

高等学校计算机基础教育规划教材

Visual Basic.NET 2008

任务导引教程

谢书良 主编

清华大学出版社



高等学校计算机基础教育规划教材

Visual Basic.NET 2008

任务导引教程

谢书良 主 编

清华大学出版社
北 京

内 容 简 介

本书是为从未学习过编程又期望能简易掌握编程的读者编写的入门教材。全部采用可视化手段描述,按任务导引教学方法进行编写,十分注重可读性和可用性。

本书内容由10章构成,第1章为VB.NET概述,第2章为运算符和表达式,第3章为程序控制结构,第4章为数组,第5章为过程,第6章为类的封装、继承和多态,第7章为可视化编程基础,第8章为数据库操作,第9章为文件操作,第10章为学生成绩管理系统的设计,第10章为深入学习VB.NET程序的编写而提供了一个完整的范例来供读者模仿、应用。

本书还为授课教师提供精心设计的配套电子课件、全部任务源代码、自我检测练习题答案和部分自我检测练习题、操作实践题的源代码。

本书可作为各级各类高等院校涉及程序设计的相关专业开设“程序设计”课程的教材,也可作为工程技术人员参考用书和有志于程序设计的社会青年的自学用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Visual Basic .NET 2008 任务导引教程/谢书良主编.--北京:清华大学出版社,2014

高等学校计算机基础教育规划教材

ISBN 978-7-302-34373-8

I. ①V… II. ①谢… III. ①BASIC 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第257287号

责任编辑:张龙卿

封面设计:徐日强

责任校对:刘 静

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795761

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:18.25

字 数:418千字

版 次:2014年1月第1版

印 次:2014年1月第1次印刷

印 数:1~3000

定 价:39.00元

产品编号:053748-01

前言

当前,IT 技术发展迅猛,日新月异。在计算机应用日益广泛的形势下,软件的概念和程序设计的应用知识已逐渐成为人们渴求的新目标。如果说数学是“培养抽象思维的工具”,物理学是“培养逻辑思维的工具”,那么程序设计则是“培养计算思维的工具”。有人预言,到 2050 年“计算思维”将成为全人类的主要思维方式,“计算思维”的精髓是“程序思维”,鉴于此,说“人人都应学习、懂得一点编程”并不为过,对于理工类乃至于文史类的高校学生来说,学一点程序设计基础和应用知识都十分必要。

程序设计语言很多,本教材之所以选择 VB.NET 语言作为零起点的程序设计入门语言,是因为 Basic 语言是国际公认的一种最基本、最简单又最规范的编程语言。正因为如此,世界上发达国家和计算机教育成功的几个软件大国都选 Basic 语言作为程序设计的入门语言。VB.NET 语言是在 Basic 语言基础上增加了可视化和面向对象等重要内容,逐步发展起来的一种安全、稳定、简单、规范、完全面向对象的编程语言。VB.NET 2008 更凭借其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件的编程,必将成为教学程序设计语言入门课程的首选和优选。入门教材应体现零起点、易学、好用,问题在于现在真正适合作为入门使用的此类教材却十分稀缺。

编写本教材就是基于这一初衷,能为学习程序设计课程的起始年级且只安排一个学期教学程序设计课程的有关本、专科专业学生提供一本真正零起点的入门教材,为期望从零开始能顺利学习程序设计理论且能较快掌握程序设计技能的广大读者提供一本简单通俗、乐学易用的程序设计入门书。

教材选取“任务导引法”思路进行编写,即在教师的具体指导之下,引发学生的学习兴趣和学习动机。这样既有利于维护教学内容的体系,也便于检测教学进程的效果。为此,本教材每章之后都专门安排了“操作实践”和“自我检测”内容。

本教材的一个重要特色是全部采用可视化手段进行描述,这样便可以有效地帮助读者消除对编程的神秘感,从而对编程产生浓厚的兴趣。

教材内容由 10 章构成,第 1 章为 VB.NET 概述,第 2 章为运算符和表达式,第 3 章为程序控制结构,第 4 章为数组,第 5 章为过程,第 6 章为

类的封装、继承和多态,第7章为可视化编程基础,第8章为数据库操作,第9章为文件操作,第10章为学生成绩管理系统的设计,为深入学习 VB.NET 程序的编写而提供了一个完整的范例供读者模仿、应用。

本教材可用 64 课时或更少课时完成,如果有课时余量,可以安排“课程设计”环节,以达到“学以致用”的效果。

“多思考,勤上机”是学好程序设计语言的重要条件,学习编程要细心、耐心并要有恒心。只有那些有志气、有毅力的人,才能品尝编程带来的愉悦。

本教材的编写是顺应程序设计语言发展历史潮流的一个新的尝试,肯定会存在许多不足之处,诚盼不吝指正,使其不断完善。

谢书良

2013 年 8 月

目 录

第 1 章 VB.NET 概述 1

1.1 程序设计的基本概念 1

1.2 几个重要概念 5

1.3 Windows 应用程序的创建 8

1.4 VB.NET 程序的上机环境——Visual Studio 2008 开发

环境简介 9

1.5 初试 VB.NET 编程 13

操作实践 19

自我测试 19

第 2 章 运算符和表达式 20

2.1 简单数据类型 20

2.2 变量和常量 22

2.3 算术运算符与算术表达式 23

2.4 赋值运算符与赋值表达式 25

2.5 关系运算符和逻辑运算符 27

操作实践 32

自我测试 33

第 3 章 程序控制结构 34

3.1 算法概述 34

3.2 顺序结构的程序 37

3.3 分支选择结构与 If 语句 38

3.4 多分支选择结构——If 语句的嵌套 40

3.5 多分支 Select Case 语句 43

3.6 循环结构和循环语句 46

3.7 循环的嵌套 53

3.8 流程控制的转移 57

3.9 结构化程序的编写 59

操作实践 64

自我测试	65
第 4 章 数组	67
4.1 一维数组的声明和使用	67
4.2 二维数组的声明和使用	74
4.2.1 二维数组的声明	74
4.2.2 二维数组的初始化	75
4.2.3 二维数组元素的引用	75
4.2.4 二维数组的应用	77
4.3 改变数组大小	80
4.4 For Each 语句	81
4.5 LBound() 函数和 UBound() 函数	82
4.6 控件数组	83
操作实践	87
自我测试	88
第 5 章 过程	90
5.1 过程概述	90
5.2 Sub 过程	91
5.3 函数过程	92
5.4 参数传递	94
5.5 过程的嵌套调用和递归调用	98
5.6 实型数据在编程中的使用	102
5.7 变量的作用范围	107
5.8 过程的作用范围	111
操作实践	114
自我测试	114
第 6 章 类的封装、继承和多态	116
6.1 类的声明和对象的创建	116
6.2 类成员的种类及其可访问性	118
6.3 字段	118
6.4 属性	119
6.5 类的构造方法和析构方法	123
6.6 方法及方法的重载	125
6.7 类的继承	127
6.8 多态性	135
操作实践	140

自我测试.....	140
第 7 章 可视化编程基础.....	142
7.1 窗体	142
7.2 控件	145
操作实践.....	161
自我测试.....	162
第 8 章 数据库操作.....	163
8.1 Access 数据库表的创建	163
8.2 数据库的简易访问	165
8.3 ADO.NET 概述	178
8.4 使用 Connection 对象连接数据库	179
8.5 创建应用程序来访问数据库	185
操作实践.....	213
自我测试.....	213
第 9 章 文件操作.....	215
9.1 文件概述	215
9.2 文件操作函数	216
9.3 顺序文件的读写操作	220
9.4 随机文件的读写操作	224
9.5 流与 System.IO 模型	228
第 10 章 学生成绩管理系统的设计	234
10.1 软件开发的一般过程	234
10.2 学生成绩管理系统的设计	234
10.3 任务实施	238
附录 A ASCII 码字符集	278
附录 B 运算符的优先级	279
附录 C 常用系统函数.....	280
附录 D 关键字.....	282
参考文献.....	283

第 1 章 VB.NET 概述

教学要求

1. 了解程序及程序设计的概念。
2. 了解面向对象程序设计的基本特点。
3. 掌握 VB.NET 程序的基本结构和程序运行的流程。
4. 熟悉 VB.NET 开发环境的基本用法。

1.1 程序设计的基本概念

1. 程序

本书从如何计算两个数的平均值这样一个最简单的问题讲起。

如果这两个数是 3 和 5,几乎可以不假思索地说出它们的平均值是 4;

如果这两个数是 23763965432 和 8456234445446456,它们的平均值是多少?那只能由计算机去完成。

不管是怎么算,人和计算机的计算步骤都是:

- (1) 确定要计算的是哪两个数;
- (2) 先求出两个数之和;
- (3) 再将此和除以 2;
- (4) 最后报告计算结果。

其实计算机自身并不会计算,必须由人来教会它。那么人们应该做什么呢?就一般的问题来说,人们要做的事应该是:针对要完成的任务,编排出正确的方法和步骤,并且用计算机能够接受的形式,把方法和步骤告诉计算机,指挥计算机完成任务。

解决问题的方法和步骤,以计算机能够理解的语言表达出来,就称为“程序”。程序是要计算机完成某项工作的代名词,是对计算机工作规则的描述。

计算机软件是指指挥计算机硬件的,没有软件,计算机是什么事也做不了,而软件都是由各种程序构成的,程序是软件的灵魂。

2. 程序设计

人们要利用计算机解决实际问题,首先要按照人们的意愿,借助计算机语言,将解决问题的方法、公式、步骤等编写成程序,然后将程序输入到计算机中,由计算机执行这个程

序,完成特定的任务,这个设计和书写程序的整个过程就是程序设计。简言之,为完成一项工作的规则的过程设计称为程序设计,从根本上说,程序设计是人的智力克服客观问题的复杂性的过程。

程序设计是根据给出的具体任务,编写一个能正确完成该任务的计算机程序。计算机程序是有序指令的集合,或者说是能被计算机执行的具有一定结构的语句的集合。

图 1-1 所示是一个简化了的计算机工作过程示意图,计算机的实际工作过程当然比这复杂得多,但它还是完整地体现了其基本工作原理,尤其体现“软件指挥硬件”这一根本思想。在整个过程中,如果没有软件程序,计算机什么也干不了,可见软件程序多么重要。如果软件程序编得好,计算机就运行得快而且结果正确;程序编得不好,则可能需要运行很久才出结果,且结果还不一定正确。程序是软件的灵魂,CPU、显示器等硬件必须由软件指挥,否则它们只是一堆没有灵性的工程塑料与金属的混合物。在这里就是要教会读者怎样用编程语言又快又好地编写程序(软件)。

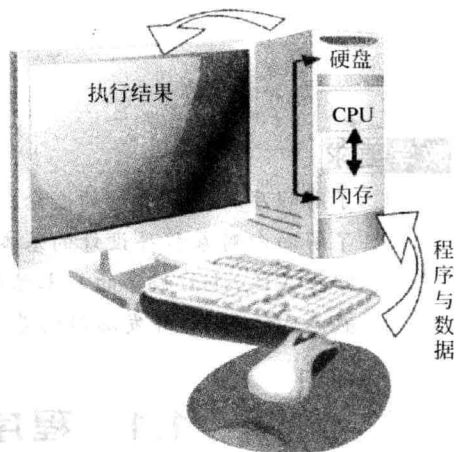


图 1-1 计算机工作过程示意图

计算机直接能够读懂的语言是机器语言,也叫作机器代码,简称机器码。这是一种纯粹的二进制语言,用二进制代码来代表不同的指令。

下面这段程序是用我们通常使用的 x86 计算机的机器语言编写的,功能是计算 $1+1$ 。

```
10111000
00000001
00000000
00000101
00000001
00000000
00000000
```

这段程序看起来像“天书”,在用按钮开关和纸带打孔的方式向计算机输入程序的时代,程序员编写的都是这样的程序。很明显,这种程序编起来很费力气,很难读懂。从那时候起,让计算机能够直接懂得人的语言就成了计算机科学家们梦寐以求的目标。

有人想出了这样的办法,编一个可以把人类的语言翻译成计算机语言的程序,这样计算机就能读懂人类语言了。这说起来容易,做起来难。就拿计算 $1+1$ 来说,人们可以用“ $1+1$ 等于几”、“算一下 $1+1$ 的结果”、“ $1+1$ 得多少”……多种说法,再加上英语、法语、日语、韩语、俄语……来描述。如果想把这些都自动转换成上面的机器码,是可望而不可及的事。所以人们退后一步,打算设计一种中间语言,它还是一种程序设计语言,但比较容易翻译成机器代码,且容易被人学会和读懂,于是诞生了“汇编语言”。

用汇编语言计算 $1+1$ 的程序如下所示。

```
MOV AX, 1
```

```
ADD AX, 1
```

这个程序的功能是什么呢？从程序中 ADD 和 1 的字样，或许我们能猜个大概。没错，它还是计算 $1+1$ 的。这个程序经过编译器（也是一个程序，它能把 CPU 不能识别的语言翻译成 CPU 能直接识别的机器语言）编译，就会自动生成前面的程序。这已经是很大的进步了，但并不理想。这里面的 MOV 是什么含义？好像是 Move 的缩写。这里的 AX 又代表什么？这是一个纯粹的计算机概念。从这个小程序，我们能看出汇编语言虽然已经开始贴近人类的语言，但还全然不像我们所期望的那样，里面还有很多计算机固有的东西必须要学习。它与机器语言的距离很近，每行程序都直接对应上例的三行代码。以后你有机会学习、使用汇编语言，到那时你将学到更多有关计算机内部的知识。

程序设计语言要无限地接近自然语言，所以它注定要不停地发展。此时出现了一道分水岭，人们把机器语言和汇编语言称为低级语言，把以后发展起来的语言称为高级语言。低级语言并不比高级语言“低级”，而是说它与计算机（硬件）的距离较近因而级别比较低。高级语言高级到什么程度呢？我们先介绍一个很著名的编程语言——Basic，看它是怎样完成 $1+1$ 计算的。

用 Basic 语言计算并显示 $1+1$ 的内容如下：

```
PRINT 1+1;
```

英文 PRINT 的中文意思是打印。比起前两个例子，它确实简单了不少，而且功能很强。前两个例子的计算结果只保存在计算机的“心脏”（CPU）内，并没有输出给用户。这个例子直接把计算结果显示在屏幕上，它才是真正功能完备的程序，从这个例子相信你已体会到高级语言的魅力了吧。



阅读材料

目前较流行的几种高级语言简介

因为高级语言易学、易用、强大，所以它发展很快，其种类之多完全可以用“百花争艳”来形容。据一位民间人士耗时多年的不完全统计，目前已经有超过 2500 种计算机语言，其中绝大多数都是高级语言。

BYTE 杂志创刊 20 周年特刊里一篇题为“程序设计语言发展简史”的文章中，列举了 1946 年起到 1995 年为止，在社会上产生一定影响的程序设计语言约有四十几种，实际上目前广泛流行的只有几种。图 1-2 是日本计算机教育家三田典玄先生提供的一幅这几种程序设计语言与软件、硬件、系统、用户四大领域的关系图。

由图中不难看出，Basic 语言正好处于中间位置，它兼顾了软件、硬件、系统和用户四大领域的要求，几乎成了其他任何一种语言无法替代的语言。几种常用的高级语言，在漫长的发展过程中互相渗透，互相借鉴，逐步形成了你中有我、我中有你的局面，其中变化最大的首当 Basic 语言，它最初是从 FORTRAN 语言脱胎而来，以后又陆续吸收了其他语言的诸多特点。国外一些计算机教育家早已注意到了这一点，他们发现已掌握 Basic 的人可以很快地学会任何一门（哪怕是很晦涩的）程序设计语言。他们都公认 Basic 语言是易学、易用、发展快、变化大、覆盖面广的初学者的入门语言。美国、日本等发达国家都选

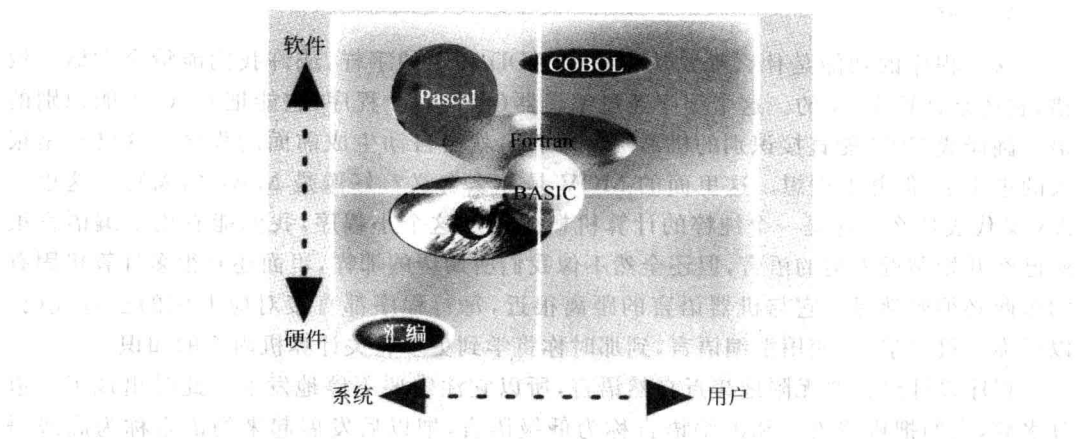


图 1-2 几种程序设计语言与软件、硬件、系统、用户四大领域的关系图

择 Basic 作为程序设计课程的入门语言,不无道理。我国在 20 世纪 80 年代以前,也几乎都是选择 Basic 语言作为程序设计课程的入门语言。

到了 20 世纪 90 年代,Basic 逐步被 QBasic 替代。QBasic 是纯粹的面向过程的解释型语言,内容又过于烦琐,特别是它不能编译,从而不能脱离系统运行,使用十分不便。

计算机技术发展很快,面向对象的事件驱动程序的开发平台早已广泛应用,微软在推出 Windows 平台之后,也把 Basic 语言扩展到 Windows 平台下,并且增加了可视化编程的成分,这就是 Visual Basic。1991 年微软公司推出了 Visual Basic 1.0 版,其功能相对较少,存在一定缺陷。但它是第一个可视化的编程工具软件,这在当时的业界引起了很大的轰动。在接下来的四年中,微软连续地推出了 Visual Basic 2.0、Visual Basic 3.0、Visual Basic 4.0 三个版本,并从 Visual Basic 4.0 开始引入了面向对象的程序设计思想,从而使 Visual Basic 成为业界最为流行的编程语言之一。1997 年微软发布了 Visual Studio 1.0,在该软件包中包含了 Visual Basic 6.0。Visual Basic 5.0 以前的各种版本,主要应用于 Windows 3.x 环境中的 16 位应用程序开发。Visual Basic 5.0 则是一个 32 位应用程序开发工具,可以运行在 Windows 95/98/2000 或 Windows NT 环境中。2000 年微软推出了 Visual Basic.NET 的测试版,并于 2002 年 3 月 22 日正式发布了 Visual Basic.NET 的中文版。

Visual Basic.NET 是在 Basic 和 Visual Basic 基础上发展起来的,是 Visual Basic 的全新版本,增加了很多新的内容,删除了某些传统的关键字,提高了类型安全性。因此,Visual Basic.NET 具有简单、现代和类型安全的特点,是一门完全面向对象的程序设计语言。

Visual Basic.NET 推出之后,版本不断升级,主要有 Visual Basic 2005、Visual Basic 2008、Visual Basic 2010,其后一般不再用 .NET 标记。

作为具有良好大众基础的 Visual Basic 的最新版本,很适合作为高校程序设计的首选语言。为此,本书编写时从零起点的角度用 VB.NET 语言进行编写,选择 Visual Basic 2008 为上机环境,期望能通过形象化、具体化的途径,达到易学好用的目的。让读者较轻

松地通过程序设计语言的学习,提升计算思维能力,为进一步从事程序设计打好较扎实的语言和应用基础。

1.2 几个重要概念

1. 对象

在客观世界中,人们处理问题都是面向对象的,对象是构成系统的基本单位。在实际社会生活中,人们都是在不同的对象中活动。

一个具体的杯子是一个对象,它的属性有:口径、型号和材质等,对它的操作(或者说它的行为)是盛水等;

一部具体的汽车也是一个对象,它的属性有:品牌、型号和排量等,对它的操作(或者说它的行为)是开动和转弯等;

一个具体的人同样也是一个对象,人的属性有性别、身高和体重等,人的行为(也可称为操作)有走路、吃饭、学习、工作等。

对象的基本特征分为静态特征和动态特征,我们不妨举两个具体的例子。

例 1: 学生在一个班级中上课、开会、开展社团活动和文体活动等。

这里的对象是班级,它的静态特征是:所属系、专业,学生人数,所在教室等;它的动态特征有:上课,开会,开展社团活动和文体活动等。

例 2: 我们所熟悉的计算机也是一个对象,它的静态特征(或者说属性)有: CPU、内存、硬盘、主板、显卡、声卡、键盘、鼠标、光驱等,它的动态特征(或者说行为)有打字、上网、游戏、编程、处理图像、听音乐、欣赏影视节目等。可以说,计算机的组成部件和计算机所做的各种事情共同描述了一部计算机。

2. 类

类则是一个抽象的概念,用来描述某一类对象所共有的、本质的属性和类的操作、行为。对象则是类的一个具体实现,又称为实例。

以杯子为例,它是描述这类对象共有的、本质的属性和操作、行为的抽象体,而大杯子和小杯子则是杯子类的某个实例,或者说是杯子类的具体对象。

类是具有共同特征的对象抽象,举例如下。

教师: 是肩负传道、授业、解惑重任的一类人;

学生: 是接受思想教育、道德教育、专业教育、人文教育的一类人。

教师和学生同属于人类,他们是人类的两个属性和行为各不相同的对象(也可称实例)。

对类的成员的访问级别可分为: 公有、私有和保护三级。若没有申明,类成员默认为私有。

类是一种数据结构,用 Class 标识。类可以封装数据成员、方法成员和其他的类。类又是创建对象的模块,VB.NET 所有的语句都必须包含在类内。因此,类是 VB.NET 语言的核心和基本构成模块。VB.NET 支持自定义的类,使用 VB.NET 编程,实质上是编

写自己的类来描述实际需要解决的问题。

类具有抽象性、隐蔽性和封装性的特征。

类的隐蔽性就体现在外界不能直接访问私有成员。

封装性使对象的数据得到了保护,所以说封装性是“面向对象”程序设计的重要特征。类是一个封装体,在其中封装了该对象的属性和操作。通过限制对属性和操作的访问权限,可以将属性“隐藏”在类的内部,公有方法作为对外的接口,在对象之外只能通过这一接口借助于对象对类的私有性成员进行具体的操作。

VB.NET 就是通过建立数据类型——类来支持封装和数据隐藏。封装性增加了对象的独立性,从而保证了数据的可靠性。一个定义完好的类,可以作为独立模块使用。

对象的属性和行为总是紧密联系在一起,属性用数据(即变量)来描述,行为则是数据的处理,要通过方法(或称函数)来实现。数据和对数据的处理,在面向过程程序设计中两者是分离的,而在面向对象的程序设计中两者是合一的,都封装在类体中。

封装性就是指将数据(变量)和数据处理(方法)都封装在类体内。我们可以理解为是把变量和相关的方法集中在一个有孔的容器中,只有在孔的边缘处的数据与方法才能与外界相通(这便是指所有的公有的成员),而其余的(指私有和保护的成员)均不受外界的影响,这个容器就是类。

在程序设计与实现中,程序设计方法正在从面向过程走向面向对象,使得编程语言与自然语言之间以及程序设计方法与实际解决问题方式之间的距离越来越近。这就意味着软件开发人员可以用更接近自然的思维方式、用更少的精力去完成同样的工作。

概括起来说,面向对象程序设计有如下优点:

- 与人类习惯的思维方式一致
- 可重用性好
- 可维护性好

正因为面向对象程序设计有众多的优点,所以今天程序设计方法逐步由面向程序设计发展为面向对象程序设计。

3. 命名空间

VB.NET 程序是利用命名空间组织起来的,即对具有相关功能的类在逻辑上进行分组,类似于存放相关物品的容器。例如一所大学中的人员模型可能包括以下几种类型:个人、学员、职员、教师、学生等。命名空间既用作程序的“内部”组织系统,也用作向外部其他程序公开自己拥有的程序元素的方法。

一般认为,命名空间有助于改善数据的构成,从而使每个使用者都可以轻松地获得想要的数据库。如果要调用某个命名空间中的程序元素,一般首先需要将命名空间所标识的命名空间内的成员导入当前编译程序中,从而可以直接使用每个被导入的类型的标识符。VB.NET 系统已经内置了常用的命名空间,编程时不需一一列出,只是在少数地方才需要增加,到时再作介绍。

4. 属性

属性是对现实世界中实体特征的抽象,它提供了一种对类或对象特性进行访问的机

制。例如,一个文件的大小、一个窗体的标题、一个控件的尺寸等都可以作为属性。属性所描述的是状态信息,在类的某个实例中,属性的值表示该对象相应的状态值。

5. 过程

为了适合编程的需要和解决代码的重复问题,高级语言中引入了过程的概念。从本质上讲,过程就是完成一定功能的程序段。该程序段有一个名字,称为过程名。从使用的角度来看,可以把过程分为 Sub 过程、函数过程等。

6. 模块

VB.NET 程序设计主要有控制台和窗体两种形式,两种方式都要应用模块(Module)结构来进行。当创建一个新的应用程序项目时,实际上是创建一个模块,在这个模块中包含了一个空白的过程,例如:

```
Module1           '模块定义
    Sub Main()     'Sub Main 过程定义
        :
    End Sub        'Sub Main 过程定义结束
End Module        '模块定义结束
```

应用程序项目的具体内容就添加在过程体内。

7. 标识符

在程序设计中会用到各种对象,如符号常量、变量、数组、过程和类型等,为了识别这些对象,必须给每一个对象一个名称,这个名称就称为标识符,所以,标识符是用于识别各种对象的字符,是用户定义的一种字符序列。标识符一般由单词或单词组合构成,也可以是一些无特定意思的字符或一串字符。其命名有三条原则:

- (1) 标识符只能由数字、字母和下划线组成,作为标识符的字符串内不能含有标点符号和%、&、!、#、@、\$、空格等字符;
- (2) 标识符必须以字母开头;
- (3) 标识符不能是系统内定的如 public、static、void 之类的关键字(见附录 D),不能与内部过程名、函数名相重。

注意:

- (1) VB.NET 语言标识符的命名不区分大小写。例如,NAME、Name 及 name 在 VB.NET 中被认为是同一个标识符。
- (2) 命名标识符时应尽量做到“见名知意”,即选有相应含义的英文单词或汉语拼音等作为标识符,如 Student(表示学生)、name(表示姓名)、gz(表示工资)等,以增加程序的可读性。

VB.NET 程序编写和运行,最简易的方法是借助于窗体使用可视化手段实现。窗体是很有代表性的 .NET 控件,任何一个控件都是对象,其多数属性和事件的用法对其他控件同样适用。

使用窗体编写 VB.NET 程序,涉及了 Windows 应用程序的创建,在此先作简单的介绍。

1.3 Windows 应用程序的创建

Windows 程序设计与传统的程序设计一个重大的区别就在于从过程驱动到事件驱动。

传统的 MS-DOS 程序主要采用顺序的、关联的、过程驱动的程序设计方法,是面向程序而不是面向用户的。Windows 程序设计是基于事件驱动的,程序的运行不是由事件的顺序来控制,而是由能触发的事件来控制,它是一种面向用户的程序设计方法。其中消息驱动机制是 Windows 程序设计的精髓。当单击鼠标、敲击键盘、窗口移动或改变大小等事件产生时,Windows 系统都会向特定的窗口发送消息。

Windows 的事件驱动与 MS-DOS 的过程驱动比较有何优越性呢?我们不妨从一个实例说起。

例如:计算女子体操运动员参加四项全能比赛的总成绩并排出名次。

MS-DOS 与 Windows 在处理该事件的过程比较如图 1-3 所示,左边是 MS-DOS 的处理过程,从图 1-3 中可以看出,它是按过程驱动、顺序处理的。在某运动员没有完成全部项目动作之前,无法计算总成绩并进行排序,更不要说同时进行另一个项目的角逐了。然而 Windows 的处理方式就不同了,它是按事件驱动的方式进行的,图的右边是它的消息队列,只要通过键盘输入任一运动员的任一个项目的成绩,都可以计算总成绩并进行排序,当然这仅是临时的排序,即使如此,运动员也可以根据这个临时的排序,制定下一步的比赛策略。不仅如此,运动员还可以同时进行不同项目的角逐,大大缩短比赛所用的时间。其处理过程之灵活、运行效率之高,显而易见。

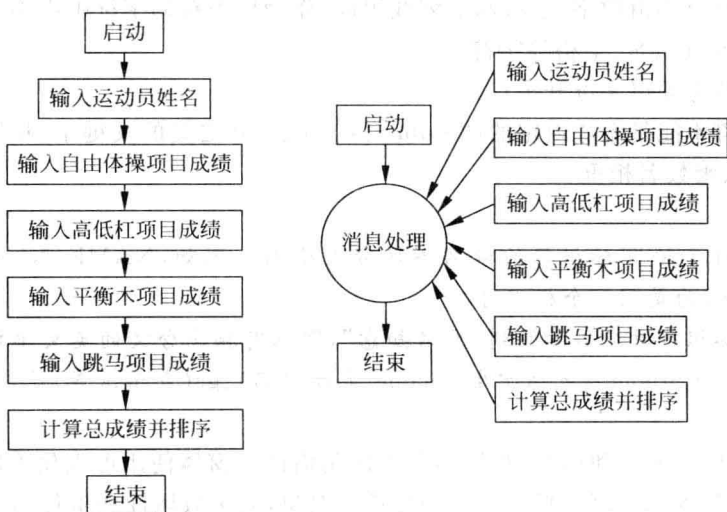


图 1-3 MS-DOS 与 Windows 处理同一事件过程比较

下面总结一下 Windows 和 MS-DOS 程序设计的区别。

1) MS-DOS 程序设计

顺序过程驱动的程序设计方法,程序有一个明显的开始,明显的顺序执行过程和明显的结束。

2) Windows 程序设计

Windows 程序设计是消息驱动的程序设计方法,可以实时随机输入,及时处理结果,完全不需要按顺序执行。

3) 事件驱动编程机制

基于 Windows 的应用程序设计方法采用的是事件驱动,通过事件的发生来进行控制。事件驱动程序设计是围绕着消息的产生和处理展开,消息是关于发生的事件的信息。

Windows 应用程序的消息来源有四种:

- 输入消息:包括键盘和鼠标的输入;
- 控制消息:用来与 Windows 的控制对象(列表框、按钮等)进行双向通信;
- 系统消息:对程序化的事件或系统时钟中断作出反应;
- 用户消息:是程序员自己定义并在应用程序中主动发出的消息。

作为一个 Windows 程序员,其工作任务就是对所开发的应用程序需要发出或需要接收的消息进行排序和管理。

事件驱动方法为程序设计提供了许多便利,对于那些需要用户大范围干预的应用程序即我们常说的“工程”或“应用系统”来说,更显露出其优越性。

1.4 VB.NET 程序的上机环境—— Visual Studio 2008 开发环境简介

按照如下顺序打开 Visual Studio 2008 开发环境。在“开始”菜单中依次单击“所有程序”|“Microsoft Visual Studio 2008”|“Microsoft Visual Studio 2008”命令即可,如图 1-4 所示。



图 1-4 Visual Studio 2008 开发环境