

序 摇 摇 言

摇摇这套教材是面向 21 世纪计算机学科系列教材。为什么要组织这套教材？根据什么编写这套教材？这些都是在这篇序言中要回答的问题。

计算机学科是一个飞速发展的学科，尤其是近十年来，计算机向高度集成化、网络化和多媒体化发展的速度一日千里。但是，从另一个方面来看，目前高等学校的计算机教育，特别是教材建设，远远落后于现实的需要。现在的教材主要是根据《教学计划 1985》的要求组织编写的。这个教学计划，在制定过程中主要参照了美国 1984 年和 1985 年的《教学计划 1985》。

近年来，计算机学科已有了长足发展，这就要求高等学校计算机教育必须跟上形势发展的需要，在课程建设和教材建设上做出相应调整，以适应面向 21 世纪计算机教育的要求。这是组织这套教材的初衷。

为了组织好这套教材，全国高等学校计算机教育研究会课程与教材建设委员会在天津召开了“全国高等学校计算机学科课程与教材建设研讨会”，在北京召开了“教材编写大纲研讨会”。在这两次会议上，代表们深入地研讨了全国高校计算机专业教学指导委员会和中国计算机学会教育委员会制定的《计算机学科教学计划 1993》以及美国 1984 年和 1985 年的《计算机学科教学计划 1985》，这是这套教材参照的主要依据。

1984 年和 1985 年的《计算机学科教学计划 1985》是在总结了从《计算机学科教学计划 1985》到现在，计算机学科十年来发展的主要成果的基础上诞生的。它认为面向 21 世纪计算机学科应包括 5 个主科目，其中 3 个主科目为核心主科，它们是：算法与分析（1）、体系结构（2）、离散结构（3）、计算科学（4）、图形学、可视化、多媒体（5）、网络计算（6）、人机交互（7）、信息管理（8）、智能系统（9）、操作系统（10）、程序设计基础（11）、程序设计语言（12）、软件工程（13）、社会、道德、法律和专业问题（14）。其中除 4 和 5 为非核心主科目外，其他 10 项均为核心主科目。

将 1993 年教学计划与 1985 年教学计划比较可看出：

（1）在 1985 年计划中，离散结构只作为数学基础提出，而在 1993 年计划中，则作为核心主科目提出，显然，提高了它在计算机学科中的地位。

（2）在 1985 年计划中，未提及网络计算，而在 1993 年计划中，则作为核心主科目提出，以适应网络技术飞速发展的需求。

（3）图形学、可视化与多媒体也是为适应发展要求新增加的内容。

除此之外，1993 年计划在下述 4 个方面做调整：

将程序设计语言引论调整为程序设计基础，将人-机通信调整为人机交互，将人工智能与机器人学调整为智能系统，将数据库与信息检索调整为信息管理，将数值与符号计算调整为计算科学。

显然，这些变化使 1993 年计划更具有科学性，也更好地适应了学科发展的需要。

在组织这套教材的过程中，充分考虑了这些变化和调整，在软件和硬件的课程体系、界面划分方面均做了相应的调整，使整套教材更具有科学性和实用性。

另外，还要说明一点，教材建设既要满足必修课的要求，又要满足限选课和任选课的要求。

因此 教材应按系列组织 反映整个计算机学科的要求 采用大拼盘结构 以适应各校不同的具体教学计划 使学校可根据自己的需求进行选择。

这套教材包括 :《微机应用基础》、《离散数学》、《电路与电子技术》、《电路与电子技术习题与实验指南》、《数字逻辑与数字系统》、《计算机组成原理》、《微机接口技术》、《计算机体系结构》、《计算机网络》、《计算机网络实验教程》、《通信原理》、《计算机网络管理》、《网络信息系统集成》、《多媒体技术》、《计算机图形学》、《计算机维护技术》、《数据结构》、《计算机算法设计与分析》、《计算机数值分析》、《汇编语言程序设计》、《~~汇编语言~~程序设计》、《~~汇编语言~~程序设计》、《~~汇编语言~~程序设计》、《~~汇编语言~~程序设计》、《操作系统原理》、《~~操作系统~~操作系统原理与应用》、《~~操作系统~~操作系统》、《软件工程》、《数据库系统原理》、《编译原理》、《编译方法》、《人工智能》、《计算机信息安全》、《计算机图像处理》、《人机交互》、《计算机伦理学》。对于 ~~陈桂云~~和 ~~粤语~~的《计算机学科教学计划 ~~圆国~~》中提出的 ~~员~~个主科目 这套系列教材均涵盖 能够满足不同层次院校、不同教学计划的要求。

这套系列教材由全国高等学校计算机教育研究会课程与教材建设委员会主任李大友教授精心策划和组织。编者均为具有丰富教学实践经验的专家和教授。所编教材体系结构严谨、层次清晰、概念准确、论理充分、理论联系实际、深入浅出、通俗易懂。

教材组织过程中 得到了哈尔滨工业大学蒋宗礼教授 西安交通大学董渭清副教授 武汉大学张焕国教授 吉林大学张长海教授 福州大学王晓东教授 太原理工大学余雪丽教授等的大力支持和帮助 在此一并表示衷心感谢。

李大友
圆国圆年 远月

目 录

第 1 章 概述	(1)
1.1 灾难的发展过程和特点	(1)
1.2 灾难的发展过程	(1)
1.3 灾难的特点	(2)
1.4 灾难集成开发环境	(2)
1.5 启动灾难	(3)
1.6 灾难集成开发环境	(3)
1.7 面向对象的程序设计	(4)
1.8 面向对象程序设计的概念	(4)
1.9 灾难中的对象和类	(5)
1.10 对象的属性、事件及方法	(5)
1.11 灾难程序设计的基本步骤	(5)
1.12 灾难程序设计的一般过程	(5)
1.13 灾难程序设计的基本步骤	(5)
1.14 灾难程序的结构	(5)
本章小结	(5)
习题一	(5)
第 2 章 程序设计基础	(6)
2.1 基本数据类型	(6)
2.1.1 数值型数据	(6)
2.1.2 逻辑型数据	(6)
2.1.3 日期型数据	(6)
2.1.4 字符串型数据	(6)
2.1.5 变体型数据	(6)
2.1.6 对象型数据	(6)
2.2 常量与变量	(6)
2.2.1 常量	(6)
2.2.2 变量	(6)
2.3 运算符与表达式	(6)
2.3.1 算术运算符与算术表达式	(6)
2.3.2 字符串运算符与字符串表达式	(6)
2.3.3 关系运算符与关系表达式	(6)
2.3.4 逻辑运算符与逻辑表达式	(6)
2.3.5 运算符的优先级	(6)
2.4 常用标准函数	(6)

摇摇摇摇圆源圆源数学函数	(猿)
摇摇摇摇圆源圆源字符串函数	(猿)
摇摇摇摇圆源圆源日期与时间函数	(源)
摇摇摇摇圆源圆源数据类型转换函数	(源)
摇摇摇摇圆源圆源格式输出函数	(源)
摇摇摇摇圆源圆源其他函数	(源)
摇摇本章小结	(源)
摇摇习题二	(源)
第 猿章 摇流程控制	(缘)
摇摇猿源猿源顺序结构	(缘)
摇摇摇摇猿源猿源几个基本语句	(缘)
摇摇摇摇猿源猿源输入数据	(缘)
摇摇摇摇猿源猿源输出数据	(缘)
摇摇摇摇猿源猿源顺序结构程序举例	(缘)
摇摇猿源猿源选择结构	(远)
摇摇摇摇猿源猿源实现选择结构的语句	(远)
摇摇摇摇猿源猿源无条件转移语句	(远)
摇摇摇摇猿源猿源选择结构程序举例	(远)
摇摇猿源猿源循环结构	(远)
摇摇摇摇猿源猿源实现循环结构的语句	(远)
摇摇摇摇猿源猿源循环结构程序举例	(远)
摇摇本章小结	(苑)
摇摇习题三	(苑)
第 源章 摇数组、枚举和自定义数据类型	(苑)
摇摇源源源源数组	(苑)
摇摇摇摇源源源源数组的基本概念	(苑)
摇摇摇摇源源源源固定大小的数组	(苑)
摇摇摇摇源源源源动态数组	(愿)
摇摇摇摇源源源源固定数组的初始化和释放动态数组占用的空间	(愿)
摇摇摇摇源源源源数组应用举例	(愿)
摇摇源源源源控件数组	(愿)
摇摇摇摇源源源源控件数组的基本概念	(愿)
摇摇摇摇源源源源建立控件数组	(愿)
摇摇摇摇源源源源控件数组和一般数组的比较	(愿)
摇摇摇摇源源源源控件数组应用举例	(愿)
摇摇源源源源枚举类型	(怨)
摇摇摇摇源源源源定义枚举类型	(怨)
摇摇摇摇源源源源枚举类型举例	(怨)
摇摇源源源源自定义数据类型	(怨)
摇摇摇摇源源源源定义自定义数据类型	(怨)

摇摇摇摇源摇摇自定义数据类型举例	(怨源)
摇摇本章小结	(怨缘)
摇摇习题四	(怨远)
第 缘章摇过程	(怨苑)
摇摇缘源摇过程的基本概念	(怨苑)
摇摇摇摇缘源摇采用过程编程的优点	(怨苑)
摇摇摇摇缘源摇过程的调用和嵌套	(怨苑)
摇摇摇摇缘源摇过程的递归调用	(怨愿)
摇摇缘源摇子过程和函数过程	(怨愿)
摇摇摇摇缘源摇子过程(杂遭过程)	(怨愿)
摇摇摇摇缘源摇函数(云对撞程过程)	(怨园)
摇摇缘源摇过程参数	(怨园)
摇摇摇摇缘源摇形式参数和实际参数	(怨园)
摇摇摇摇缘源摇按值传递参数	(怨园)
摇摇摇摇缘源摇按地址传递参数	(怨园)
摇摇摇摇缘源摇使用可选参数和不定数量的参数	(怨园)
摇摇缘源摇变量的作用范围	(怨园)
摇摇缘源摇过程应用举例	(怨园)
摇摇本章小结	(怨园)
摇摇习题五	(怨园)
第 远章摇窗体	(怨园)
摇摇远源摇创建窗体	(怨园)
摇摇远源摇窗体的属性	(怨园)
摇摇远源摇窗体的常用事件和方法	(怨园)
摇摇摇摇远源摇事件	(怨园)
摇摇摇摇远源摇方法与语句	(怨园)
摇摇远源摇窗体	(怨园)
摇摇摇摇远源摇创建窗体应用程序	(怨园)
摇摇摇摇远源摇窗体的方法及其子窗体布置	(怨园)
摇摇本章小结	(怨园)
摇摇习题六	(怨园)
第 苑章摇菜单设计	(怨园)
摇摇苑源摇菜单编辑器(酝藻灶怎未越器)	(怨园)
摇摇苑源摇创建下拉式菜单	(怨园)
摇摇苑源摇创建快捷菜单	(怨园)
摇摇本章小结	(怨园)
摇摇习题七	(怨园)
第 愿章摇标准控件	(怨园)
摇摇愿源摇命令按钮(悦燥皂皂葬葬怎蚤)	(怨园)
摇摇摇摇愿源摇命令按钮的常用属性	(怨园)

摇摇摇摇愿愿愿摇命令按钮的常用事件	(页码)
摇摇摇摇愿愿愿摇命令按钮的常用方法	(页码)
摇摇愿愿愿摇标签(摇摇与文本框(摇摇用)	(页码)
摇摇摇摇愿愿愿摇摇标签	(页码)
摇摇摇摇愿愿愿摇摇文本框	(页码)
摇摇愿愿愿摇摇定时器(摇摇)	(页码)
摇摇摇摇愿愿愿摇摇定时器常用属性	(页码)
摇摇摇摇愿愿愿摇摇定时器的事件	(页码)
摇摇愿愿愿摇摇滚动条(杂摇)	(页码)
摇摇摇摇愿愿愿摇摇滚动条常用属性	(页码)
摇摇摇摇愿愿愿摇摇滚动条常用事件与方法	(页码)
摇摇愿愿愿摇摇选项按钮(摇摇日)、复选框(摇摇日)与框架(摇摇)	(页码)
摇摇摇摇愿愿愿摇摇选项按钮	(页码)
摇摇摇摇愿愿愿摇摇复选框	(页码)
摇摇摇摇愿愿愿摇摇框架	(页码)
摇摇愿愿愿摇摇列表框(摇摇)和组合框(摇摇)	(页码)
摇摇摇摇愿愿愿摇摇列表框	(页码)
摇摇摇摇愿愿愿摇摇组合框	(页码)
摇摇愿愿愿摇摇驱动器列表框、目录列表框和文件列表框	(页码)
摇摇摇摇愿愿愿摇摇驱动器列表框(摇摇)	(页码)
摇摇摇摇愿愿愿摇摇目录列表框(摇摇)	(页码)
摇摇摇摇愿愿愿摇摇文件列表框(摇摇)	(页码)
摇摇愿愿愿摇摇图片框(摇摇)和图像(摇摇)	(页码)
摇摇摇摇愿愿愿摇摇图片框	(页码)
摇摇摇摇愿愿愿摇摇图像	(页码)
摇摇本章小结	(页码)
摇摇习题八	(页码)
第 怨章 摇常用 粤摇载控件	(页码)
摇摇愿愿愿摇公共对话框(摇摇)	(页码)
摇摇摇摇愿愿愿摇公共对话框的基本属性与方法	(页码)
摇摇摇摇愿愿愿摇“打开”对话框与“另存为”对话框	(页码)
摇摇摇摇愿愿愿摇“颜色”对话框	(页码)
摇摇摇摇愿愿愿摇“字体”对话框	(页码)
摇摇摇摇愿愿愿摇“打印”对话框	(页码)
摇摇摇摇愿愿愿摇调用“宰摇帮助”对话框	(页码)
摇摇愿愿愿摇宰摇公用控件	(页码)
摇摇摇摇愿愿愿摇摇滑块(杂摇)控件	(页码)
摇摇摇摇愿愿愿摇摇进程条(摇摇)控件	(页码)
摇摇摇摇愿愿愿摇摇图像列表(摇摇)控件	(页码)
摇摇摇摇愿愿愿摇摇工具栏(摇摇)控件	(页码)

摇摇摇摇摇摇摇摇摇摇状态栏(杂项)控件	(页码)
摇摇摇摇摇摇摇摇摇摇多媒体控件	(页码)
摇摇摇摇摇摇摇摇摇摇配置驱动程序控件	(页码)
摇摇摇摇摇摇摇摇摇摇配置防火墙控件	(页码)
摇摇本章小结	(页码)
摇摇习题九	(页码)
第 8 章 摇文件	(页码)
摇摇摇摇摇摇摇摇摇摇文件及其概述	(页码)
摇摇摇摇摇摇摇摇摇摇文件基本文件结构	(页码)
摇摇摇摇摇摇摇摇摇摇文件类型	(页码)
摇摇摇摇摇摇摇摇摇摇文件处理	(页码)
摇摇摇摇摇摇摇摇摇摇顺序文件	(页码)
摇摇摇摇摇摇摇摇摇摇顺序文件的打开和关闭	(页码)
摇摇摇摇摇摇摇摇摇摇顺序文件的读写操作	(页码)
摇摇摇摇摇摇摇摇摇摇随机文件	(页码)
摇摇摇摇摇摇摇摇摇摇随机文件的打开和关闭	(页码)
摇摇摇摇摇摇摇摇摇摇随机文件的读写操作	(页码)
摇摇摇摇摇摇摇摇摇摇二进制文件	(页码)
摇摇摇摇摇摇摇摇摇摇常用文件处理函数	(页码)
摇摇本章小结	(页码)
摇摇习题十	(页码)
第 9 章 摇数据库编程	(页码)
摇摇摇摇摇摇摇摇摇摇数据库基础	(页码)
摇摇摇摇摇摇摇摇摇摇关系数据库的结构	(页码)
摇摇摇摇摇摇摇摇摇摇数据库引擎及其接口	(页码)
摇摇摇摇摇摇摇摇摇摇访问数据库的类型	(页码)
摇摇摇摇摇摇摇摇摇摇结构化查询语言(杂项)	(页码)
摇摇摇摇摇摇摇摇摇摇可视化数据管理器(杂项)控件	(页码)
摇摇摇摇摇摇摇摇摇摇创建数据库	(页码)
摇摇摇摇摇摇摇摇摇摇数据窗体设计器	(页码)
摇摇摇摇摇摇摇摇摇摇利用数据控件访问数据库	(页码)
摇摇摇摇摇摇摇摇摇摇数据控件的常用属性、方法和事件	(页码)
摇摇摇摇摇摇摇摇摇摇数据绑定控件	(页码)
摇摇摇摇摇摇摇摇摇摇数据访问对象(阅读)	(页码)
摇摇摇摇摇摇摇摇摇摇阅读的分层结构	(页码)
摇摇摇摇摇摇摇摇摇摇利用阅读访问数据库	(页码)
摇摇摇摇摇摇摇摇摇摇数据对象(阅读)控件	(页码)
摇摇摇摇摇摇摇摇摇摇控件的常用属性	(页码)
摇摇摇摇摇摇摇摇摇摇控件的常用方法	(页码)
摇摇摇摇摇摇摇摇摇摇控件的常用事件	(页码)

摇摇本章小结	(圆缘元)
摇摇习题十一	(圆缘元)
第 圆章 摇摇程序调试与错误处理	(圆缘愿)
摇摇摇摇摇摇灾月程序错误	(圆缘愿)
摇摇摇摇摇摇灾月语法错误(杂类错误)	(圆缘愿)
摇摇摇摇摇摇灾月编译错误(悦类错误)	(圆缘愿)
摇摇摇摇摇摇灾月运行错误(砸类错误)	(圆缘愿)
摇摇摇摇摇摇灾月逻辑错误(蕴类错误)	(圆缘愿)
摇摇摇摇程序调试	(圆缘园)
摇摇摇摇摇摇灾月的工作模式	(圆缘园)
摇摇摇摇摇摇灾月调试工具	(圆缘园)
摇摇摇摇摇摇灾月调试窗口	(圆缘园)
摇摇摇摇错误处理	(圆缘园)
摇摇摇摇摇摇灾月错误捕获语句(欲类语句)	(圆缘园)
摇摇摇摇摇摇灾月栽函数与栽语句	(圆缘元)
摇摇摇摇栽对象	(圆缘元)
摇摇摇摇摇摇灾月栽对象的属性	(圆缘元)
摇摇摇摇摇摇灾月栽对象的方法	(圆缘元)
摇摇本章小结	(圆缘园)
摇摇习题十二	(圆缘园)
附录 粤灾月常用属性、事件和方法	(圆缘愿)
附录 月灾月系统错误代码	(圆缘元)

第 1 章 概 述

VB (Visual Basic) 语言是面向对象的程序设计语言,是目前在 Windows 操作系统平台上广泛使用的 Windows 应用程序开发工具。了解 VB 的发展过程和功能特点,熟悉其集成开发环境以及面向对象程序设计的基本概念和过程,对以后的学习是十分有益的。

1.1 VB 的发展过程和特点

1.1.1 VB 的发展过程

Windows 操作系统的产生,推动了 Windows 应用程序的开发。起初人们只能利用 Windows API (应用编程接口) 的一千多个函数来编写 Windows 应用程序,这给程序设计人员造成了很大的困难。在此情况下,美国微软 (Microsoft) 公司从 20 世纪 90 年代开始推出了一系列可视化的 Windows 应用程序开发工具,如当今流行的 Microsoft Visual Studio 工具套件,VB 便是其中一种。

最早的 VB 出现于 1991 年,这就是 VB 1.0 版本。它标志着 Microsoft 开始将 Basic 语言向可视化编程方向发展,虽然它的功能极其有限,但却具有跨时代的意义。伴随着 Windows 操作平台的不断成熟,VB 版本也不断升级。从 1991 年秋季推出 VB 2.0 到 1993 年 4 月推出 VB 3.0,在很短时间内,VB 已初具规模。人们利用 VB 3.0 可以快速创建各种 Windows 应用程序,例如多媒体应用程序和图形操作界面。然而 VB 的真正推广使用是在 1995 年 8 月推出能够开发 32 位应用程序的 VB 4.0 之后。在 VB 4.0 版本中,Microsoft 利用面向对象的最新技术,极大地扩展了 VB 的功能,提供了创建自定义属性与过程、自定义类模块以及强大的数据库管理能力,这使得用户可以自由创建自定义对象和类,开发各种 MIS 系统。1997 年 VB 5.0 出现,它在开发环境、ActiveX 控件创建、对 Internet 的支持和开发能力等多方面进行了改进和扩充,并且极大地提高了应用程序的运行速度。随着互联网的迅速发展和 Windows 98 的应用,1998 年秋季 Microsoft 又推出了 VB 6.0 版本,该版本在 VB 5.0 的基础上又有了进一步的加强、提高和完善。

VB 6.0 为满足不同层次的用户需要,提供了三个版本:学习版 (Learning)、专业版 (Professional) 和企业版 (Enterprise)。学习版是为初学者提供的 VB 6.0 基础版本,其中包括所有的内部标准控件以及网络、表格和数据绑定等控件;专业版除了学习版的全部功能外,还包括 ActiveX 控件、Internet 控件和 Crystal Report Writer 等开发工具,适用于专业开发人员;企业版则是 VB 6.0 的最高版本,它是专为用户创建功能强大的分布式应用程序、高性能的客户 / 服务器应用程序以及 Internet / Intranet 上的应用程序而设计的,该版本不仅包括专业版的全部功能,而且还具有自动管理器、部件管理器、数据库管理工具、Microsoft Visual SourceSafe 面向工程版的控制系统等等。

1.1.2 VB 的特点

Visual Basic 是在早期 Basic 语言的基础上发展起来的一种可视化的 Windows 平台上的程序开发工具。在这里，“Visual”为“可视化”的意思。所谓“可视化”是一种开发图形用户界面（GUI）的方法，用户不再需要编写大量代码去描述界面的外观，而只需在可视化的编程环境中，将形象直观的预置对象直接拖放在屏幕上即可，这一点是 VB 最显著的特点。“Basic”则表明 VB 是对原有 Basic 语言的一种扩展，它继承了 Basic 语言语法简练，简单易学的特点。但是，在 Windows 环境下编程与在 DOS 环境下编程有着很大区别，VB 不仅仅只是 Basic 的扩展，它采用了新的事件驱动和面向对象的编程机制，而且可以直接调用 Windows API 中的函数，这使得 VB 的应用范围极其广泛。利用 VB 用户不仅可以编写各种 Windows 应用程序，而且还能将 Excel 电子表格、Word 字处理软件等嵌入其中直接使用；此外还可以通过调用 Access 等多种格式的数据库系统，对数据库进行管理与操作。

VB 具有以下七个主要特点：

（1）面向对象的可视化编程，使开发人员只需编写很少量的程序代码，就可以快速开发出标准的 Windows 应用程序，极大地提高了程序设计效率。

（2）事件驱动的编程机制，比传统的过程驱动方式更适合于 Windows 图形用户界面的开发。因为在图形用户界面应用程序中，程序的执行是由用户的操作动作即事件决定的，而不是由事先设计好的程序流程所控制。这样在 VB 中程序员只需对各种用户动作编写相应的处理代码，程序既简练又易于维护。

（3）友好的 VB 集成开发环境，使用户可以在其中很轻松地进行界面设计、代码编写、程序调试及编译运行等工作。

（4）可以对多种数据库系统进行数据访问，包括 Microsoft Access、dBASE、Microsoft FoxPro、Paradox 和 Microsoft SQL Server 等。

（5）通过 ActiveX 技术，可以使用其他程序提供的功能，例如 Microsoft Word 字处理软件、Microsoft Excel 电子表格以及其他 Windows 应用程序。

（6）支持动态数据交换（DDE）、动态链接库（DLL）、对象链接与嵌入（OLE）以及 ActiveX 技术。动态数据交换技术可使 VB 应用程序与其他 Windows 应用程序之间建立动态的数据通信；动态链接库技术使 VB 程序可以方便地调用 C 语言、汇编语言或 Windows API 的函数；对象链接与嵌入技术允许将其他各种基于 Windows 的应用软件作为一个对象链接或嵌入到 VB 应用程序中，并对其进行操作；ActiveX 技术则是 OLE 技术的进一步发展。

（7）完善的联机帮助功能。通过与 VB 6.0 的安装程序捆绑在一起的 Microsoft MSDN（Microsoft Developer Network）联机帮助文档，用户可以随时方便地得到各种帮助信息，以解决用户在开发过程中遇到的各种各样的问题。

1.2 VB 6.0 集成开发环境

VB 不仅是一种程序设计语言，而且是一个完全集成化的 32 位应用程序的可视化开发工具，程序设计人员通过与图形界面的交互完成应用程序的创建、修改、调试、运行以及发布。

1.2.1 启动 VB 6.0

要进入 VB 6.0 集成开发环境，首先需要启动 VB 6.0。与一般的 Windows 应用软件一样，VB 6.0 的启动方式有 3 种。

(1) 使用“开始”菜单方式启动 VB。单击 Windows 98 任务栏中的“开始”按钮，选择“程序”菜单，然后打开“Microsoft Visual Basic 6.0 中文版”菜单中的“Microsoft Visual Basic 6.0 中文版”程序，即可启动 VB 6.0。

(2) 使用命令行方式启动 VB。单击 Windows 98 任务栏中的“开始”按钮，然后单击“运行”菜单，屏幕上出现如图 1.1 所示的“运行”对话框，用户可以在对话框中输入命令行，例如：“C:\Program Files \ Microsoft Visual Studio\VB\Vb6.exe”，然后单击“确定”按钮即可启动 VB 6.0。

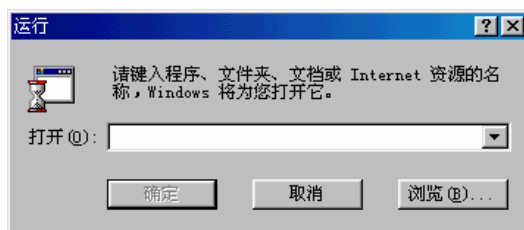


图 1.1 “运行”对话框

(3) 使用快捷方式启动 VB。利用 Windows 98 建立快捷方式的功能，在桌面上建立 VB 6.0 程序的快捷图标，双击该图标则可立即启动 VB 6.0。

VB 6.0 启动后，屏幕上会首先显示一个“新建工程”对话框，如图 1.2 所示。对话框中有 3 个选项卡：“新建”用于创建一个新的工程；“现存”用于选择和打开一个现有的工程；“最新”则用于列出和打开一个最近建立或使用过的工程。双击“新建”选项卡中的“标准 EXE”项（默认选项）或直接单击“打开”按钮，即可进入 VB 6.0 集成开发环境。在图 1.2 中，共列出了以下 9 种 VB 应用程序的类型。



图 1.2 “新建工程”对话框

(1) 标准 EXE：这是典型的 VB 应用程序，最终可生成一个标准的可执行文件（扩展名为 exe）。

(2) ActiveX EXE 与 ActiveX DLL：仅在专业版中提供，用于生成 ActiveX 构件。它是支持对象链接与嵌入（OLE）技术的自动化服务器程序。这两种类型的应用程序在编程时完全相同，区别在于最终一个编译成为可执行文件（exe），另一个则编译成为动态链接库（dll）。

(3) ActiveX 控件：仅在专业版中提供，用于创建用户的 ActiveX 控件。

(4) ActiveX 文档 EXE 和 ActiveX 文档 DLL：ActiveX 文档指的是可以在 Web 浏览器环境中运行的 VB 应用程序。与前面相似，二者的区别在于最终编译生成的一个是可执行文件，一个是动态链接库。

(5) VB 应用程序向导：利用该向导，用户可以很快建立起应用程序的框架，以减轻编程工作量。

(6) VB 企业版控件：这是 VB 企业版提供的类型，用于开发 VB 控件。

(7) 外接程序：该类型应用程序主要用来创建用户自己的外接程序，以扩展 VB 集成开发环境的功能。

1.2.2 VB 6.0 集成开发环境

与 Windows 环境类似，VB 6.0 集成开发环境由标题栏、菜单栏、工具栏以及一些专用的窗口组成。这些窗口如图 1.3 所示，包括窗体设计器、工具箱、工程资源管理器、属性窗口、代码编辑器和窗体布局窗口等。

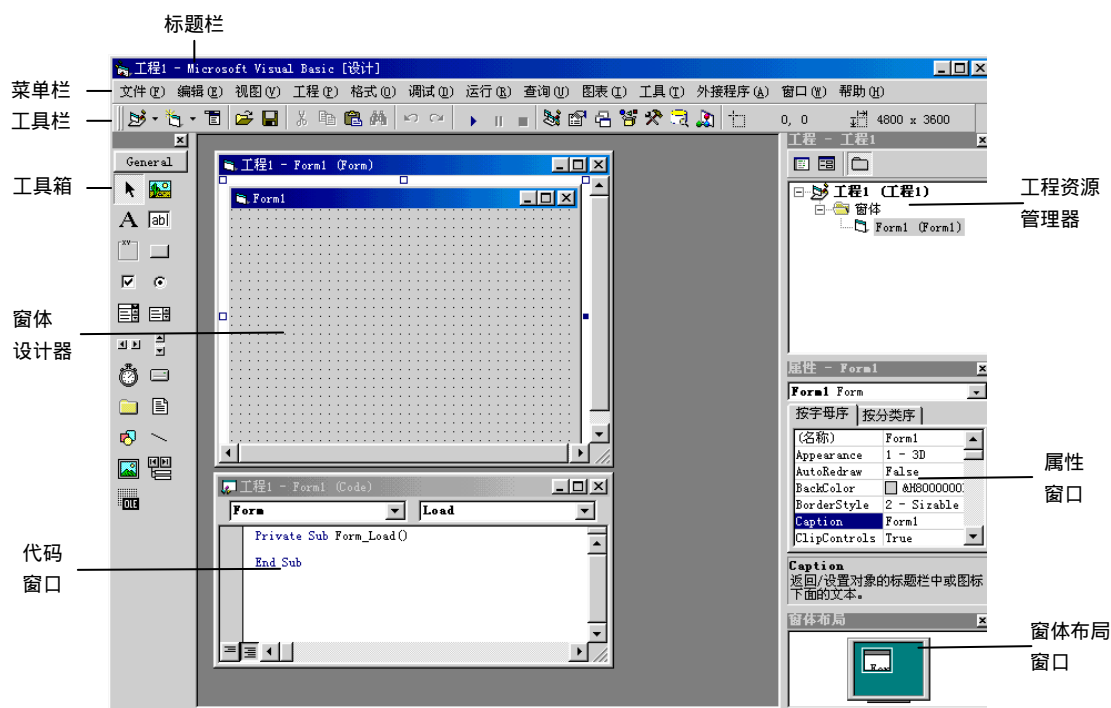


图 1.3 VB 6.0 集成开发环境

1. 标题栏、菜单栏和工具栏

标题栏主要用于显示当前激活的工程名称及其状态，此外还有窗口控制菜单图标和最小化、最大化 / 还原、关闭按钮。

菜单栏用于显示 VB 6.0 菜单系统，包括“文件”、“编辑”、“视图”、“工程”、“窗口”、“格式”、“调试”、“运行”、“工具”、“外接程序”和“帮助”等 11 个主菜单。单击某主菜单，即可打开一个包含子菜单项的下拉菜单。

工具栏以图标按钮形式为用户提供了多种常用命令的快捷访问方式。

2. 工程资源管理器 (Project Explorer)

如图 1.4 所示，工程资源管理器以树状的层次结构显示组成当前工程的所有文件，并对工程进行管理。



图 1.4 工程资源管理器

所谓“工程”是 VB 应用程序的基本单位，工程对应的文件称为工程文件，其扩展名为 vbp，它是用来管理应用程序对象的文件，其中保存着工程所需的所有文件的一个列表。这些文件包括窗体文件 (frm)、标准模块文件 (bas)、类模块文件 (cls)、资源文件 (res)、二进制数据文件 (frx)、ActiveX 控件 (ocx)、用户文档文件 (dob) 等。一般一个工程就代表一个应用程序，但是在 VB 6.0 中也可以使用工程组的概念来创建一个应用程序，即一个应用程序可以由一个工程组内的数个工程构成。平时工程组内的各个工程单独运行，当需要时便可将组内的多个工程组合成一个可执行文件来运行。

在工程资源管理器中有 3 个按钮：“查看代码”按钮用于切换到“代码编辑器”，显示和编辑代码；“查看对象”按钮用于显示所选模块相应的对象窗口，如选定一个窗体，则按下“查看对象”按钮将“窗体设计器”打开；“切换文件夹”按钮用于显示或隐蔽包含在工程文件夹中所有文件下的项目列表。

利用工程资源管理器用户可以方便地查看工程中的对象和文件。单击鼠标右键，可以显示一个快捷菜单，通过其上的菜单项可实现相应功能，如添加文件、保存文件、删除文件和打印文件等。

3. 工具箱 (ToolBox)

VB 工具箱用于存放和显示建立 VB 应用程序所需的各种控件。标准工具箱如图 1.3 所示，其中包含有 20 个 VB 标准控件和 1 个指针图标，其中指针不是 VB 控件，它仅用于移动窗体和控件，并调整它们的大小。

另外，VB 6.0 允许用户将其他的 ActiveX 控件添加到工具箱中，具体方法有 2 种。

(1) 选择“工程”菜单下的“部件”菜单项，打开“部件”对话框，如图 1.5 所示，然

后在“控件”选项卡中选定所需控件，单击“确定”按钮，即可将所选的 ActiveX 控件添加到工具箱中。

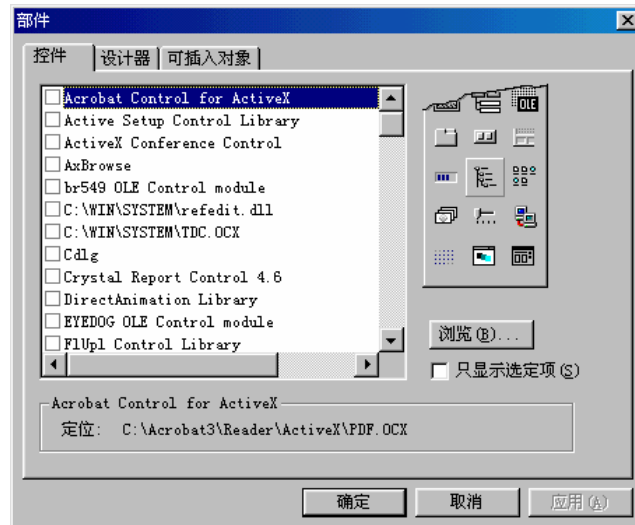


图 1.5 “部件”对话框

(2) 在工具箱中建立用户自定义的选项卡。首先在工具箱上单击鼠标右键，从弹出的快捷菜单中选择“添加选项卡”，出现的“添加选项卡”对话框，如图 1.6 所示。在其中输入要建立的选项卡名称，并按“确定”按钮，此时在工具箱底部便出现所建立的选项卡图标，用鼠标左键将所需的控件“拖到”选项卡上即可。这种方法可以有效地避免由于控件过多造成的屏幕布局浪费，使控件的查找更迅速。

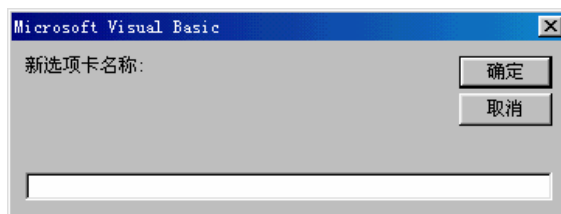


图 1.6 “添加选项卡”对话框

4. 窗体设计器 (Form Designer)

“窗体设计器”，又称为“窗体窗口”或“对象窗口”，如图 1.3 中所示，主要用来设计应用程序的界面，并编辑应用程序。窗体设计器的位置和大小可以随意进行调整。窗体即窗口，是 VB 应用程序的重要组成部分。一个应用程序可以拥有多个窗体，但每个窗体都必须有一个惟一的标识名称，VB 在默认状态下分别以 Form1、Form2 命名窗体。

在程序设计状态下所有窗体都是可视的，窗体上具有可改变间距的控件定位网格线。在程序运行时通过设置窗体的 Visible 属性来控制其可视性，但窗体上的网格线不会显示出来。

5. 代码编辑器 (Code Editor)

“代码编辑器”，又称为“代码窗口”，用来编写、显示和编辑程序代码，如图 1.3 中所示。通常代码编辑器是看不见的，可通过以下 4 种方法打开代码编辑器：