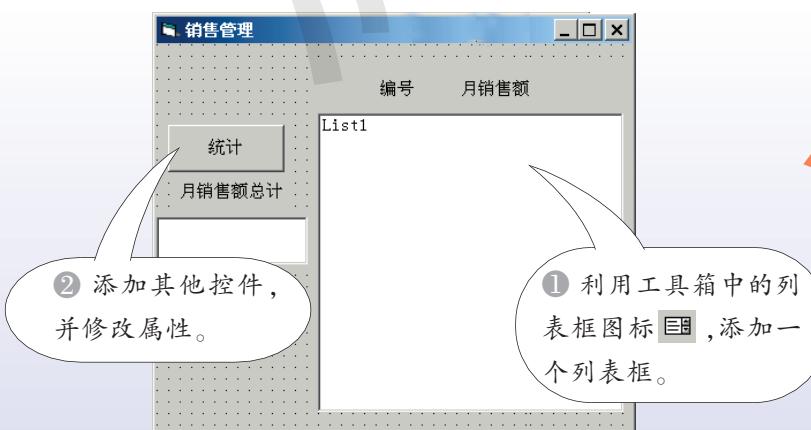
第6步:*i*=*i*+1,转第3步执行;

第7步:结束。

3. 实现应用程序。

(1) 按以上算法,用For...Next语句写出循环部分的程序代码。

(2) 新建工程,参照图2-43设计窗体。



提 示 板

数组是一组相同类型变量的集合。

在VB中,数组a(4)表示一个含有5个数组元素的数组,各元素分别为a(0)、a(1)、…、a(4)。详见学习指引。

提 示 板

列表框是一行一行显示数据列表的方框。本程序中可用列表框来输出电脑的编号和月销售额。

图 2-43 制作窗体

(3) 完成以下代码，并将其添加到代码窗口中。

提 示 板

向列表框中添加一行数据的方法是AddItem。如语句List1.AddItem j的作用是向列表框中添加变量j的值。清空列表框的方法是Clear，如语句List1.Clear。

```

Private Sub Command1_Click()
    Dim s As Single
    Dim i As Integer
    Dim x(35) As Single
    s = 0
    List1.Clear
    For i = 1 To 35 Step 1
        _____ = Val(InputBox("输入第" & i & "种电脑的月销售额"))
        List1.AddItem i & vbTab & x(i)
    _____
    Next i
    Text1.Text = _____
End Sub

```

4. 调试、运行程序。

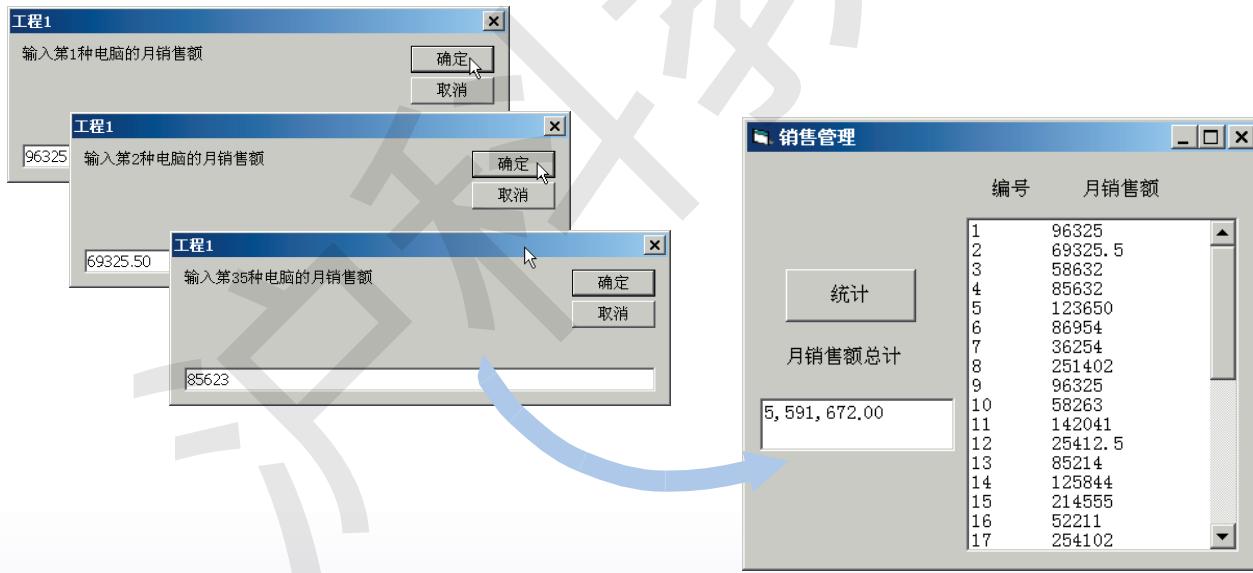


图 2-44 程序运行界面

活动三 在循环结构中深入应用数组

问题：小明的爸爸希望在程序中增加一些功能，如找出某月销售额最高的电脑的编号及月销售额。小明在帮助解决该问题时，决定在活动二的窗体的基础上再添加一个命令按钮来实现该项功能。

1. 分析问题。

依据以下的问题分析报告,找出求解过程中出现的重复操作内容及控制循环的方法。



问题分析

找出某月销售额最高的电脑,可转化为找出数组 $x(35)$ 中的最大值,并记下该数组元素的下标。可以设一个变量 $maxj$ 来记录最大数组元素的下标,将其初值设为 1,然后将 $x(maxj)$ 与数组 $x(35)$ 中的元素逐一进行比较,如果某一数组元素 $x(i)$ 比 $x(maxj)$ 大,就将其下标 i 赋给 $maxj$,再将 $x(maxj)$ 与下一个数组元素进行比较,……,直至比较结束,变量 $maxj$ 的值就是所找到的最大数组元素的下标, $x(maxj)$ 即为求解的最大值。

由以上分析可知:循环控制变量 i 的初值为:_____,终值为:_____,步长为:_____。

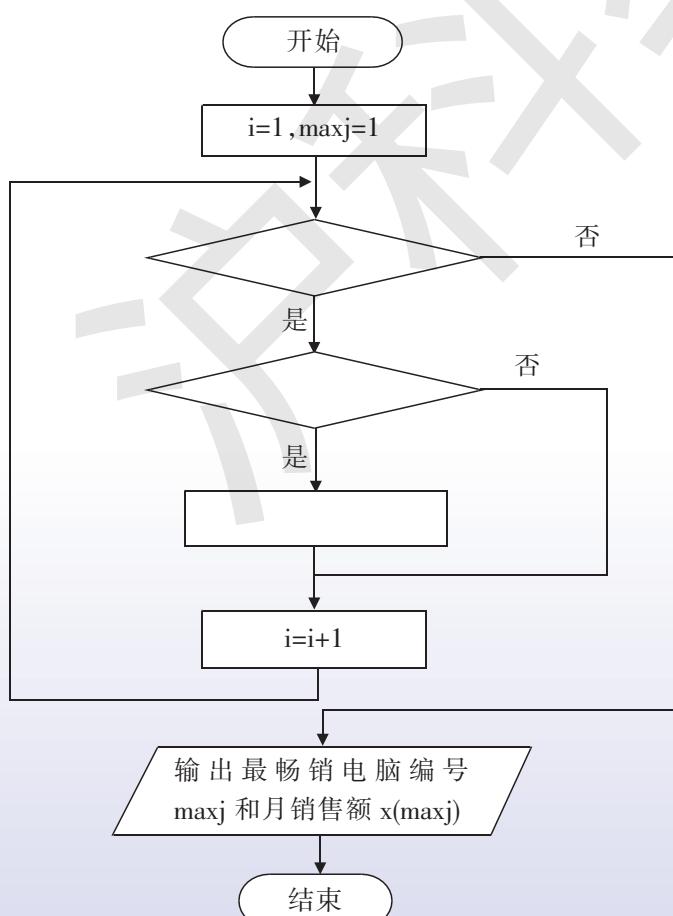
循环体中的操作为:如果_____那么_____。

2. 设计算法。

根据问题分析,完成流程图。

提 示 板

根据以上的问题分析,求解过程中的重复操作为比较两数,并根据比较结果判断是否修改 $maxj$ 的值。



3. 实现应用程序。

(1) 设计窗体。依图 2-45 在活动二创建的窗体上添加控件,完成窗体。



图 2-45 制作窗体

提 示 板

数组x(35)的定义位置与前述不同,是被写在了代码窗口的最上方,而不是Command1_Click()过程中。这样该数组还可以在其他的事件过程中被引用。定义在代码窗口最上部且不在任何过程内的变量称为窗体级变量。窗体级变量可供该窗体中所有的事件过程使用。详见第3章。

(2) 将原 Command1_Click()过程中的数组定义语句移至代码窗口的顶部位置。

(3) 编写代码,输入到 Command2_Click()过程中。

```
Private Sub Command2_Click()
    Dim i As Integer
    Dim maxj As Integer
    maxj = 1
    For i = 1 To 35
        If _____ Then
            maxj = i
        End If
    Next i
    Text2.Text = "编号:" & maxj & vbTab & "月销售额:"
    & Format(x(maxj), "#,###.00")
End Sub
```

4. 调试、运行程序。

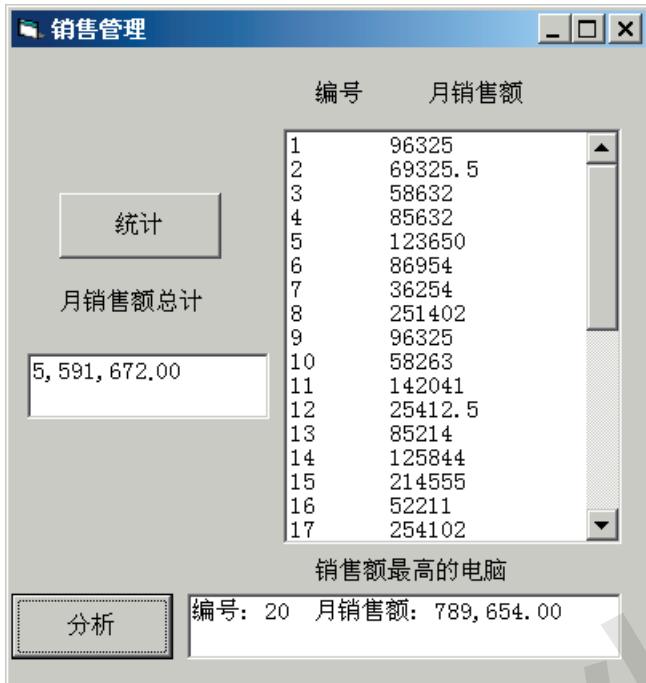


图 2-46 程序运行界面

活动四 用Do...Loop语句实现循环结构

问题:小明的爸爸决定对某种电脑进行促销。促销方案为:买第一台时需付全价 6400 元,买第二台时只需付全价的 95%,依次类推,买后一台的价格是前一台的 95%,但最低价不得低于 3800 元,如果低于 3800 元就按 3800 元的价格购买。有一位顾客需为单位购置电脑,他计划购买电脑的费用是 50000 元,求该顾客最多能买几台电脑,需付多少钱。

1. 问题分析。

本问题的解决思路是:一、每买一台电脑,需要计算这台电脑的价格,然后累加到总金额上,当总金额超过 50000 元时,就停止循环。因此,本循环过程中的重复操作是计算电脑的单价及总金额。二、在计算电脑的单价时,还需要作一个判断:如果打折后的价格大于 3800 元,那么在前一次价格的基础上打折,折扣率为 95%,否则价格即为 3800 元,不再打折,折扣率可看作为 100%。

设电脑的价格为 p ,折扣率为 m ,购买电脑的台数为 n ,购买电脑的总金额为 s 。

① 折扣率 m 的值需要根据前一台电脑的价格 p 来确定。如果 $p > 3800$,那么 $m = \underline{\hspace{2cm}}$;否则 $\underline{\hspace{2cm}}$ 。

提 示 板

前面所解决的问题中循环次数都是确定的,但本活动中循环多少次事先并不知道,而是通过判断购买电脑的总金额是否超过 50000 元来控制循环。

② 根据促销方案,购买某台电脑的价格是在前一台的价格上再打折,可采用累乘的方式计算某台电脑的价格。计算公式为 $p=p\times\text{_____}$ 。

③ 采用累加的方式,购买电脑的总金额的计算公式为 $s=s+\text{_____}$ 。

2. 算法设计。

(1) 阅读以下算法。

算法 1

第 1 步:令 $p = 6400$;
 第 2 步:令 $n = 1$;
 第 3 步:令 $m = 0.95$;
 第 4 步: $s = p$;
 第 5 步:如果 $s \leq 50000$,则执行第 6 步,否则执行第 10 步;
 第 6 步: $p = p \times m$;
 第 7 步:如果 $p < 3800$,则 $p = 3800, m = 1$;
 第 8 步: $s = s+p$;
 第 9 步: $n = n+1$,转第 5 步执行;
 第 10 步:输出 n 和 s ;
 第 11 步:结束。

算法 2

第 1 步:令 $p = 6400$;
 第 2 步:令 $n = 1$;
 第 3 步:令 $m = 0.95$;
 第 4 步: $s = p$;
 第 5 步:如果 $s \leq 50000$,则执行第 6 步,否则执行第 10 步;
 第 6 步: $p = p \times m$;
 第 7 步:如果 $p < 3800$,则 $p = 3800, m = 1$;
 第 8 步: $s = s+p$;
 第 9 步: $n = n+1$,转第 5 步执行;
 第 10 步: $n = n-1$;
 第 11 步: $s = s-p$;
 第 12 步:输出 n 和 s ;
 第 13 步:结束。

提 示 板

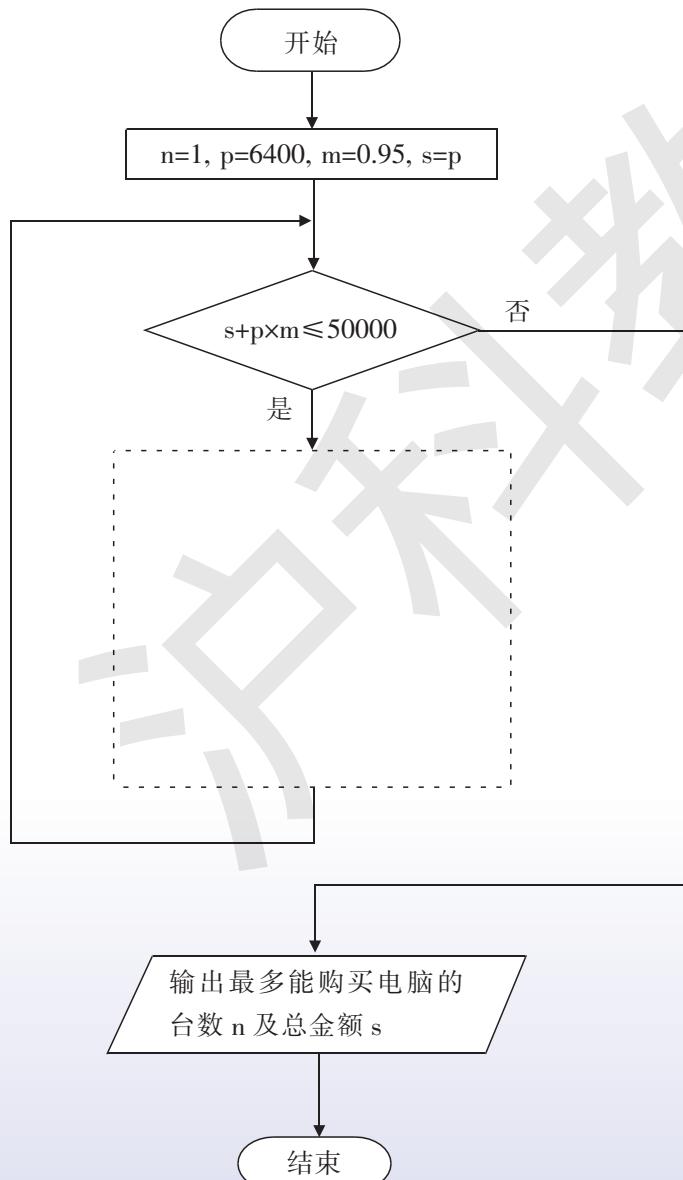
分析循环结构时,尤其要注意最后一次循环的执行情况。如在本活动中要注意最后一次循环开始前后 s 值的变化,判断退出循环时 s 是否小于等于 50000。



讨论

- (1) 两个算法求出的s是大于50000,还是小于等于50000?
哪个算法的输出结果正确? 为什么?
- (2) 如果将算法1第4步中控制循环的条件 $s \leq 50000$ 改为 $s+p \times m \leq 50000$,该算法的运行结果正确吗? 为什么?
- (3) 能否将算法1或算法2中的第6步和第7步互换位置?
为什么?

- (2) 完成流程图。



3. 实现应用程序。

(1) 新建工程。

(2) 阅读以下代码, 并将其添加到代码窗口中。

提 示 板

本程序中, 每次执行Do While语句, 首先判断条件 $s + p * m \leq 50000$, 若条件成立, 则执行后面的语句, 遇到Loop语句返回到Do While再次进行条件判断; 若条件不成立, 则跳转到Loop语句后结束循环。

```
Private Sub Form_Click( )
    Dim s As Single, m As Single, p As Single
    Dim n As Integer
    n = 1
    p = 6400
    m = 0.95
    s = p
    Do While s + p * m <= 50000
        p = p * m
        If p < 3800 Then
            p = 3800
            m = 1
        End If
        s = s + p
        n = n + 1
    Loop
    Print "最多可购买"; n; "台电脑, "; "需付"; Format(s, "#,###.
    00"); "元钱。"
End Sub
```

讨 论

- (1) 本程序中控制循环的表达式是什么? 程序中通过哪些语句来改变该表达式的值?
- (2) 循环结束后 s 的值是否会大于 50000?
- (3) 若按照算法2编写程序, 需要怎样修改以上代码?

4. 调试、运行程序。

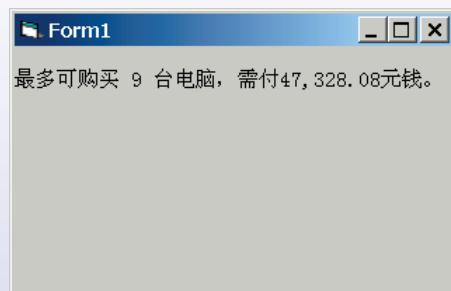


图 2-47 运行结果



学习指引

计算机最擅长的工作就是重复操作。在编程中采用实现循环结构的语句,可以有规律地反复执行某程序段,方便地解决重复操作问题。

■ 循环结构

要构建一个循环结构,需要关注两个方面的问题:一是重复要做的是什么事,二是如何控制循环。

重复要做的是什么事,可以从问题分析入手解决。而对循环操作的控制,一般是通过使用一个循环控制变量来进行。循环控制的三要素是:

- ① 循环控制变量的初值;
- ② 循环控制变量的终值;
- ③ 在循环体中具有使循环控制变量由初值向终值变化的语句。

先考虑一个简单的例子。

例 1:求 50!。

问题分析

$50! = 1 \times 2 \times 3 \times \dots \times 50$,要做多次乘法。

算法设计①

第 1 步:令 $s=1$;

第 2 步: $s=s \times 1$;

第 3 步: $s=s \times 2$;

...

第 51 步: $s=s \times 50$ 。

观察上面的算法,可以发现每一步的操作都在做乘法,所不同的是累乘的数据不同,但是这些数据的变化是有规律的,可以得到通项公式: $s=s \times i$ 。

接下来就要确定循环控制变量。求阶乘的循环控制是由通项公式的执行次数决定的,一共执行 50 次,此时循环控制变量相当于一个计数器,从 1 开始,到 50 结束,每一次递增 1。循环控制变量的变化和通项公式中的变量 i 的变化规律是一样的,所以在本例中就把 i 当作循环控制变量。

算法设计②

第 1 步:令 $s=1$;

第 2 步:令 $i=1$;

第 3 步:如果 $i \leq 50$,则执行第 4 步,否则执行第 6 步;

第 4 步: $s=s \times i$;

第 5 步: $i=i+1$,转第 3 步执行;

第 6 步:结束。

流程图如图 2-48 所示。

值得注意的是，在循环体中一定有一句使该循环能趋于结束的语句，如这里的 $i=i+1$ ，这样循环控制变量才能从初值变化到终值，结束循环。

在可采用循环结构的程序解决的问题中，也会遇到一些循环执行次数不确定的问题。

例 2：由用户输入若干个数，求平均值，当输入的值为-1 的时候循环终止。

问题分析

循环控制变量是每次用户输入的值，设为 x 。首先执行一次输入操作，获得 x 的初值，如果 x 不等于-1，则累加 x ；再执行输入操作改变 x 的值，再次判断 x 的值，如果不等于-1 则继续累加 x ；一直继续下去，直到输入的值等于-1，结束循环，并计算出所有输入的数的平均值。

算法设计

第 1 步：令 $s=0, n=0$ ；

第 2 步：输入一个值到 x ；

第 3 步：如果 $x \neq -1$ ，则执行第 4 步，否则执行第 7 步；

第 4 步： $s=s+x$ ；

第 5 步： $n=n+1$ ；

第 6 步：输入一个值到 x ，转第 3 步执行；

第 7 步：求平均值 s/n ；

第 8 步：结束。

在这个例子中，程序没有运行前是不知道循环次数的，由 x 的值是否为-1 来决定循环是否停止。 x 值的变化是没有规律的，通过在循环体中执行一个输入操作来改变 x 的值， x 相当于一个标志量。

■ 实现循环结构的语句

VB 提供了多种实现循环结构的语句，主要有 Do...Loop 语句和 For...Next 语句。

1. Do...Loop 语句

Do...Loop 语句有两种语法形式。

形式 1：Do While/Until...Loop

格式：

Do While/Until 条件

语句块

Loop

Do While...Loop 语句的执行顺序如图 2-49(a)所示，先判断条件 P1 是否成立，如果成立就执行语句块 1，执行完后再次判断条件 P1 是否成立，如果成立继续执行语句块 1，如果不

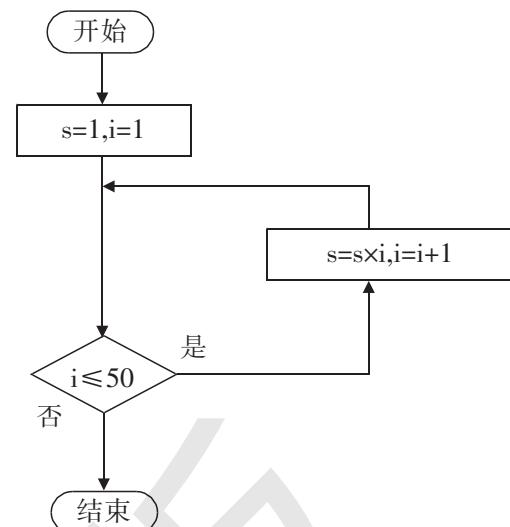


图 2-48 流程图

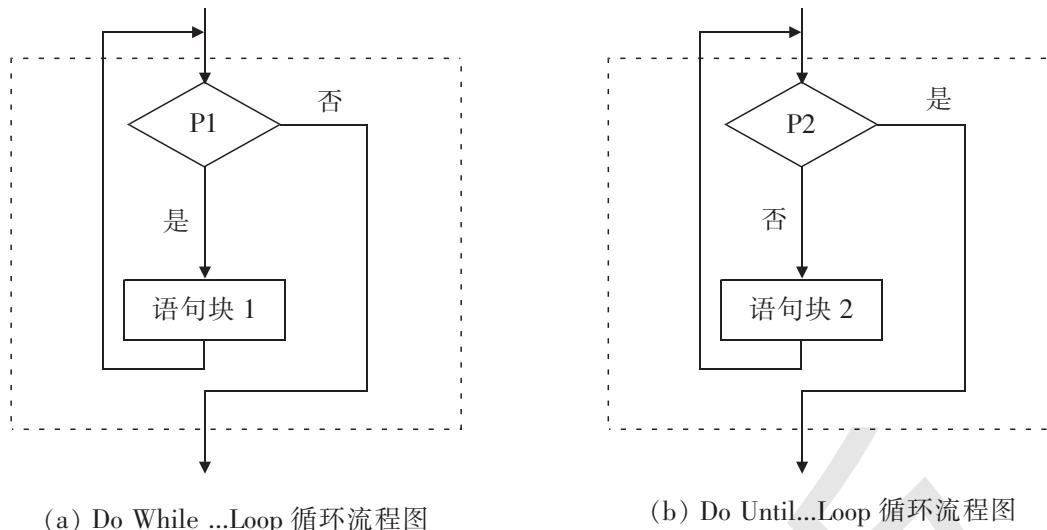


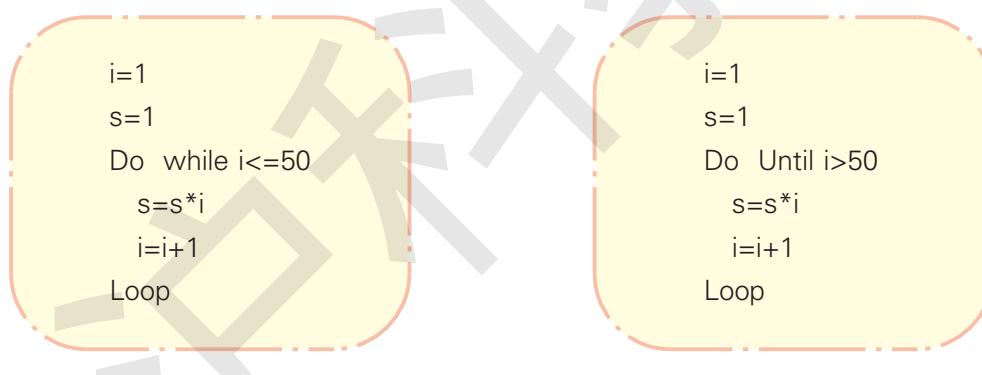
图 2-49

成立,就脱离循环体。

Do Until...Loop 语句的执行顺序如图 2-49(b)所示,它与 Do While...Loop 语句的执行顺序非常相似,区别在于判断条件 P2 不成立时执行语句块 2,判断条件 P2 成立就脱离循环体。

条件 P1 与条件 P2 的逻辑正好相反,P1 是满足循环执行的条件,P2 是终止循环执行的条件。

使用 Do While/Until...Loop 语句求解例 1 的代码分别为:



Do While...Loop 语句首先判断循环条件 $i \leq 50$ 是否成立,成立才能进入循环体,执行完循环体的两条语句后,再次进行循环条件的判断,直到 $i \leq 50$ 不成立为止。Do Until...Loop 语句的执行顺序与 Do While...Loop 语句的执行顺序相同,只是判断条件是 $i > 50$,与 Do While...Loop 语句的判断条件 $i \leq 50$ 正好相反。

形式 2:Do...Loop While/Until

格式:

```
Do
  语句块
Loop While/Until 条件
```

Do...Loop While/Until 语句的执行顺序如图 2-50 所示,与 Do While/Until...Loop 语句的区别在于循环体的第一次执行。前者第一次不需要判断,直接进入循环体,而后者先要进行

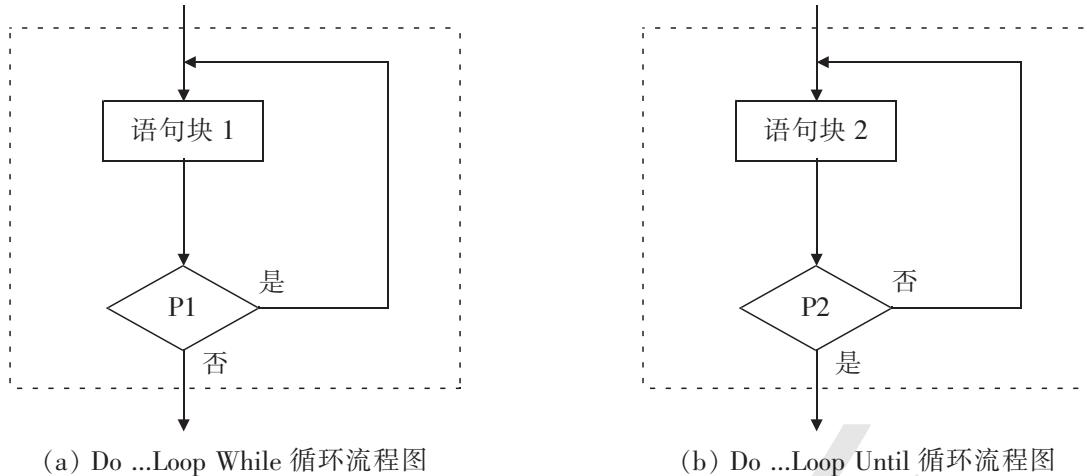


图 2-50

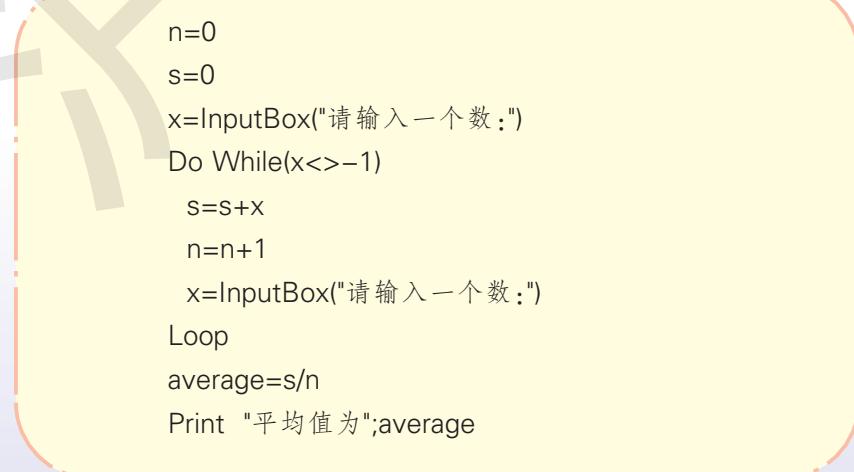
条件判断,才能决定是否能够进入循环体。

使用 Do...Loop While/Until 语句求解例 1 的代码分别为:



事实上 Do...Loop 语句也能够方便地实现循环次数不确定的循环结构。

例如,使用 Do While...Loop 语句求解例 2 的代码:



除了由循环控制变量的值来判断是否跳出循环体外,还可以使用 Exit Do 语句跳出 Do...Loop 语句循环。

例3:判断一个数是不是素数。

问题分析

如果一个数是素数,那么这个数不能被从2开始到这个数的平方根的整数部分中的任何数整除。所以判断一个数是不是素数,就需要验证这个数能不能被这些整数整除。只要它能被其中的一个数整除,则这个数就不是素数,不需要再继续验证;否则这个数就是素数。设Number是需要验证的数。

算法设计

第1步:令i=2;

第2步:令m=√Number

第3步:如果i<m,则执行第4步,否则执行第6步;

第4步:如果Number能被i整除,则执行第6步;

第5步:i=i+1,转第3步执行;

第6步:如果i<m,则Number不是一个素数,否则Number是一个素数;

第7步:结束。

● 使用Do While...Loop语句求解例3的代码:

```
i=2
m=Sqr(Number)
Do While i<=m
    If(Number mod i=0) Then
        Exit Do
    End If
    i=i+1
Loop
If i<m Then
    Print Number; "不是一个素数"
Else
    Print Number; "是一个素数"
End If
```

循环的控制包括两个部分:一是循环控制变量i的值从2变化到 $\sqrt{\text{Number}}$,i每次增加1;二是在循环体中使用条件语句判断Number能否被i整除,如果能被i整除,则使用Exit Do直接跳出循环体。

程序在两种状态下都会跳出循环,而这两种状态的含义是不一样的,当i的值大于 $\sqrt{\text{Number}}$ 时跳出循环,表示Number是一个素数;当执行Exit Do语句跳出循环,则表示Number不是一个素数,所以紧接着循环结构使用一个If/Else/End If语句加以区分。

2. For...Next语句

大多数高级程序语言还提供了For语句来实现循环结构。VB提供的For...Next语句用

于实现循环次数确定的循环结构,此时循环控制变量相当于一个计数器。For...Next 语句将循环控制的三要素集中在一行,结构更紧凑,也更易于理解。

语句的格式如下:

For 循环控制变量=初值 To 终值[Step 步长]

循环体语句块

Next[循环控制变量]

说明: ① 循环控制变量必须为整型;

② 步长一般为正,初值小于终值,缺省步长为 1;若步长为负,初值应大于终值。

● 使用 For...Next 语句求解例 1 的代码:

```
s=1
For i=1 to 50
    s=s*i
Next i
```

例 4:每次将一张白纸沿垂直和水平中线对半撕成四张纸片,然后,取其中一张纸片照样撕成四张纸片,一直下去。计算撕了 n 次后纸片的张数。

问题分析

纸片的张数是一个等差数列,每撕一次,增加 3 张纸片。设撕了 n 次后纸片的张数为 result。

算法设计

第 1 步:令 result=1;

第 2 步:令 i=1;

第 3 步:如果 i>n,则执行第 6 步,否则执行第 4 步;

第 4 步: result=result+3;

第 5 步:i=i+1,转第 3 步执行;

第 6 步:结束。

● 使用 For...Next 语句求解例 4 的代码:

```
result = 1
For i = 1 To n
    result = result + 3
Next i
```

在 For...Next 语句的循环体中并没有出现改变循环控制变量 i 的值的语句 i=i+1,这是和 Do...Loop 语句最大的不同。For...Next 语句是通过 Step 和 Next 的配合来改变循环变量的值。如缺省 Step 语句,表示循环控制变量每次增加的步长为 1。

■ 循环的嵌套

与选择结构语句一样,循环结构语句也可以进行嵌套。

例 5: 编写程序打印下列图形。

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

问题分析

这个图形共有 6 行,第 1 行打印 1 个“*”,第 2 行打印 2 个“*”,……,第 6 行打印 6 个“*”。虽然可以用 6 句打印语句来实现,但当打印的行数较多时,显然需要更好的解决方法。本问题可归纳为:在第 i 行输出 i 个“*”, $i \leq 6$ 。

算法设计

第 1 步: 令 $i=1$;

第 2 步: 如果 $i \leq 6$, 则执行第 3 步, 否则执行第 6 步;

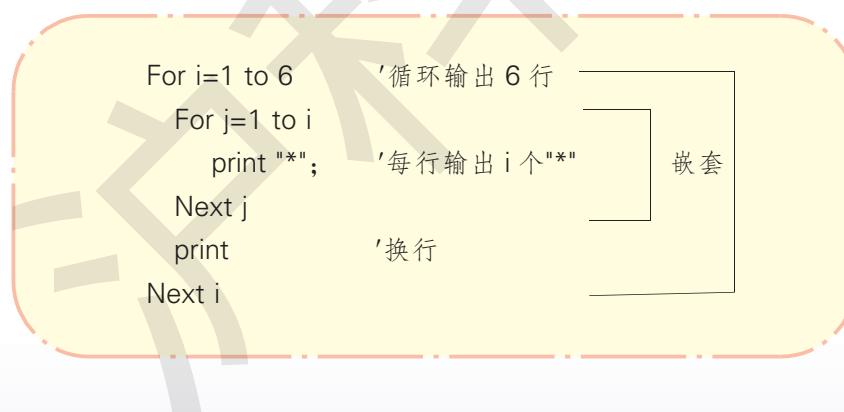
第 3 步: 令 $j=1$;

第 4 步: 如果 $j \leq i$, 则执行第 5 步, 否则换行, $i=i+1$, 转第 2 步执行;

第 5 步: 打印一个“*”, $j=j+1$, 转第 4 步执行;

第 6 步: 结束。

使用 For...Next 语句求解例 5 的代码:



在这个输出 6 行星号的程序代码中,For 循环语句中的循环体中还包含了一个 For 循环语句。像这样在一个循环结构中包含了另一个循环结构,称为循环的嵌套。外层的循环结构称为外循环,内层的循环结构称为内循环。要注意的是,外循环必须完全包含内循环,也就是说内外层 Next 语句的位置不能交叉。

■ 数组

在循环结构中,常需要处理大批数据,如使用简单变量对数据进行存取、操作,十分不方便,这时就要用到数组。

1. 数组及数组维数

数组是一组类型相同的数据的有序集合。它使用同一个名称来标识整个数组,每个分量称为数组元素,数组元素在数组中顺序排列。每一个数组元素都由一个唯一的顺序号(下标)来标识。数组维数由数组元素中下标的个数决定,一个下标表示一维数组,两个下标表示二维数组。VB 中有一维数组,二维数组,……,最多可达 60 维数组。

例如,表示一个班级中 60 名学生的英语成绩,可以使用一维数组 English(60),第 5 名学生的英语成绩是 English(5),第 i 名学生的英语成绩是 English(i)。又如,表示这个班级学生本学期 5 门功课的成绩,可以定义一个二维数组 Score(60,5),第 5 个学生的第 2 门功课的成绩是 Score(5,2),第 i 个学生的第 j 门功课的成绩是 Score(i,j)。

2. 数组的定义

数组必须先定义后使用,数组定义就是让系统在内存中分配一个连续的存储单元,用来存储数组元素,定义的内容一般包括数组名、类型、维数及数组大小。

数组的定义方法如下:

(1) 一维数组。

一维数组的定义格式为:

Dim 数组名(下标) [As 数据类型]

说明: ① 下标必须为常数,不可以为表达式或变量;

② 一维数组的大小为上界-下界+1,下标的最小下界为-32768,最大上界为 32767,如果省略下界,其默认值为 0。

例如,Dim A(10) As Double 定义了一个双精度型的数组,从 A(0)到 A(10)共有 11 个数组元素。Dim B(-3 To 5) As Integer 定义了一个整型的数组,从 B(-3)到 B(5)共有 9 个元素。

(2) 多维数组。

多维数组的定义格式为:

Dim 数组名(下标 1[,下标 2,⋯]) [As 数据类型]

说明: ① 下标个数决定数组的维数,最多 60 维。

② 每一维的大小为上界-下界+1,数组的大小等于各维大小的乘积。

例如,Dim C(-1 To 5, 4) As Long 定义了一个二维长整型数组,第一维的下标范围是-1~5,第二维的下标范围是 0~4,有 7×5 共 35 个数组元素。

3. 数组的引用

数组元素的引用是通过下标标识的。数组名加下标能唯一标识一个数组元素。

在使用数组时,通常使用一个变量来控制数组的下标。变量的值发生变化,所处理的数组元素也随之变化。在一个循环结构中,这个变量通常作为循环控制变量来控制一批数组元素的操作。

例如,下面的语句可为数组 A 中的 11 个数组元素赋值。

```
Dim A(10) As Integer
For i=0 To 10
    A(i)=i
Next i
```

下面再通过一个实例来说明数组对批量数据的操作方法。

例 6:求一个班级中 60 名学生的英语成绩总分和平均分。

```

Dim English(1 To 60) As Single
Dim Sum As Single
Dim Average As Single
'输入 60 名学生的英语成绩
For i = 1 To 60
    English(i) = InputBox("请输入第" & i & "名学生的英语成绩")
Next i
'求班级总分
Sum = 0
For i = 1 To 60
    Sum = Sum + English(i)
Next i
'求班级平均分
Average = Sum / 60
'输出结论
Print "总分:" & Sum
Print "平均分:" & Average

```

体验活动

1. 阅读计算 $1 \times 2 \times 3 \times 4 \times 5$ 的程序,回答以下问题,并上机验证。

```

Dim i As Integer
Dim s As Integer
s=1
For i=1 To 5 step 1
    s=s*i
Next i

```

- (1) 如果把程序中的 step 1 删去,能不能得到正确的结果?
- (2) 如果把程序中的 step 1 改成 step -1,应该如何修改程序?
- (3) 如果程序中缺少 s=1,最终 s 的值是多少?
- (4) 如果要求计算 $1 \times 2 \times 3 \times \dots \times n$,应该如何修改程序?
- (5) 如果要求计算 $2 \times 4 \times 6 \times 8 \times \dots \times 100$,应该如何修改程序?

2. 上机调试下列程序,在“运行”菜单中选择“中断”命令或在“调试”菜单中选择“逐语句”调试命令,检查运行状态下各个变量的值,确定发生错误的位置,找出程序错

> 接上页

误的原因,修改并重新运行程序,反复调试直到程序运行正确。

程序 1:求 $a=1+3+5+\cdots+(2n-1)$ 的值。

```
Dim a As Integer, i As Integer, n As Integer
```

```
n = Val(InputBox("请输入整数"))
```

```
For i = 1 To n
```

```
    a = a + 2 * n - 1
```

```
Next a
```

```
Print "a="; a
```

程序 2:求 100 以内能除尽 100 的数。

```
Dim i As Integer
```

```
i = 100
```

```
Do While i > 0
```

```
    i = i - 1
```

```
    If 100 Mod i = 0 Then Print i
```

```
Loop
```

	错误	原因	修改方法
程序 1			
程序 2			

3. 假设 2030 年我国某城市人口总数为 200 万, 预计该市人口年平均增长率为 9‰, 试计算、打印出在该市人口总数超过 500 万之前, 该市逐年的预计人口总数。

(1) 分析问题, 列出 2030 年后该市第 i 年人口总数的计算公式。

(2) 设计算法, 根据循环的控制条件及退出循环时的预计人口总数, 分析算法的正确性。

(3) 编写程序, 用列表框输出结果, 调试运行程序, 直至运行结果正确。

4. 随机产生一些 1~100 之间的整数, 直到产生的数为 50 为止。用四种 Do 语句编写同一个程序, 比较各种语句的格式、功能和区别。

提示: Fix(Rnd*100) 可以随机产生一个 100 以内的整数。

5. 打印九九乘法表。

提示: 需要用到循环的嵌套, 将循环控制变量分别作为乘数和被乘数。

6. 从键盘上输入 10 个整数, 并放入一个一维数组中, 然后进行如下交换: 第一个元素与第十个元素互换, 第二个元素与第九个元素互换, ……, 第五个元素与第六个元素互换。分别输出数组原来各元素的值和交换后各元素的值。

第五节 模块化程序设计

在实际生活中,当遇到一个大型的、复杂的问题时,人们通常会把这个大问题分解为几个部分,分别加以解决。如果某些部分还比较复杂,就再进行细分,当各部分问题被一一解决后,大问题也就迎刃而解了。

例如,学校举办运动会,通常会对运动会的各个环节进行分工和安排。具体机构和负责的工作见图 2-51。

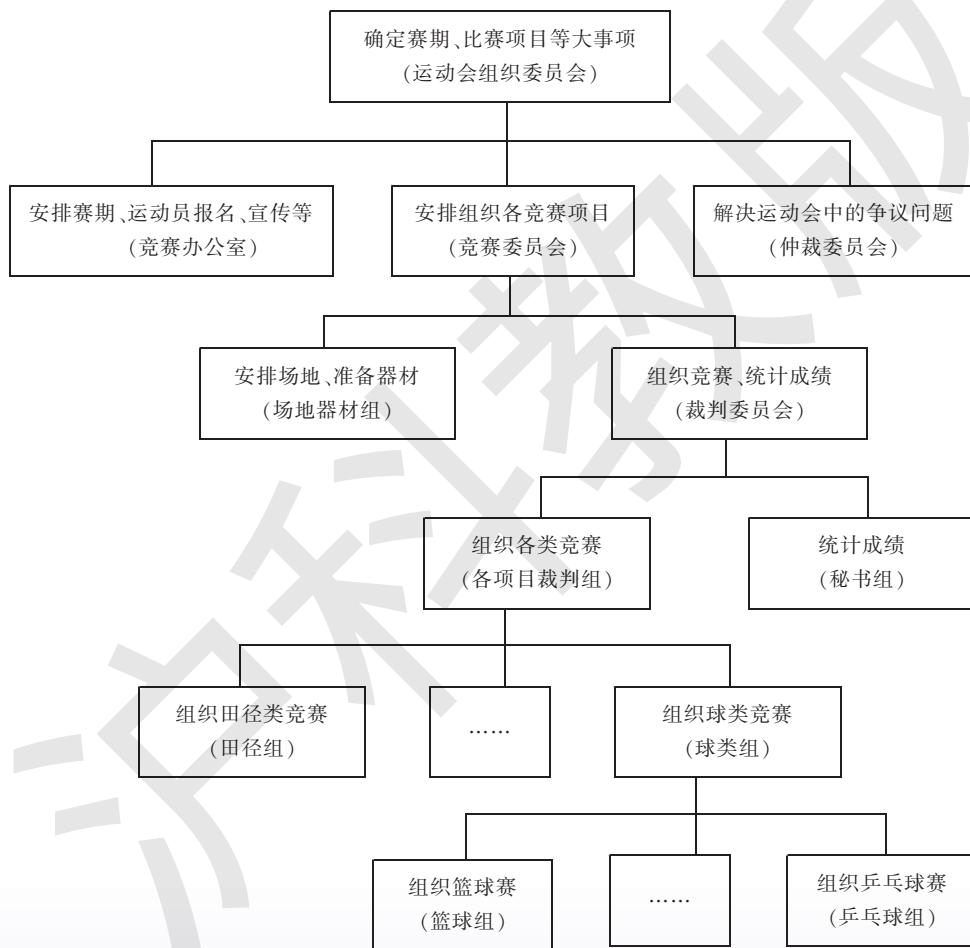


图 2-51 运动会机构模块示意图

当需要用计算机解决的问题比较大型,设计程序时,也可以采用这种方法。先从整体角度将程序分解为几个大的模块,再逐步进行细分,直至分解为若干个容易解决的模块。

活动一 了解自顶向下、逐步求精的程序设计方法

问题:科技馆开展暑期青少年计算机技能培训,培训包括三门课程:软硬件基础、文字处理技巧和网络技术应用。每

个部分的培训单独考核,考核以百分制计分。根据三门课程的成绩按比例计算总分,三部分所占的比例分别为:软硬件基础占 20%,文字处理技巧占 50%,网络技术应用占 30%。总分超过 70 分,每门功课不低于 60 分的学生能得到合格证书,每门功课都不低于 85 分的学生可以得到优秀证书。

培训部的李老师已经拿到一份第一期培训班 25 个学生三门课程的成绩表。请编写一个程序,使李老师只需要输入学生的姓名和成绩,计算机就能判断该学生能否获得合格证书或优秀证书,并输出学生的姓名、三门课程的成绩、总分,以及获得何种证书的信息。

1. 分析问题与设计算法。

(1) 要解决本问题,需要用到 25 个学生的姓名、三门课程的成绩并计算出他们的总分。

设使用以下数组:

姓名 : $sname(25)$ 三门课程的成绩 : $score(25,3)$

总分 : $sum(25)$

那么, $sname(i)$ 表示: _____

$score(i,j)$ 表示: _____

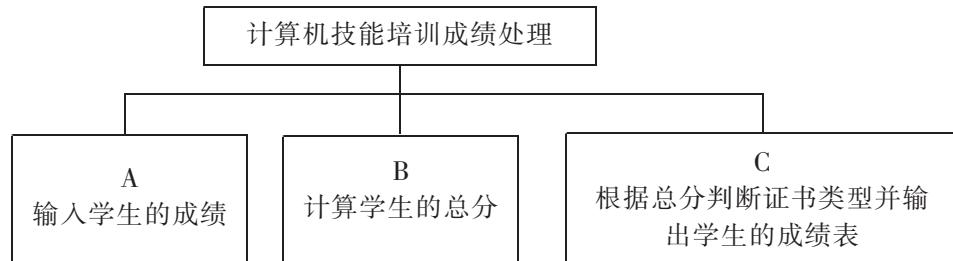
$sum(i)$ 表示: _____

(2) 阅读李老师分析本活动的过程。

需要解决的问题是:根据培训班学生三门课程的成绩表,统计学生的学习成绩和总分,并判断该学生能获得哪一种证书。可以将问题的解决分为三个环节:

- ① 输入学生的成绩;
- ② 计算学生的总分;
- ③ 输出学生的成绩表和证书类型。

解题的基本思路有了,将思路转化为计算机的算法,可作进一步细化,即先进行顶层设计,如图 2-52 所示,得到三个模块 A、B、C。



其中,模块A需实现的功能是输入35名学生的成绩,因此,可以使用一个循环结构,用以实现输入多位学生的成绩,如图2-53所示。

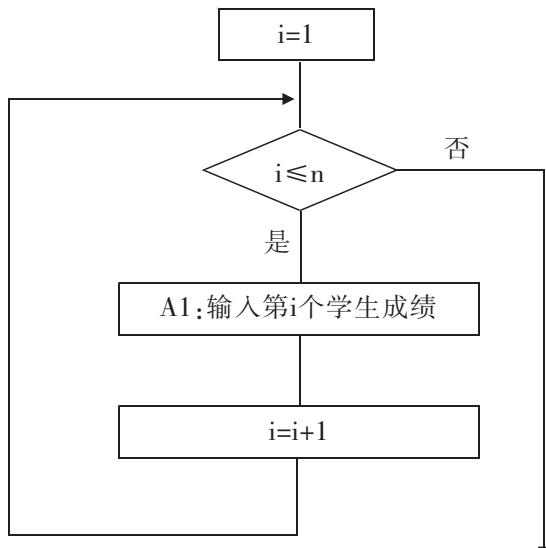


图 2-53 模块 A 的流程图

模块A中含有一个模块A1,模块A1可细化为实现输入一个学生的成绩的功能。模块A1是模块A的下一层模块,如图2-54所示,模块A1由4个输入操作构成。

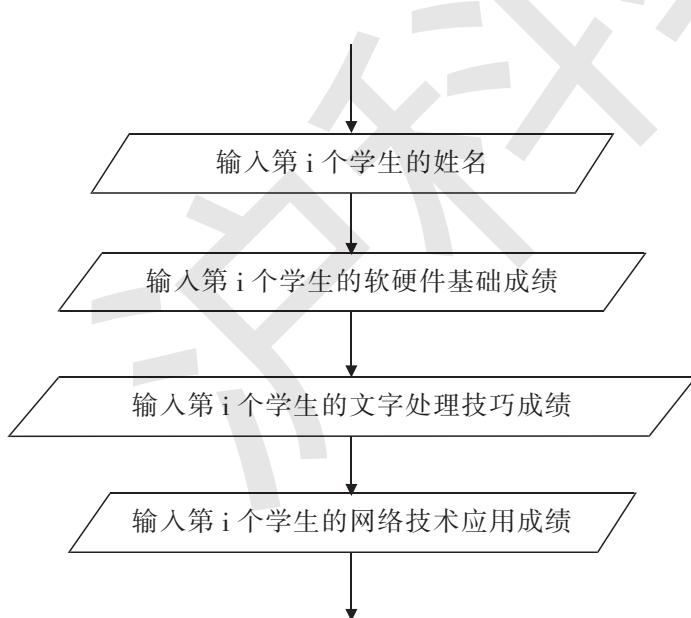


图 2-54 模块 A1 的流程图

模块B实现的功能是计算学生的总分,如图2-55所示,也可由一个循环结构实现。循环操作就是计算每一个学生的总分,通过一个运算式就可以完成。

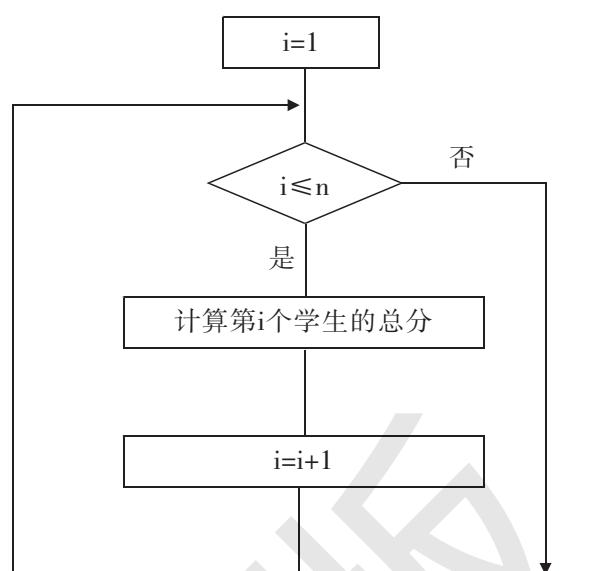


图 2-55 模块 B 的流程图

模块 C 的功能如图 2-56 所示，首先输出成绩表的表头，再输出成绩表的内容。成绩表的内容包括姓名、成绩、总分和证书类型。模块 C1 需要进一步细化，如图 2-57 所示。

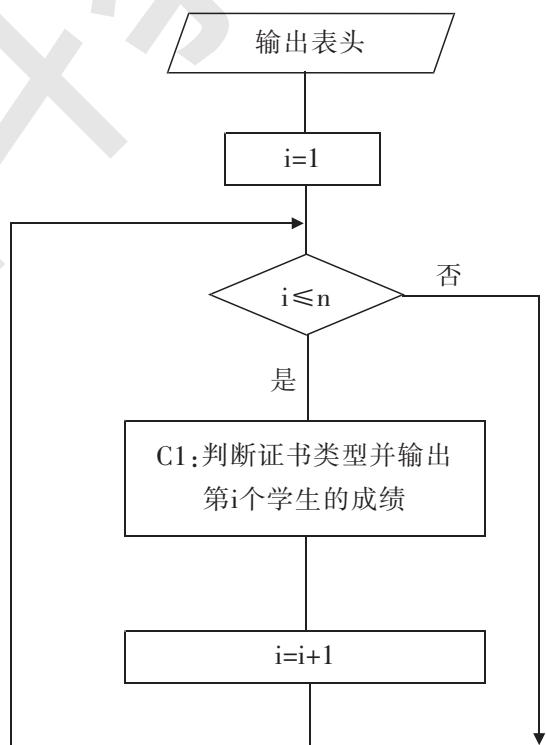


图 2-56 模块 C 的流程图

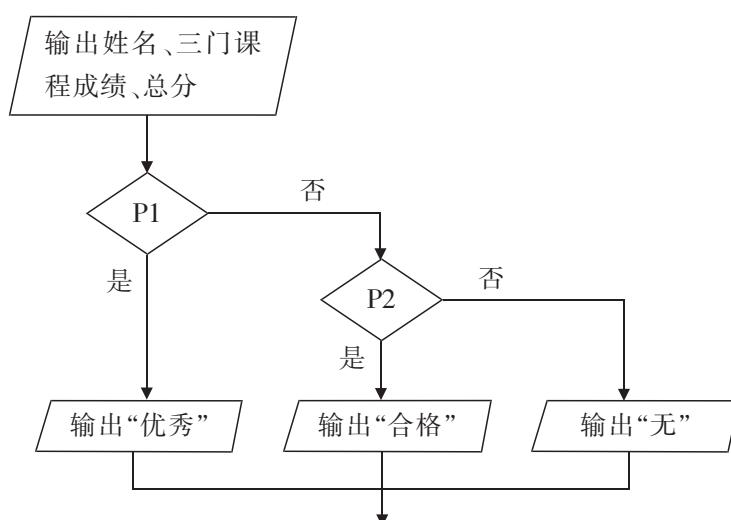


图 2-57 模块 C1 的流程图

(3) 列出计算第 i 个学生的总分的计算公式。

(4) 列出判断获得证书类型的 $P1$ 和 $P2$ 的表达式。

(5) 根据李老师的分析,用自然语言整理出完整算法。

第 1 步: 输入学生的成绩

循环 i 从 1 到 n

- ① _____
- ② _____
- ③ _____
- ④ _____

第 2 步: 计算学生总分

循环 i 从 1 到 n

$sum(i) =$ _____

第 3 步: 输出成绩表

输出表头

输出成绩表

循环 i 从 1 到 n , 步长为 1

① 输出第 i 位学生的姓名、三门课程成绩、总分

② 如果 _____

 则 输出“优秀”

 否则 如果 _____

 则 输出“合格”

 否则 输出“无”

2. 阅读程序代码,在下划线处完成功能注释。

```
Private Sub Form_Click( )
    Const n As Integer = 25
    Dim sname(25) As String '学生的姓名
    Dim score(25, 3) As Integer '学生的三门课程成绩
    Dim sum(25) As Integer '学生的总分
    Dim i As Integer
    Dim prompt As String
    '
    For i = 1 To n
        prompt = "输入第" & i & "个学生姓名"
        sname(i) = InputBox(prompt)
        prompt = "输入" & sname(i) & "的软硬件基础成绩"
        score(i, 1) = Val(InputBox(prompt))
        prompt = "输入" & sname(i) & "的文字处理技巧成绩"
        score(i, 2) = Val(InputBox(prompt))
        prompt = "输入" & sname(i) & "的网络技术应用成绩"
        score(i, 3) = Val(InputBox(prompt))
    Next i
    '
    For i = 1 To n
        sum(i) = score(i, 1) * 0.2 + score(i, 2) * 0.5 + score(i,
3) * 0.3
    Next i
    '
    '输出表头
    Print "姓名" + vbTab + "成绩 1" + vbTab + "成绩 2" +
vbTab + "成绩 3" + vbTab + "总分" + vbTab + "证书"
    '
    For i = 1 To n
        '输出一位学生的姓名、成绩和总分
        prompt = sname(i) & vbTab & score(i, 1) & vbTab &
score(i, 2) & vbTab & score(i, 3) & vbTab & sum(i) & vbTab
        '
        If (score(i, 1) >= 85 And score(i, 2) >= 85 And score(i,
3) >= 85) Then
```

> 接上页

```

Print prompt & "优秀"
Else
    If (sum(i) > 70 And score(i, 1) >= 60 And score(i,
2) >= 60 And score(i, 3) >= 60) Then
        Print prompt & "合格"
    Else
        Print prompt & "无"
    End If
End If
Next i
End Sub

```

活动二 使用函数和过程实现程序代码

在活动一的逐级分析过程中,会产生一些模块,如A、B、C、A1、C1。如果将这些模块作为一个独立的功能单元进行编码,在需要时进行调用,整个程序结构将变得更加简洁。在VB中,采用函数或过程可以创建一个独立的功能单元。试用函数和过程改写程序。

1. 根据模块A完成ReadIn()子过程,说出其实现的功能。

```

Sub ReadIn (i As Integer)
    prompt = "输入第" & i & "个学生姓名"
    sname(i) = InputBox(prompt)
    prompt = "输入" & sname(i) & "计算机文化的成绩"
    score(i, 1) = Val(InputBox(prompt))
    prompt = "输入" & sname(i) & "计算机维护维修实践的
成绩"
    score(i, 2) = Val(InputBox(prompt))
    prompt = "输入" & sname(i) & "因特网实践的成绩"
    score(i, 3) = Val(InputBox(prompt))
End Sub

```

提 示 板

在程序设计中可以定义一些实现相对独立功能的程序块,以备程序运行需要时调用。其中一种为**函数**,另一种为**过程**。

本例中,ReadIn()是子过程,i是一个整型的参数,可以接受调用时传递过来的参数值。

过程与函数详见学习指引。

2. 根据模块 C 完成 judge() 函数, 说出其实现的功能。

```

Function judge(i As Integer) As String
    If (score(i, 1) >= 85 And score(i, 2) >= 85 And score
        (i, 3) >= 85) Then
        judge = "优秀"
    Else
        If (sum(i) > 70 And score(i, 1) >= 60 And score
            (i, 2) >= 60 And score(i, 3) >= 60) Then
            judge = "合格"
        Else
            judge = "无"
        End If
    End If
End Function

```

提 示 板
judge()是一个函数, 有返回值, 故在参数括号后面还需要说明返回值的数据类型。详见学习指引。

3. 阅读下面使用子过程 ReadIn()和函数 judge()改写后的 Form_Click()事件过程, 找出调用子过程和函数的语句。

```

Const n As Integer = 25
Dim sname(25) As String '学生的姓名
Dim score(25, 3) As Integer '学生的三门课程成绩
Dim sum(25) As Integer '学生的总分
Private Sub Form_Click()
    Dim i As Integer
    Dim prompt As String
    '输入学生姓名和成绩
    For i = 1 To n
        Call ReadIn(i)
    Next i
    '计算总分
    For i = 1 To n
        sum(i) = score(i, 1) * 0.2 + score(i, 2) * 0.5 +
        score(i, 3) * 0.3
    Next i
    '打印成绩表
    Print "姓名" + vbTab + "成绩 1" + vbTab + "成绩 2" +
    vbTab + "成绩 3" + vbTab + "总分" + vbTab + "证书"
    For i = 1 To n
        prompt = sname(i) & vbTab & score(i, 1) &

```

提 示 板
调用子过程的语句是 Call 子过程名 (参数值列表), 调用函数的语句是函数名 (参数值列表)。详见学习指引。

接上页

```

vbTab & score(i, 2) & vbTab & score(i, 3) & vbTab & sum(i)
& vbTab & judge(i)
Print prompt
Next i
End Sub

```

4. 调试、运行程序。

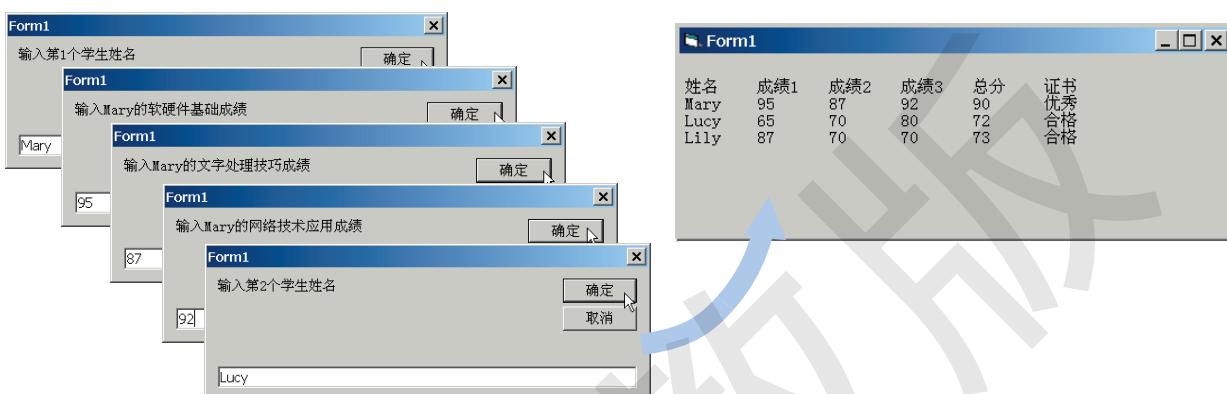


图 2-58 程序运行界面



讨论

对比活动一和活动二的两个程序,你认为哪一种程序结构更好?好在哪里?



学习指引

编写一个较大的程序时,过去常用的方法是:用一组语句实现一个功能,再用一组语句实现下一个功能,一组一组地写下去,直至组成一个大程序。这称为“自下而上”的程序设计方法。在解决本节活动中的“统计计算机技能培训成绩”问题时,李老师采用的却是另一种程序设计方法:先进行顶层设计,再对各部分进行细化。这种方法使程序的结构变得更加简洁,程序的调试也更加简单了。

■ 自顶向下、逐步求精的程序设计方法

现在广泛提倡的就是本节活动中李老师采用的称为“自顶向下 (top-down)、逐步求精 (stepwise refinement)”的程序设计方法。该方法是把整个设计过程分出层次来,逐步加以解决。每一步都是在前一步的基础上进行的,是对前一步设计的细化和具体化。这好比画一幅画,先进行整体规划,确定一幅画由哪些内容组成,勾勒出大致的轮廓,然后再考虑各部分的

形状是怎样的,最后再进行细节润色。

许多程序从逻辑上可以分为3个阶段:程序变量的初始化阶段;输入数据值并相应地调整变量的处理阶段;计算并显示结果的终止阶段。

在自顶向下的程序设计方法中,最顶层仅仅描述了程序的功能,它虽然是程序的一个完整表示,但很少提供足够的信息,不能直接用程序语句表示,因此需要进行细化处理。

对每一部分进行细化,遇到不能很清楚解决的步骤再向下一层延伸。依次进行第二层、第三层的细化,直至每一步都是一个基本操作为止。每次分步处理,都是算法在一个层次上的细化。

这样考虑问题,思路比较清晰,有条不紊,而且把一个大问题分解为若干个小问题考虑,降低了问题的复杂性,也便于程序实现。

■ 过程和函数

在设计一个大的应用程序时,经常会发生实现类同功能的程序段在程序中多次反复出现的情况。如果单独编写这些程序段,供程序中各部分,甚至各应用程序共享使用,就可以减少编写代码的工作量,而且也便于修改和查看。

在程序设计中可以定义一些实现相对独立功能的程序块,以备程序运行需要时调用。其中一种为**函数**(function),另一种为**过程**(process)。函数与过程的区别在于,函数经执行调用后,以带返回值的函数名为变量参与表达式运算,而过程只能通过过程调用语句调用执行,不能以过程名为变量参与表达式运算。

VB开发环境本身提供了许多可应用程序直接调用的函数,称为系统函数,如InputBox就是一个VB的输入函数。此外,VB中还包括处理事件的事件过程,如前面用到的Command1_Click()和Form_Click()。

VB的子过程与函数也可以由用户自己编写,并通过调用语句进行调用。

1. 自定义过程

在活动二中我们定义了一个ReadIn()子过程,用于实现输入一个学生的姓名和三门课程的成绩。定义一个子过程,首先要确定子过程的名称和参数,然后在过程体中书写实现具体功能的程序语句。过程的名称是用户自定义的,同样应遵守VB中标识符的命名规则,尽可能地体现所实现的程序功能,做到望名生义。参数是调用一个子过程时需要知道的数据,调用ReadIn()子过程时,参数i表示当前需要输入的是第i个学生的姓名和成绩。

定义子过程的一般格式:

```
Sub 子过程名([参数 As 类型[,参数 As 类型]...])
    [实现具体功能的程序语句]
End Sub
```

子过程定义好后,可以在任何一个函数和子过程中进行调用。

调用子过程的一般格式:

```
Call 子过程名(参数值列表)
```

子过程是没有返回值的,通过Call命令进行调用。在活动二程序的Form_Click()中调用了ReadIn()子过程,在调用时,将参数值i传递给该子过程,表示当前需要处理的是第i个学生。

例如,上一节的学习指引中有这样一道例题:

编写程序打印下列图形：

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

问题分析

该问题较难进行分析，在上一节中是采用 For...Next 语句的循环嵌套来完成的。如果先编写一个过程，用于在第 *i* 行打印 *i* 个“*”，该问题解决起来就方便多了。因为问题中有一个参数 *i*，而且只要求打印，不需要返回值，所以要构建的是一个子过程。

编写程序

定义子过程的代码如下

```
Sub Dayin(i As Integer)
    Dim j As Integer
    For j=1 to i
        print "*";
    Next j
    print
End Sub
```

完成打印图形时，需要调用 Dayin()子过程，代码如下：

```
Dim i As Integer
For i=1 to 6
    Call Dayin(i)
Next i
```

2. 自定义函数

在活动二中定义了一个 judge()函数，用于判断一个学生获得的证书是“优秀”、“合格”还是“无”。构造一个函数的思路一般与分析数学应用题有些类似，要知道已知什么、求什么。将已知条件放在参数中，在调用时通过参数传递得到，所求的值可通过函数的返回值返回。

函数定义的一般格式如下：

```
Function 函数名([参数 As 类型 1[,参数 As 类型 2]...]) As 类型 n
    [实现具体功能的程序语句]
    函数名=返回值
End Function
```

因为函数是有返回值的，所以在参数的括号后面需要定义函数的类型，函数的类型也就

是函数返回值的类型。调用 judge() 函数时,需要告诉函数当前需要处理的是第几个学生的成绩,所以需要一个参数来接受学生的序号;函数最后返回的是一个字符串,所以函数的类型是 String。在调用函数的时候,还要考虑接受函数的返回值,所以函数的程序语句中通常使用赋值语句,如 judge=“优秀”表示函数的返回值是字符串“优秀”。

调用函数的一般格式为:

函数名(参数值列表)

因为函数是有返回值的,所以调用的时候常和赋值语句一起使用,将返回值存放在变量中,其格式为:

变量名=函数名(参数值列表)

例如,计算 $7! + 11! - 10!$ 。

问题分析

可以先编写一个过程,用于计算 $n!$,计算方法在上一节中已有介绍。因为要计算 $n!$,过程中必须有一个参数接受 n 的值,并有一个返回值作为计算结果,所以需要定义的是函数。又因为 $n!$ 的计算结果通常较大,所以将函数的返回值类型设为 Long。

编写程序

定义函数的代码如下

```
Function fact(n As Integer)As Long
    Dim i As Integer
    Dim s As Long
    s=1
    For i =1 to n
        s=s*i
    Next
    fact=s
End Function
```

计算 $7! + 11! - 10!$ 时,需要调用 3 次 fact() 函数,代码如下:

```
Dim s As Long
s=fact(7) +fact(11) -fact(10)
```

函数调用可以包含在一个表达式中,这样函数返回值就能直接参加表达式运算。例如,在活动二中,judge() 函数的调用出现在如下的表达式中,函数返回的字符串直接参加字符串的“&”运算。

```
prompt = sname(i) & vbTab & score(i, 1) & vbTab & score(i, 2) & vbTab & score(i, 3)
& vbTab & sum(i) & vbTab & judge(i)
```

在函数的参数中也可以包含一个函数的调用。例如,如果定义了一个 Max() 函数,用于得到两数之中的较大数,则下面的函数调用实现的就是求三数之中的较大数。

$z=Max(Max(a,b),c)$ 。

3. 参数传递

主调过程通过参数传递数据给被调过程,过程间传递的主要是执行一个过程需要的已知数据。例如要调用 judge() 函数,首先要传递学生的总分和三门课程的成绩。要调用 ReadIn() 子过程,首先要传递当前需要处理的是第几位学生,如图 2-59 所示。通常称函数定义处的参数为形式参数。调用时所对应的参数值是实在参数。实在参数可以是一个常量也可以是一个变量,还可以是一个表达式。

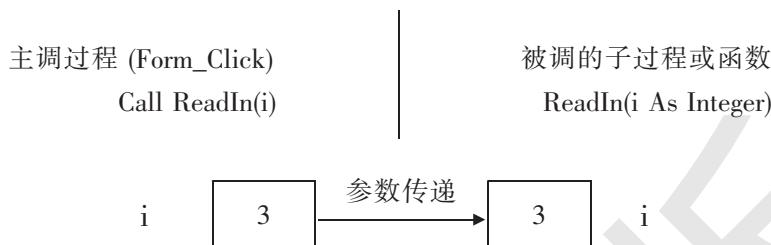


图 2-59 参数传递

形式参数与在过程中定义的变量大致相同,其区别在于,形式参数的初值是由调用一个过程时传递的实在参数数据得到的。所以调用一个过程时,实在参数的个数、数据类型、位置要与形式参数一一对应。

■ 模块化设计的思想

大多数解决现实问题的程序都比书中所介绍的程序要大得多,对于一个复杂的问题通常需采取“分而治之”的策略,也就是模块化的思想。

运用软件工程的思想来组织软件开发工作,一般要经过需求分析、设计程序、编码等阶段。需求分析所做的事情是,根据用户的要求确定程序应当具备哪些功能、达到什么性能。设计程序阶段大致又可再分为两个子阶段,即程序的模块化设计和模块内的逻辑设计。在模块化设计中,要按照需求分析把程序分割成许多模块(module),建立以模块为单位的多层次结构体系,各个模块之间形成调用关系。模块化之后还要确定每一模块内的具体算法,并准确表达这些算法。整个程序设计完成之后,就可以使用特定的程序语言来进行编码了。

本节活动中对计算机技能培训成绩处理程序进行的模块化设计,可以表示为如图 2-60 所示的模块结构图:

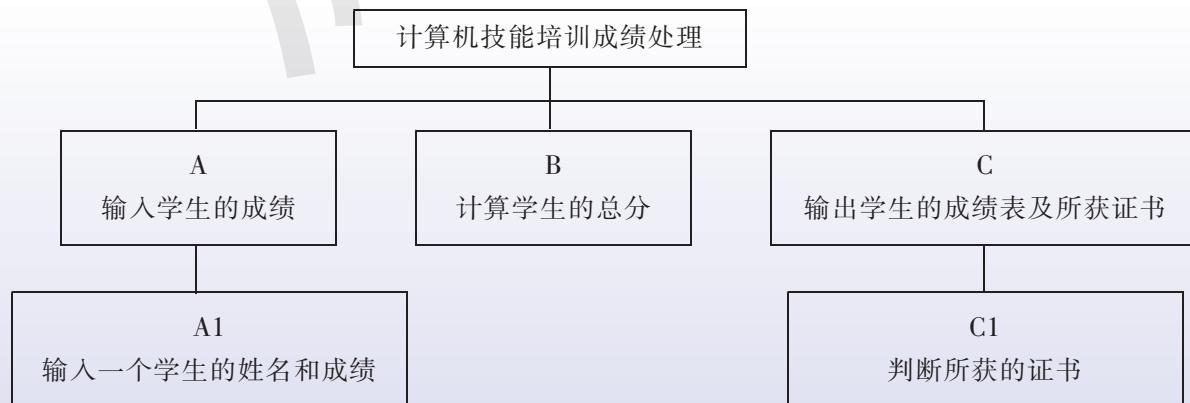


图 2-60 模块结构图

整个程序由模块 A、模块 B 和模块 C 构成。模块 A 处理的是学生成绩的输入,在模块 A 中多次调用了模块 A1。模块 A1 处理的是一个学生的姓名和成绩的输入;模块 C 处理的是学生成绩表及所获证书的输出,在模块 C 中多次调用了模块 C1,模块 C1 处理的是一个学生所获证书的判断。在顶层的设计中,不需要考虑学生信息的输入细节,也不需要考虑判断学生所获证书的实现方法,这两个细节交给模块 A1 和 C1 去考虑。

又如图 2-61 演示了一个工资处理程序模块化设计的过程。一个程序首先被分为三个基本步骤:输入、处理、输出。再对每一个模块继续向下细化,直至每一个模块都相对独立地完成一个基本功能。

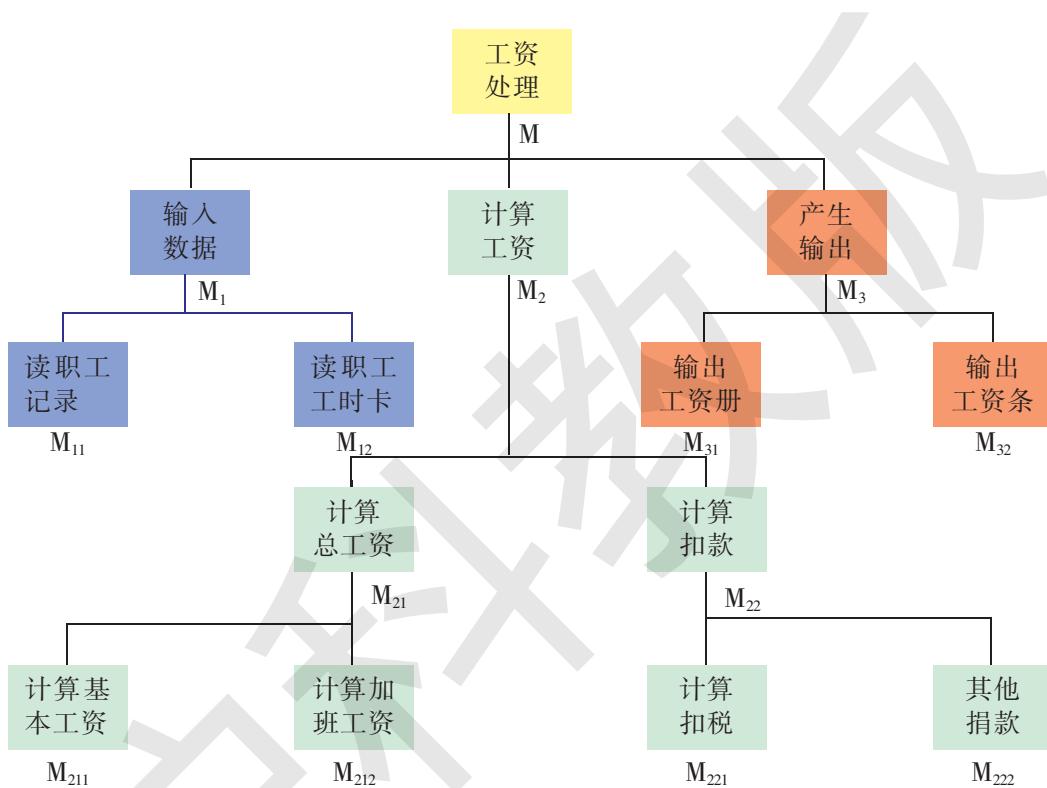


图 2-61 模块化设计示例

体验活动

- 编写程序,打印 1~1000 之间能被 3、5、7 整除的整数。

提示:先编写一个函数用来判断某个数能否被 3、5、7 整除,如果能,则返回一个布尔值 True,否则返回布尔值 False。然后在程序中调用这个函数来判断 1~1000 中能被 3、5、7 整除的整数。

- 编写程序,在窗体中输出三组同心圆,这三组同心圆的圆圈个数分别为 2、4、6,如图 2-62 所示。

> 接上页

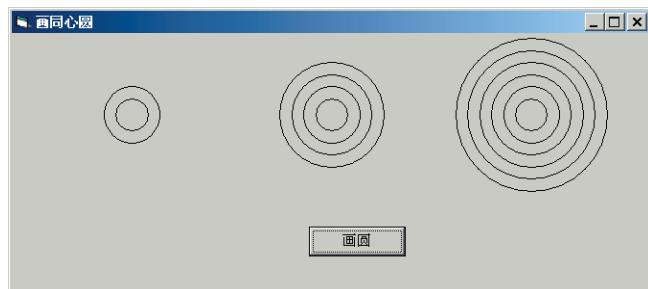


图 2-62 运行结果

要求:在程序中调用过程完成同心圆的绘制。

提示:

(1) 可采用 Circle 方法在窗体中画出一个圆。Circle 方法的格式如下:

Circle 圆心坐标,半径长度,颜色

例如:“Circle (1500,1000),200”,该语句将在窗体坐标(1500,1000)处画一个半径为 200 的圆。

(2) 本问题中,每组同心圆的个数和圆心的横坐标位置不同,在调用绘制同心圆的过程时可通过参数传递来控制。

3. 编写一个猜数游戏的程序。该程序在启动时随机产生一个 1~1000 之间的整数,用户输入猜测的数后,程序立即反馈是猜中了,还是猜大了,或是猜小了。若没有猜中,则继续猜数直至正确。猜中后,程序根据用户猜的次数进行评价,若次数小于 10,输出“你非常幸运”;若等于 10,输出“猜数方法正确”,若大于 10,输出“需要改正猜数的方法”。程序运行界面参见图 2-63。



图 2-63 程序运行界面



后续活动

1. 试为小学生编写一个四则运算练习程序,该程序应具有如下的功能:由计算机自动出 10 道题,学生答题时由计算机判断输入的答案是否正确。答题结束后,计算机将给出所做题目的正误判断,并统计得分,给出评价。考虑到小学生能力的不同,计算机出题的题目可分初、中、高三个等级:初级的题目只涉及 10 以内的四则运算;中级的题目涉及 100 以内的四

则运算；高级的题目涉及 1000 以内的四则运算。根据小学生的需要还可选择做单一的加法或减法练习。

要求：

- (1) 分析问题，完善并完成如图 2-64 所示的模块结构设计。
- (2) 使用可视化编程方法设计图形用户界面。
- (3) 尽可能在程序中使用过程或函数。

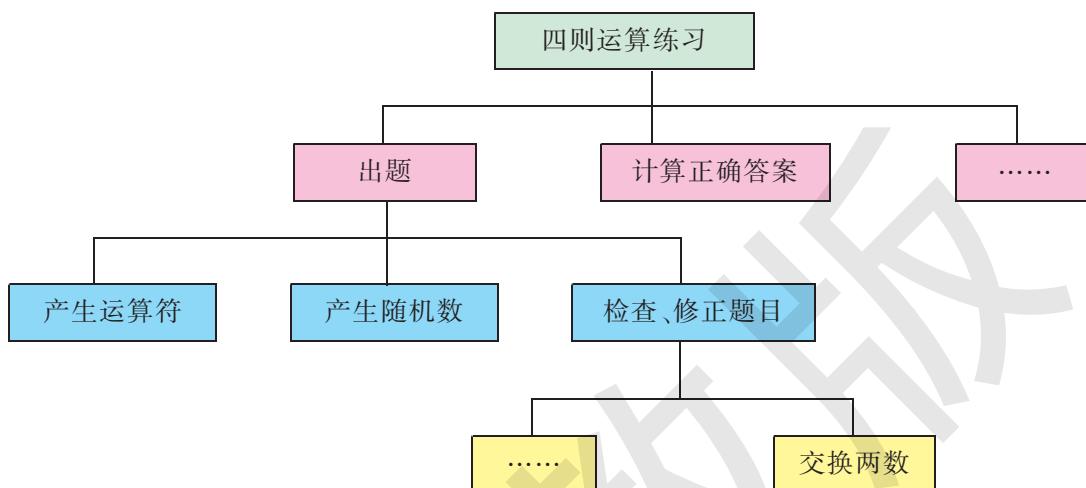


图 2-64 模块结构图

提示：

- (1) 在小学生的四则运算中，减法的减数应小于等于被减数，除法的除数应小于等于被除数。另外，除数不能为 0。因此，出题后必须对除法或减法的题目进行检查和修正。如果遇到减数大于被减数时，可采用交换两数的办法解决这一问题；如果遇到除数为 0，则重新生成除数。
- (2) 在计算答案时，应考虑到小学生的除法运算需要分别求出商和余数两个值。
- (3) 可运行配套光盘中的“operation.exe”程序，参考该程序的图形用户界面，如图 2-65

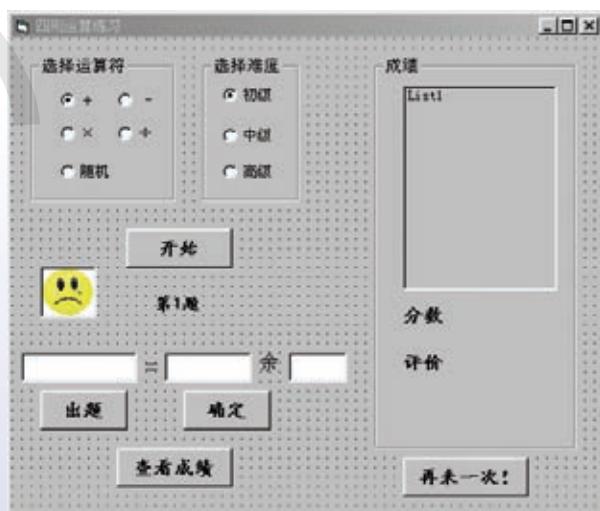


图 2-65 程序运行界面

所示。

2. 现在城市普通居民购买商品房通常采取个人购房抵押贷款的方式,解决一次性付款有困难的问题,比如首先支付35%的房款,剩下的分10年还清,请调查你所在的城市中购买商品房贷款的有关政策,设计一个贷款计算器,计算通过贷款购买一套商品房后,每个月所需支付的还款金额。程序运行界面参见图2-66。

提示:

(1) 城市居民的个人住房贷款一般会由个人住房公积金贷款和商业贷款构成,下表给出了某银行年个人住房贷款利率表,不同种类和不同年限的贷款年利率是不同的。

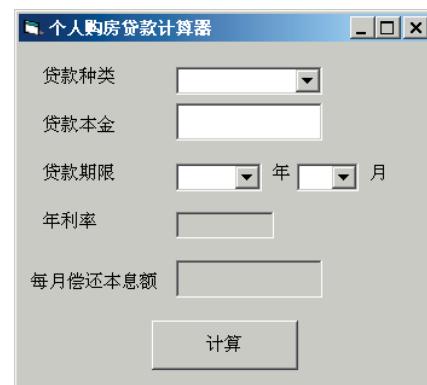


图2-66 程序运行界面

贷款项目		年利率(%)
个人住房公积金贷款	五年以下(含五年)	2.75
	五年以上	3.25
(自营性)个人购置住房 (按揭)贷款	五年以下(含五年)	4.75
	五年以上	4.90

(2) 可在窗体中使用一种新控件:组合框。

组合框的特点是由一个文本框和下拉列表框构成,具有两种控件的共同属性和特征。用户可以直接在组合框中输入值,也可以从列表中选择列表项到组合框。要获取组合框中的值,可以通过Text属性;如果想获取的是所选择列表项的序号,可以通过ListIndex属性,列表项序号从0开始计数。在Form_Load事件过程中可使用AddItem方法为组合框装入列表项,在Click事件中可以对用户选择列表项的操作作处理。

(3) 窗体功能分析。

① 贷款种类组合框。选择项有两种:个人住房公积金贷款和商业贷款。Click事件过程:重新确定年利率。贷款类型可以通过ListItem属性得到。

② 贷款期限年份和月份组合框。年份的选择项从0到30,月份的选择项从0到11。Click事件过程:重新确定年利率。贷款的年份和月份可以通过Text属性得到。

③ “计算”命令按钮。Click事件过程:计算每月偿还的本息额。

(4) 可使用函数来完成以下三个功能模块。

① 确定年利率;

② 计算个人住房公积金贷款的每月偿还本息额;

③ 计算商业贷款的每月偿还本息额。


评价

一级指标	评价内容	评价(★★★、★★、★)		
		自评	互评	教师评
信息技术知识	了解 VB 可视化开发环境。			
	了解对象、属性、方法、事件、事件驱动的基本概念。			
	了解窗体、控件及其作用。			
	了解 VB 语言中的语句、数据类型、常量、变量、表达式、系统函数及使用方法。			
	了解选择结构实现语句 If…Then、Select Case 的区别及用法。			
	了解循环结构实现语句 For…Next、Do…Loop 语句的区别及用法。			
	了解函数和过程的实现及调用方法。			
	了解模块化程序设计思想。			
信息技术能力	能使用说明语句、赋值语句、输入输出语句等基本语句实现顺序结构。			
	能根据问题分析及算法描述，适当地使用选择语句来编程实现选择结构，解决问题。			
	能根据问题分析及算法描述，适当地使用各种循环语句实现循环结构，解决问题。			
	学会在 VB 中调试程序。			
	能使用函数和过程实现程序。			
行为	能积极参与小组讨论，积极发表自己的意见并听取同学的意见，合作完成问题分析、设计算法、编写程序等环节。			
	在阅读程序的过程中，关注程序的编写风格。			
	在使用过程和函数实现程序时能关注模块化程序设计思想的优越性。			
体会：				