



21 世纪高职高专规划教材
高等职业教育规划教材编委会专家审定

VB.NET CHENGXU SHEJI
XIANGMU JIAOCHENG

VB.NET 程序设计项目教程

主 编 王平华
副主编 周学军 孔 璐 胡菊芬



北京邮电大学出版社
www.buptpress.com

21 世纪高职高专规划教材

高等职业教育规划教材编委会专家审定

VB.NET 程序设计项目教程

主 编 王平华

副主编 周学军 孔璐 胡菊芬

北京邮电大学出版社

• 北 京 •

内 容 简 介

本教材是根据作者多年从事高职计算机软件教学经验和实际项目开发体会为基础,以培养学生实际动手能力为目标,采用“目标和任务驱动法相结合”的理论,从提出任务目标到任务目标的解决都以简明生动的语言,深入浅出地进行阐述。教材中采用的案例多数来自实际项目,案例完整,使读者根据每节提出的学习目标在案例中轻松学习。全书以 VS. NET2005 为蓝本,将 VB. NET 程序设计内容分为 11 章,包括 VS. NET2005 概述、项目中基础知识、项目中类应用开发、项目中控件应用开发、项目中多线程应用开发、项目中多媒体应用开发、项目中文件应用开发、项目中网络应用开发、项目中数据库应用开发、项目中打印与水晶报表应用开发和项目中应用程序发布。

本书适合于计算机专业的高职、专科或本科学生学习,也适合作为从事 VB. NET 开发人员的参考用书。

图书在版编目(CIP)数据

VB. NET 程序设计项目教程/王平华主编. —北京:北京邮电大学出版社,2008

ISBN 978-7-5635-1863-0

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

中国版本图书馆 CIP 数据核字(2008)第 162527 号

书 名: VB. NET 程序设计项目教程

主 编: 王平华

责任编辑: 王志宇

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt. edu. cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 22.75

字 数: 567 千字

印 数: 1—3 000 册

版 次: 2008 年 10 月第 1 版 2008 年 10 月第 1 次印刷

ISBN 978-7-5635-1863-0

定 价: 38.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

VB. NET 是 Microsoft. NET 系列中一门强大的、现代的、纯面向对象的开发工具,它为程序员提供了快速的开发和发布平台。

本书根据作者多年从事高职软件技术专业教学 and 实际项目开发的体会编写而成。全书对管理信息系统项目中可能用到的知识和技术作了系统的介绍,共 11 章,分别对项目中运用的技术进行了讲解,主要内容包括:VS. NET2005 概述、项目中基础知识、项目中类应用开发、项目中控件应用开发、项目中多线程应用开发、项目中多媒体应用开发、项目中文件应用开发、项目中网络应用开发、项目中数据库应用开发、项目中打印与水晶报表应用开发和项目中应用程序发布等,本书中使用的语法都参考使用了微软 VB. NET2005 的 MSDN。

本书最大的特色是作者用实际项目的开发经验作基础,以任务驱动引导读者学习,本书有如下特点:

1. 任务驱动法

本书从第 2~10 章都采用了任务驱动法编写。首先提出任务目标;其次对任务目标进行分析;再展示任务目标的实现过程;最后讲解与任务目标相关的知识。

2. 实用性

本书的作者具有多年从事 .NET 实际项目的开发经验,先后主持完成了十几个管理信息系统。教材中出现的案例都是实际项目中的部分模块,程序的编写和实际项目开发一致,符合纯面向对象设计的要求。

3. 循环法

教材一般都是先介绍知识点,再讲相关应用,最后得到应用结果。而本书与传统的写法恰恰相反。不过,本书还可以从每节的任务相关知识先读,再读任务实现过程,最后形成任务目标。这样,符合习惯看传统教材的读者。看完后再遵循逆序法读本教材,就形成了循环法读书,使读者更容易接受理解本书所讲知识。

另外,本书还配有电子教案,需要的读者请与出版社联系。

本教材由王平华完成了第 2~9 章的编写,周学军、孔璐、胡菊芬分别完成了第 1 章、第 10 章、第 11 章的编写。编写本书时,还得到了单文增、胡永超、曾海峰在排版和课件方面的大力支持,在此表示衷心感谢。

由于时间仓促及水平有限,书中难免存在错误和不妥之处,敬请读者批评指正。

目 录

第 1 章 VS.NET2005 概述

1.1 VB.NET 的发展历程	1
1.2 .NET 基本概念和平台组成	2
1.3 .NET 框架	2
1.3.1 .NET 公共语言运行环境	3
1.3.2 .NET 框架基础类库	3
1.3.3 面向 Windows 的 .NET 编程	3
1.3.4 面向网络应用的 .NET 编程	3
1.4 VS.NET 开发环境	4
习题	6

第 2 章 项目中基础知识

2.1 计算圆的面积	7
2.2 求 $1! + 2! + \cdots + n!$ 之和	17
2.3 实现两个整数变量中的数值交换	28
2.4 对一组整数进行升序排序	32
习题	35

第 3 章 项目中类应用开发

3.1 面向对象和类的基本概念	36
3.2 项目中类的方法、事件、属性的应用开发	37
3.2.1 类的定义	37
3.2.2 类的三大特征	47
3.2.3 抽象类	55
3.2.4 最终类	58
3.3 项目中接口的应用开发	62
习题	66

第 4 章 项目中控件应用开发

4.1 项目中常用控件应用开发	67
4.1.1 医院收费系统登录界面设计	67
4.1.2 我的资源管理器	90

4.1.3 答题测试面板	114
4.2 项目中对话框应用开发	123
习题	134
第 5 章 项目中多线程应用开发	
5.1 简单的多线程应用程序开发	135
5.2 同步多线程应用程序开发	141
习题	153
第 6 章 项目中多媒体应用开发	
6.1 多媒体播放器应用开发	154
6.2 画图程序应用开发	159
习题	176
第 7 章 项目中文件应用开发	
7.1 记事本程序应用开发	177
7.2 通讯录程序应用开发	211
习题	231
第 8 章 项目中网络应用开发	
8.1 P2P 的聊天程序应用开发	232
8.2 简易浏览器程序应用开发	246
习题	249
第 9 章 项目中数据库应用开发	
9.1 医院药品信息管理应用开发	250
9.2 对象在数据库中存取应用开发	318
习题	323
第 10 章 项目中打印与水晶报表应用开发	
10.1 医院门诊发票打印应用开发	324
10.2 水晶报表应用开发	343
第 11 章 项目中应用程序发布	
11.1 应用程序的部署及发布概述	349
11.2 药品信息管理程序的部署及发布	349
参考文献	358

第 1 章 VS.NET2005 概述

Microsoft.NET 作为新一代的程序运行平台,在应用程序开发领域中开创了一个新时代。VB.NET 语言是一种简单但功能强大的编程语言,用于编写企业应用程序。它从 QBasic 语言演化而来。

通常所说的.NET,指的是.NET Framework,它管理和支持.NET 程序的执行;而 Visual Studio.NET 则是目前最优秀的应用程序集成开发环境之一,为 VB.NET 应用程序提供了极大的便利。

学习目标

- VB.NET 的发展历程
- VS.NET 的开发环境

1.1 VB.NET 的发展历程

VB.NET 的全称是 Visual Basic.NET,也称为 Visual Basic 8.0,Visual Basic(VB)是一种由微软公司开发的包含协助开发环境的事件驱动编程语言。从任何标准来说,VB 都是世界上使用人数最多的编程语言之一——不管是盛赞 VB 的开发者还是抱怨 VB 的开发者。它源自于 BASIC 编程语言。VB 拥有图形用户界面(GUI)和快速应用程序开发系统,可以轻易的使用 ADO.NET 连接数据库,或者轻松的创建 ActiveX 控件。程序员可以轻松的使用 VB 提供的组件快速建立一个应用程序。

VB.NET 是由最早的 BASIC 语言发展过来的,从 VB 1.0 到 VB 8.0 的发展大概经历了以下几个阶段:

- (1) 1991 年 4 月,Visual Basic 1.0 Windows 版本发布。
- (2) 1992 年 9 月,Visual Basic 1.0 DOS 版本发布。
- (3) 1992 年 11 月,VB 2.0 发布。它对上一个版本的界面和速度都有所改善。
- (4) 1993 年,VB 3.0 发布,分为标准版和专业版。其中包含一个数据引擎,可以直接存取 Access 数据库。
- (5) 1995 年 8 月,VB 4.0 发布了 32 位版本和 16 位的版本。其中包含了对类的支持。
- (6) 1997 年 2 月,VB 5.0 发布。程序员可以用 32 位的版本转入由 4.0 版本创建的 16 位程序,并且能顺利编译。同时还包含了对用户自定义控件的支持。

- (7) 1998 年,VB 6.0 发布。
- (8) 2001 年,Visual Basic. NET 和. NET Framework 发布。
- (9) 2003 年,Visual Basic. NET 2003 和. NET Framework 1.1 发布。
- (10) 2004 年,微软开放了 Visual Studio .NET 2005 的测试版本(代号 Whidbey)。包含了. NET Framework 2.0 的测试版本。
- (11) Visual Studio. net 2005 在 2005 年 11 月 7 日发布。

1.2 .NET 基本概念和平台组成

Microsoft. NET 是微软开发的平台,是一种面向网络、支持各种用户终端的开发平台环境。Visual Studio. NET 是 Microsoft 为实现其. NET 技术而开发的一整套工具组件。它把开发功能强大、性能可靠的企业网络解决方案进行了简化。

利用 Visual Studio. NET,用户可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。

1.3 .NET 框架

.NET 框架包括以下组件:

- (1) 公共语言运行环境(CLR)
- (2) NET 框架类库(BCL)
- (3) 数据库访问组件(ADO. NET 和 XML)
- (4) 基于 ASP. NET 编程框架的网络服务(Web Services)和网络表单(Web Forms)
- (5) Windows 桌面应用界面编程组件(Win Forms)

图 1-1 刻画了. NET 框架的整体结构。

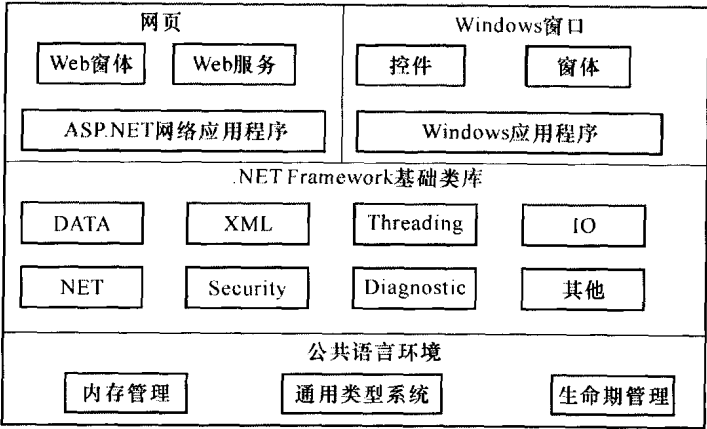


图 1-1 .NET 框架的整体结构

1.3.1 .NET 公共语言运行环境

VB.NET2005 和微软公司的其他开发语言共用一个称为 VS.NET2005 的集成开发环境,这些语言包括 C#.NET、C++.NET 等,它们共享同一个 .NET Framework 框架。这些语言都可以共享 Framework 中的类。有了 .NET Framework 就可以建立 .NET 中的公共语言运行环境,公共语言运行环境(Common Language Runtime,CLR)主要是为了解决各种不同编程语言之间相互调用同一个程序的问题。CLR 的管理工作包括以下几项:

(1) 确定加载代码的方式和时机,并管理对象在内存中的分配。代码执行时,CLR 将决定代码所需要的类和方法,并且按照需要将原先编译过的中间语言(IL)进行即时编译(JIT)。

(2) 对托管所占用的内存进行管理,通过垃圾收集器(GC)实现无用内存的回收。

(3) 通过使用基于异常处理机制的通用错误处理方式,对托管代码中的错误进行处理和传递。

(4) 维护 CLR 和应用程序的安全性。在 .NET 框架中,安全性有两种形式:代码访问的安全性和角色的安全性。前者用于确保在安全的上下文中执行代码,后者用于控制对系统资源的访问。

1.3.2 .NET 框架基础类库

通常各种程序语言都有自己的函数库,函数库里主要存放一些预先定义好的、实现一些常用功能的函数,这样用户编程时,可以直接调用函数库里的函数。面向对象的编程语言的函数库是由一些预先定义好的类构成的。函数和类的功能是一致的,比如 C 语言中输入输出功能是通过函数实现的。对于不同的编程语言,函数库(类库)的内容和存取方式都各不相同,这就给不同的编程语言之间相互调用造成了巨大的障碍。

.NET 框架的基础类库就是为了解决这个问题而建立的公共类库(函数库)。各种符合公共语言规范的编程语言都可以调用其中的内容。基础类库将系统的基本功能,如网络存取、通信协议、窗口对象等分成几个基本类,程序语言可以直接调用基本类,也可以在基本类的基础上,增加一些功能,建立新的类(这就叫做继承)来调用。

1.3.3 面向 Windows 的 .NET 编程

.NET 中提供了快捷的 Windows 窗体平台应用程序的构建。用户只需要利用 VS.NET 中提供的开发工具栏中的控件就可以构建 Windows 平台,如果用户觉得这些控件无法满足要求,可以利用 .NET 中的自定义用户组件自己创建以满足要求。

1.3.4 面向网络应用的 .NET 编程

ASP.NET 是关于如何开发网络应用程序,制作网页的一个规范,具体的网络应用程序可以用 VB.NET、C++、JavaScript 等编程语言来编写。

ASP.NET 的功能很强大,远远优于传统的 ASP。ASP.NET 的一大优点是它的执行

速度快。ASP.NET 是编译后执行的,当 ASP.NET 文件第一次被请求时,系统编译这个文件,以后的请求就不需要再重新编译了。所以第一次请求 ASP.NET 文件时等待时间比较长,以后再次调用该文件时等待时间就很短了。当然,如果请求的 ASP.NET 文件被修改过,就要重新编译了。

1.4 VS.NET 开发环境

安装好 VS.NET2005 的安装程序后,在“开始”的“程序”菜单中找到“Microsoft Visual Studio 2005”后单击,就可以启动 VS.NET 的开发环境,启动后就会出现如图 1-2 所示 VS.NET 开发环境的界面。

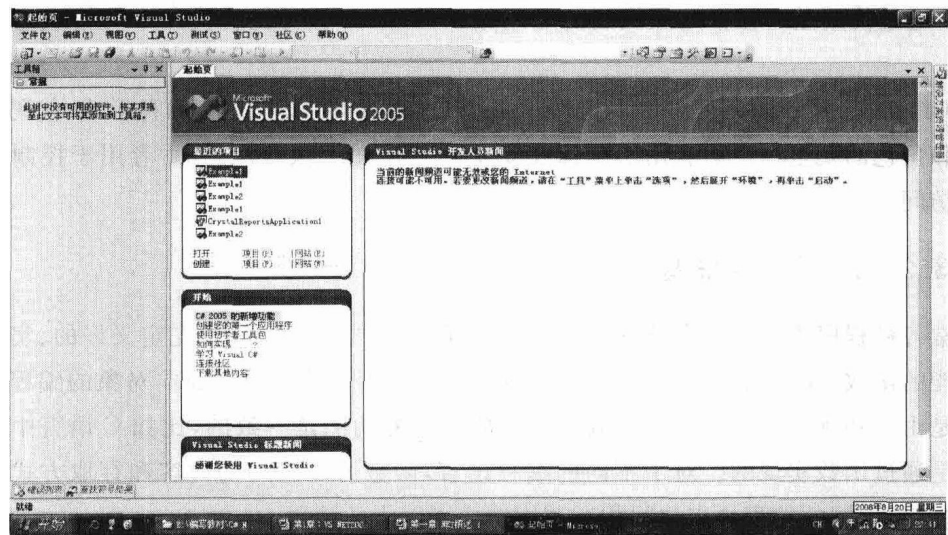


图 1-2 VS.NET 开发环境界面

在这里,我们看到的是 VS.NET2005 开发环境,进入正式开发前,需要选择 VB.NET 语言作为开发工具,只要在新建工程时确定开发语言即可。选择“文件”菜单中的“新建项目”后,会出现如图 1-3 所示的新建项目窗口。

在“新建项目”中选择“Visual Basic”语言,然后选择右边的项目模板即可。假定选择“Windows 应用程序”,确定后出现如图 1-4 所示的 VS.NET 集成开发环境界面。

在图 1-4 所示的集成开发环境中,在最上面的菜单中提供了开发时所需的各种命令指令及一些开发向导操作。“工具栏”上的按钮提供了开发中一些常用的快捷操作。

“工具箱”中列出了所有 VS.NET 中的可用控件。所谓的控件是构建窗体所需的组件,有的能够在运行时显示出来,有的则不会,比如 Button、Label 等会在窗体上显示,而 Image List 则不会。在工具箱上共分了“所有 Windows 窗体”、“公共控件”、“容器”、“菜单和工具栏”、“数据”、“组件”、“打印”、“对话框”、“Crystal Report”和“常规”等 10 项,分类存放这些编程中需要使用的控件。如果工具栏上没有编程中需要使用的控件,可以通过鼠标右击弹

出菜单中的“添加选择项”，根据向导进行添加。

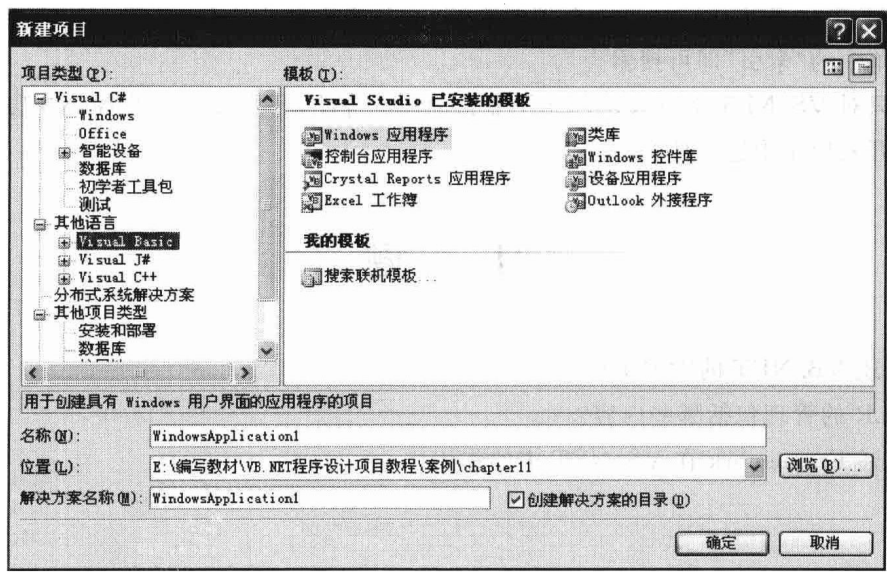


图 1-3 新建项目窗口

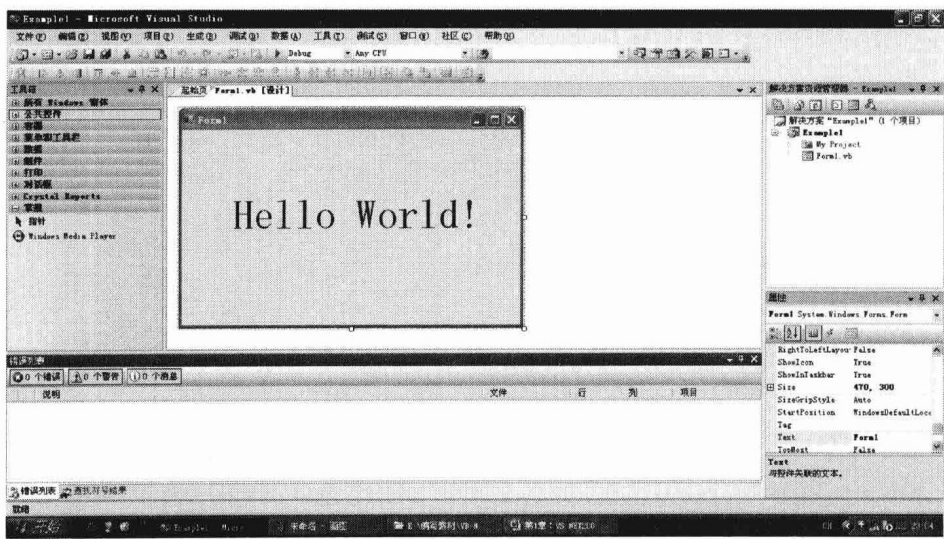


图 1-4 VS.NET 集成开发环境界面

在图 1-4 中的最右边,有一个“解决方案资源管理器”的视图窗口,在该窗口中可以对当前项目中所需要的各种文件进行管理,例如增加文件或删除、打开等操作。资源管理器的视图窗口下面有“属性窗口”,通过该窗口,可以完成对选中的控件进行相关属性值的更改,以便达到设计所需。

图 1-4 的最下面,有一个“错误列表”视图窗口,该窗口的作用是将系统进行编译中出现的一些错误信息列出。该错误信息分为“错误”、“警告”和“消息”3 种,程序员可以根据这些信息进行程序修改后再编译。

任何一本教材或参考书不可能将项目中涉及的开发工具中的知识都讲全,我们可以通过 VS. NET 中自带的帮助文档查询到所需知识,只需安装 VS. NET 的 MSDN 即可,选中“帮助”菜单中的“索引”即可搜索查询。

本章只对 VS. NET 的开发环境作了非常简单的介绍,开发环境中的各部分功能需要在具体项目开发应用中进行体会。

习 题

1. 简述 VB. NET 的发展历史。
2. CLR 的管理包括哪些内容?
3. 简述 Framework 在 VS. NET 中的作用。

第2章 项目中基础知识

任何一门语言都是由基本语法、规格等知识组合而成,VB.NET也不例外,要想利用VB.NET开发项目,必须熟练掌握其基本语法、数据类型、程序控制流程等各种基础知识,本章通过案例引导读者学习这些知识。

学习目标

- 变量、常量的定义,数据类型的使用
- VB.NET 中的各种运算符
- VB.NET 中的数组的定义及使用
- VB.NET 中的过程定义及使用

2.1 计算圆的面积

通过一个计算圆的面积的简单程序学习掌握有关变量、常量的定义,数据类型的使用。

1. 任务目标

利用VB.NET中的控制台应用程序,创建一个能够计算圆的面积的程序,其中圆的半径可以随意输入,并且能够对输入错误的半径程序进行处理,如图2-1所示。

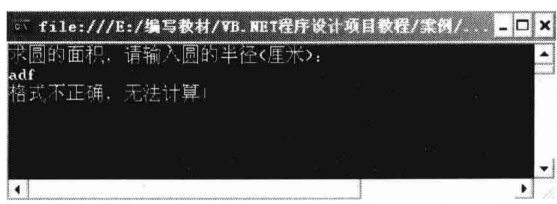


图 2-1 错误半径输入

输入正确的半径后,程序能够正确计算出圆的面积,如图2-2所示。

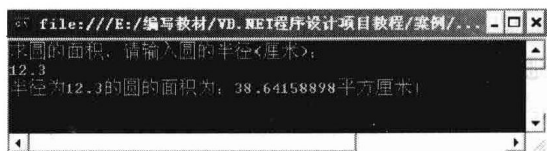


图 2-2 计算圆的面积

2. 任务分析

根据任务目标可以看出,程序的类型首先可以确定为控制台程序,这在 VB.NET 中有明确的应用程序类型,只需要作出选择即可。程序的任务非常明确,是计算圆的面积,圆的面积公式为:圆的面积=圆的半径的平方 $\times$ 圆周率,其中圆的半径是可变的,也就是说半径是不确定的,由用户随机输入;而圆周率是固定的,即约为 3.141 592 6。这是我们要清楚的第二个问题。第三个问题是需要对用户随机输入的字符进行检测,检测其是否符合圆的半径,符合才能计算圆的面积,不符合要求向用户告之。最后要求输出计算结果。

根据任务提出的目标,整个任务的实现需要以下代码完成:

```
Module Module1
    ''' <summary>
    ''' 利用控制台程序求圆的面积
    ''' </summary>
    ''' <remarks></remarks>

    Sub Main()
        Console.WriteLine("求圆的面积,请输入圆的半径(厘米):") '输出提示信息
        Static Dim PI As Double = 3.1415926 '定义常量
        Dim r As Double = 0 '定义存储圆的半径变量 r
        '获取用户的半径输入,并对输入的值进行异常格式检测
        Try
            r = CType(Console.ReadLine(), Double)
        Catch ex As InvalidCastException '捕获无效格式转换异常
            Console.WriteLine("格式不正确,无法计算!") '输出错误提示信息
            Console.Read() '使屏幕停止,按任意键结束
            Exit Sub '退出当前方法,终止程序继续执行
        End Try
        Console.WriteLine("半径为" & r & "的圆的面积为:" & (r * r * PI) & "平方厘米!")
        '输出圆的面积
        Console.Read() '使屏幕停止,按任意键结束
    End Sub
End Module
```

3. 任务相关知识

任务完成比较简单,在完成任务中,使用了许多没有学习过的新内容,下面就这些内容及相关知识做详细的介绍。

(1) Console 类

Console 是在控制台程序中才能使用的一个类,表示控制台应用程序的标准输入流、输出流和错误流。案例中通过该类(有关类的学习第 3 章作了详细介绍)的静态方法 WriteLine 输出信息,ReadLine 读取键盘信息等。该类的功能是通过它的属性和方法来完成。

1) Console 属性

① BackgroundColor: 获取或设置控制台的背景色。属性值为一个 ConsoleColor, 指定

控制台的背景色;也就是显示在每个字符后面的颜色。默认为黑色。ConsoleColor 值是由各种颜色组成的集合。

② CapsLock: 获取一个值, 该值指示 CapsLock 键盘切换键是打开的还是关闭的。True 表示打开, False 表示关闭。

③ CursorVisible: 获取或设置一个值, 用以指示光标是否可见。True 表示可见, False 表示隐藏。

④ Error: 获取标准错误输出流。只读属性。

⑤ In: 获取标准输入流。

⑥ ForegroundColor: 获取或设置控制台的前景色。属性值为一个 ConsoleColor, 指定控制台的背景色; 也就是显示在每个字符后面的颜色。默认为黑色。ConsoleColor 值是由各种颜色组成的集合。

⑦ InputEncoding: 获取或设置控制台用于读取输入的编码。属性类型为 Encoding, 属性值用于读取控制台输入的编码。

⑧ NumberLock: 获取一个值, 该值指示 NumLock 键盘切换键是打开的还是关闭的。True 表示打开, False 表示关闭状态。

⑨ Title: 获取或设置要显示在控制台标题栏中的标题。属性值为字符串型。

2) Console 方法

① Beep: 通过控制台扬声器播放提示音。

语法格式定义:

```
Public Shared Sub Beep
```

② Clear: 清除控制台缓冲区和相应的控制台窗口的显示信息。

语法格式定义:

```
Public Shared Sub Clear
```

③ Read: 从标准输入流(用户输入)读取下一个字符。该方法的读取是以 ASCII 码返回。

语法格式定义:

```
Public Shared Function Read As Integer
```

说明: 返回值是指输入流中的下一个字符; 如果当前没有更多的字符可供读取, 则为 -1。

④ ReadLine: 从标准输入流(用户输入)读取下一行字符。

语法格式定义:

```
Public Shared Function ReadLine As String
```

说明: 读取的内容以字符串返回。

⑤ ResetColor: 将控制台的前景色和背景色设置为默认值。

语法格式定义:

```
Public Shared Sub ResetColor
```

⑥ Write: 将指定值的文本表示形式写入标准输出流。可以输出 VB.NET 中所有的基本类型数据。

语法格式定义:

```
Public Shared Sub Write( value As 基本数据类型 )
```

说明: 该方法有多个重载方法, 在第 3 章会讲述方法的重载。

⑦ WriteLine:将指定的输出内容按行输出。可以输出 VB.NET 中所有的基本类型数据。

语法格式定义:

```
Public Shared Sub WriteLine(value As 基本数据类型)
```

说明:该方法有多个重载方法,在第 3 章会讲述方法的重载。

(2) 常量

编写程序的过程中,需要使用到一些固定不变的数据,如求圆的面积时,圆周率是一个固定的值,这个值不会随程序的运行而改变,也就是说在程序的运行期间和设计期间都不会改变该值,这就是常量。常量有时也叫做常数,它是指在程序运行期间不会改变的量,可以是任何数据类型。使用常量增加了程序代码的可读性和可维护性。实际上常量就是将一个数据存储在特定名字的内存中,而且这个内存中的数据在运行的过程中不会发生改变,使用这个特定名字时就相当于使用了这个内存中的数据。

根据 VB.NET 的规定,常用的定义语法的格式如下:

```
[修饰词] Const 常量名 As 数据类型 = 初始值
```

说明:修饰词是限制访问权限,一般有 Public、Private、Protected 等。常量名是由一组字符串组成,这个字符串要遵循 VB.NET 中的标识符命名规则。数据类型是指 VB.NET 中的基本数据类型,如 Integer、Double 等。初始值是对常量进行赋值即初始化,必须在定义时进行赋值,否则会出错。

标识符是指在程序设计的过程中用来存储值的一组字符串。标识符命名必须遵循以下规则:

- 标识符不能是 VB.NET 中的关键字,如 IF、Case、For 等。
- 标识符必须以字母开头,即标识符的第一个字符必须是 A~Z 的 26 个字母中的一个,也可以是小写,VB.NET 中标识符不区分大小写。
- 标识符的组成可以是字母、数字及下划线(“_”)的任意组合。

通常,常量的标识符用大写字母,以便区分变量的标识符。这是一个编程习惯,在 VB.NET 中并没有硬性规定。

另外,在 VB.NET 中,声明常量除了用 As 数据类型的方式外,有部分类型还可以使用 VB.NET 中约定的符号来替代 As 数据类型,具体的替代符号请参考表 2-1。

表 2-1 常量替代符

约定符号	指定数据类型	约定符号	指定数据类型
%	整型(Integer)	#	双精度浮点型(Double)
&	长整型(Long)	@	小数型(Decimal)
!	单精度浮点型(Single)	\$	字符串型(String)

表中列出了 6 种可替换的类型,其他类型没有相应的符号。

例如:案例中定义 PI=3.1415626,根据表 2-1 就可以改成:

```
Const PI# = 3.1415926
```

(3) 变量

在程序运行期间,需要保存一些值,如求圆的面积中的半径,它要存储用户输入的半径

值,这些值在程序运行前是无法预知的。而这些保存的值就需要变量来存储。变量就是在程序运行期间,它的值根据程序需要可以改变的量。这一点和常量恰恰相反,也是变量和常量的主要区别。

“变量”具有名称(用来引用变量所含值的词)和数据类型(它确定变量可存储数据的类型)。将一个数据存储在变量中,称为赋值,在定义变量时就进行赋值称为变量的初始化。

变量的定义:

`Dim 变量名 As 数据类型 [= 变量初始化值]`

说明:变量名是由一组字符串组成,符合标识符的命名规则。数据类型是指 VB. NET 中的普通数据类型或复合数据类型。变量初始化值可选项,即对变量进行赋值。

使用 Dim 的方式定义变量,一般是定义在某个方法内部,作为局部变量使用。如果需要定义模块级变量,如类的成员变量,就可以使用以下格式:

`[Public/Private] 变量名 As 数据类型 [= 变量初始化值]`

说明:Public 和 Private 是变量的访问权限,在第 3 章有详细的讲解。其他用法和 Dim 一样。

另外,使用 Dim 语句定义变量时可以同时定义多个变量,变量名中间用逗号隔开;如果各个变量的数据类型不一样,可以使用 As 分开指定各自变量的数据类型。例如:

`Dim x,y as Integer` ‘连续定义了 x,y 两个整型变量

`Dim i as Integer,str as String` ‘连续定义了 i 整型、str 字符串型的两个变量

在连续定义的多个变量中,不能在变量定义时进行数据的初始化。另外,在同一行中定义多个变量时可以对变量进行初始化,中间用逗号隔开。例如:

`Dim x as Integer = 3,y as Integer = 4` ‘同一行中定义了 x,y 变量且进行了变量初始化和常量定义一样,有时为了方便,也可以使用表 2-1 中的约定符来替代数据类型。

例如:

`Dim x % = 3` ‘定义了 Integer 变量,且进行了初始化

在定义变量时注意变量名的命名,除了遵循标识符的命名规则外,最好对变量名的命名做到望文知义,虽然 VB. NET 没有作出明文规定,但这样的做法提高了程序的可读性。

有时,需要一个变量保存的值在程序运行结束后该值继续存在,以便另外程序调用时继续使用保存的数值,这时需要静态变量来完成这一任务。静态变量是相对 Dim 的动态变量而言。Dim 定义的变量一般都在某个过程中作为局部变量使用,寿命周期比较短,而静态变量生命周期比较长,如在模块的某个方法中定义一个静态变量,该静态变量不会随着方法调用结束而生命周期终止,这时的生命周期又和整个模块的生命周期一样。静态变量的定义格式:

`Static 变量名 As 数据类型`

这里的格式说明和 Dim 定义变量类似,只是将 Dim 替换成 Static。例如:

`Static x as Integer` ‘定义一个静态整型变量

另外,变量命名时尽可能遵循以下规则:

- 定义目标明确的变量,每一个变量不要有多种意义。
- 为变量赋予意义比较强的名字,如 strName,一看就知道是字符串型的名字变量。
- 在变量中混合使用大小写字母。