

<<Visual Basic.NET程序设计>>

图书基本信息

书名：<<Visual Basic.NET程序设计>>

13位ISBN编号：9787301163061

10位ISBN编号：7301163061

出版时间：2010-9

出版时间：北京大学出版社

作者：杨光祖 编

页数：218

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

Visual Basic.NET语言是由美国Microsoft（微软）公司开发的一种功能强大，具有面向对象、分布式、可移植等性能的多线程动态计算机编程语言。因其简单易学、开发快捷、功能强大，深受广大专业和非专业计算机程序开发人员的喜爱。Visual Basic.NET（简称VB.NET）继承了BASIC语言面向普通使用者和易学易用的优点，同时又引入了可视化图形用户界面的程序设计方法和面向对象的程序设计，成为当今世界使用最为广泛、最有影响的程序开发语言之一。

本书共分为9章，每章都配有大量丰富的练习题和实验实训题。

第1章：VB.NET概述。

介绍.NET框架、集成开发环境。

第2章：VB.NET简单程序设计。

介绍VB.NET的编程步骤、窗体和基本控件、程序调试方法、环境设置和帮助的使用。

第3章：VisualBasic.NET语言基础。

主要介绍VB.NET程序的书写规则、基本数据类型、常用系统函数、程序设计结构。

第4章：数组与过程。

主要介绍数组的概念、操作，过程的调用和参数传递，变量与作用域。

## <<Visual Basic.NET程序设计>>

### 内容概要

本书根据编者多年从事计算机软件项目开发与软件教学的实际经验编写而成，以培养学生实际动手能力为目标，从教学实践的角度出发，立足于提高学生的程序设计应用能力。

全书理论分析透彻严谨，实例丰富生动，内容由浅入深，能快速引导学生进入VB.NET编程世界。

本书以Visual Studio 2003为蓝本，将内容分为9章，包括VB.NET的程序设计语言规范、函数、数组；Windows程序设计界面技术、控件；面向对象的程序设计方法与原理；数据库系统设计与程序调试等内容。

每章配有大量的练习题和切合实际的实验实训题目，供教师和学生选择使用。

本书可作为高职高专和本科应用型、技能型人才培养的计算机相关专业的教材及初、中级培训班的教材，也可作为广大VB.NET编程爱好者学习和提高的参考书，同时也适合VB.NET的初、中级用户使用。

## 书籍目录

第1章 VB.NET概述 1.1 NET框架简介 1.1.1 Visual Studio介绍 1.1.2 NET Framework概述 1.2 Visual Studio的安装和启动 1.2.1 Visual Studio支持的操作系统及软件环境要求 1.2.2 Visual Studio.NET安装步骤 1.3 集成开发环境 1.3.1 Visual Studio.NET集成开发环境 1.3.2 Visual Basic.NET集成开发环境 1.4 第一个例子 1.5 习题 1.6 实验实训第2章 VB.NET简单程序设计 2.1 VB.NET的编程步骤 2.1.1 控制台程序编程步骤 2.1.2 WinForm程序编写步骤 2.2 窗体 2.2.1 窗体结构 2.2.2 窗体属性 2.2.3 窗体事件 2.2.4 窗体的方法 2.2.5 设置启动窗体 2.2.6 例子 2.3 基本控件 2.3.1 标签(Label) 2.3.2 文本框(TextBox) 2.3.3 命令按钮(Button) 2.4 程序调试 2.5 环境设置 2.6 使用帮助 2.6.1 自定义动态帮助 2.6.2 使用在线帮助 2.7 习题 2.8 实验实训第3章 Visual Basic.NET语言基础 3.1 Visual Basic.NET程序的书写规则 3.1.1 关键字与标识符 3.1.2 程序书写规则 3.1.3 注释 3.1.4 语句格式中的符号约定 3.2 基本数据类型 3.2.1 数值数据类型 3.2.2 文本数据类型 3.2.3 布尔数据类型 3.2.4 日期数据类型 3.2.5 对象数据类型 3.3 常量与变量 3.3.1 常量 3.3.2 变量 3.4 运算符和表达式 3.4.1 算术运算符 3.4.2 连接运算符 3.4.3 关系运算符 3.4.4 逻辑运算符 3.4.5 表达式 3.5 常用系统函数 3.5.1 数学函数 3.5.2 字符处理函数 3.5.3 日期时间函数 3.5.4 类型转化函数 3.5.5 随机数函数Rnd 3.6 顺序结构程序设计 3.6.1 赋值语句 3.6.2 InputBox函数和MsgBox函数 3.7 选择结构程序设计 3.7.1 单分支选择结构语句 3.7.2 双分支选择结构语句 3.7.3 多分支选择结构语句 3.8 循环结构程序设计 3.8.1 For...Next循环语句 3.8.2 While循环语句 3.8.3 Do...Loop循环语句 3.8.4 跳离语句Exit 3.8.5 循环嵌套 3.9 习题 3.10 实验实训第4章 数组与过程 4.1 数组, 枚举, 结构的概念、操作 4.1.1 数组 4.1.2 枚举 4.1.3 结构 4.2 过程的定义和调用 4.2.1 子过程的定义和调用 4.2.2 函数过程的定义和调用 4.3 参数传递 4.3.1 传值与传址 4.3.2 数组参数的传递 4.3.3 参数数组的传递 4.4 变量与过程的作用域 4.4.1 变量的作用域 4.4.2 过程的作用域 4.5 习题 4.6 实验实训第5章 常用控件 5.1 常用控件的使用方法 5.2 文本控件 5.2.1 RichTextBox控件 5.3 按钮控件 5.3.1 Button控件 5.3.2 RadioButton、CheckBox和GroupBox控件 5.4 列表选择控件 5.4.1 ListBox列表框 5.4.2 CheckedListBox控件 5.4.3 ComboBox组合框 5.5 对话框控件 5.5.1 FileDialog文件对话框控件 5.5.2 FontDialog控件和ColorDialog控件 5.6 日期时间控件 5.6.1 MonthCalendar控件 5.6.2 DateTimePicker控件 5.7 图片框PictureBox控件 5.8 习题 5.9 实验实训第6章 界面设计 6.1 菜单设计 6.1.1 设计主菜单 6.1.2 ContextMenu控件 6.2 工具栏ToolBar控件 6.3 状态栏StatusBar控件 6.4 多文档界面程序(MDI) 6.5 习题 6.6 实验实训第7章 面向对象编程技术 7.1 面向对象程序设计概述 7.1.1 面向对象的基本概念 7.1.2 对象与类 7.2 类的创建与使用 7.2.1 创建类 7.2.2 创建属性 7.2.3 创建方法 7.2.4 创建事件 7.2.5 示例 7.3 类的继承与重载 7.3.1 类的继承 7.3.2 类的重载 7.4 命名空间 7.5 习题 7.6 实验实训第8章 数据库编程 8.1 数据库概述 8.1.1 数据库基础 8.1.2 关系数据库及常用术语 8.1.3 建立数据库 8.1.4 实训操作 8.2 结构化查询语言SQL 8.2.1 SQL语句基础 8.2.2 查询语句 8.2.3 添加语句 8.2.4 更新语句 8.2.5 删除语句 8.2.6 实训操作 8.3 ADO.NET 8.3.1 ADO.NET概述 8.3.2 体系结构 8.3.3 实训操作 8.4 数据库访问 8.4.1 利用可视化工具进行数据库访问 8.4.2 编写代码进行数据库访问 8.4.3 实训操作 8.5 综合实例 8.5.1 界面设计 8.5.2 类设计 8.5.3 代码实现 8.6 习题第9章 程序调试与异常处理 9.1 程序调试 9.1.1 错误种类 9.1.2 调试与排错 9.2 异常(Exception)类 9.3 异常处理方法 9.3.1 Try...Catch语句 9.3.2 On Error语句 9.3.3 Throw抛出异常 9.4 习题 9.5 实验实训

章节摘录

控件是开发应用程序用来设计用户界面的重要元素，也是应用程序中人机交互的基本渠道，使用控件进行应用程序的开发，是可视化程序设计的一个重要特征，掌握常用控件的使用方法是Visual Basic程序设计的基础。

控件是具有图形界面的对象，它同样具有属性、方法、事件，通常控件都是以OCX为后缀的特殊文件（有经验的程序员都自己编写一些适合自己进行应用程序开发的控件）。

使用控件进行应用程序的开发，给编写程序带来了很大的方便。但是，这些控件仅仅是躯壳而已，不能实现任何功能，要想使得整个程序实现所设计的功能，还得向其中添加程序代码。

一般将其分为两大类：（1）内部控件（常用控件 / 标准控件）；（2）扩展的ActiveX控件。这一章主要介绍常用控件的使用。

Visual Basic中的控件数量众多，默认的工具箱中就有46个常用控件，它们都以快捷按钮的形式显示在工具箱中，名称位于按钮图标的右侧。

5.1 常用控件的使用方法1.控件的生成在工具箱中有很多标准控件的按钮，这些按钮就是这些控件的模板，当需要在窗体中生成一个控件时，常用两种生成方法。

.....

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>