第一图书网, tushu007.com

<<网络程序设计案例教程>>

图书基本信息

书名：<<网络程序设计案例教程>>

13位ISBN编号：9787302171799

10位ISBN编号：7302171793

出版时间：2008-6

出版时间：清华大学出版社

作者：孙践知

页数：251

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

内容概要

本书将基于网络程序设计所必须掌握的知识归纳为若干案例，每个案例解决一个问题。
初学者只需模仿案例，获得直接体验，然后再学习和案例相关的知识。
通过一个单元接着一个单元的模仿、学习，逐步地构成完整的知识体系。
　本书涉及ASP.NET和SQL Server两部分内容，全书共8章及3个附录，系统地介绍了使用ASP.NET开发基于网络应用程序的基本概念、方法和技能，涉及C#的基本语法规则、ASP.NET常用控件、内置对象、类库、SQL Server数据库基础知识、ASP.NET和数据库的连接方式及XML数据访问等内容。
　本书采用了最新版本的软件Visual Studio 2005和SQL Server 2005。
　本书适合作为高等院校网络程序设计课程的教材，书中涉及了大量的最新网络程序设计技术，也可作为程序开发人员的参考书。

书籍目录

第1章 ASP.NET概述 1.1 基本概念 1.1.1 B/S和C/S结构 1.1.2 静态网页和动态网页 1.1.2.1 静态网页 1.1.2.2 动态网页 1.1.3 Web服务器的概念 1.1.4 HTTP协议 1.1.5 .NET架构 1.2 任务1：安装ASP.NET 1.2.1 要求和目的 1.2.2 操作步骤 1.2.3 相关知识 1.2.3.1 ASP.NET概述 1.2.3.2 ASP.NET开发工具 1.2.3.3 ASP.NET的工作过程 1.3 任务2：建立最简单的ASP.NET程序 1.3.1 要求和目的 1.3.2 操作步骤 1.3.3 相关知识 1.3.3.1 ASP.NET解决方案 1.3.3.2 ASP.NET文件类型 1.3.3.3 代码分离技术 1.3.3.4 Visual Studio 2005开发环境 1.4 任务3：建立最简单的可交互的ASP.NET程序 1.4.1 要求和目的 1.4.2 操作步骤 1.4.3 相关知识 1.4.3.1 ASP.NET页面程序结构 1.4.3.2 C#程序结构 1.4.3.3 Web页面的处理过程 1.4.3.4 类和名字空间 1.4.3.5 控件、属性和方法 1.4.3.6 事件和事件驱动 1.4.3.7 HTML服务器控件和Web服务器控件 1.5 小结 1.6 作业第2章 Web服务器控件 2.1 任务1：设置文本字体 2.1.1 要求和目的 2.1.2 操作步骤 2.1.3 相关知识 2.1.3.1 命令按钮 2.1.3.2 文本框 2.1.3.3 单选按钮 2.1.3.4 复选框 2.2 任务2：图片浏览器 2.2.1 要求和目的 2.2.2 操作步骤 2.2.3 相关知识 2.2.3.1 Image控件 2.2.3.2 标签和下拉列表框控件 2.2.3.3 页面属性 2.2.3.4 页面指示符 2.3 任务3：选择出行时间和目标城市 2.3.1 要求和目的 2.3.2 操作步骤 2.3.3 相关知识 2.3.3.1 列表框控件 2.3.3.2 日历控件 2.3.3.3 布局菜单 2.3.3.4 格式菜单和布局工具栏 2.4 任务4：设计用户注册界面 2.4.1 要求和目的 2.4.2 操作步骤 2.4.3 相关知识 2.4.3.1 必填验证控件 2.4.3.2 比较验证控件 2.4.3.3 范围验证控件 2.4.3.4 正则表达式验证控件 2.4.3.5 验证摘要控件 2.4.3.6 自定义验证控件 2.5 任务5：设计页面Banner 2.5.1 要求和目的 2.5.2 操作步骤 2.5.3 相关知识 2.5.3.1 容器控件 2.5.3.2 Web用户控件 2.5.3.3 定义Web用户控件 2.5.3.4 Web用户控件和Web页面的关系 2.6 小结 2.7 作业第3章 ASP.NET内置对象 3.1 任务1：获取来访者信息 3.1.1 要求和目的 3.1.2 操作步骤 3.1.3 相关知识 3.1.3.1 Request对象概述 3.1.3.2 Browser属性 3.1.3.3 UrlReferrer属性 3.1.3.4 AcceptType属性 3.1.3.5 Headers属性 3.1.3.6 QueryString属性 3.2 任务2：使用Cookie登录网站 3.2.1 要求和目的 3.2.2 操作步骤 3.2.3 相关知识 3.2.3.1 Response对象概述 3.2.3.2 Cookie对象概述 3.2.3.3 Cookie的应用 3.2.3.4 对Cookie对象进行操作 3.2.3.5 Cookie对象的常用属性和方法 3.3 任务3：为页面设置生存时间 3.3.1 要求和目的 3.3.2 操作步骤 3.3.3 相关知识 3.3.3.1 Session对象概述 3.3.3.2 Session和Cookie的比较 3.3.3.3 Session对象的常用属性和方法 3.4 任务4：简单聊天室 3.4.1 要求和目的 3.4.2 操作步骤 3.4.3 相关知识 3.4.3.1 Application对象概述 3.4.3.2 Application对象的常用属性和方法 3.5 任务5：动态在线人数统计 3.5.1 要求和目的 3.5.2 操作步骤 3.5.3 相关知识 3.5.3.1 Global.asax配置文件 3.5.3.2 Web.config配置文件 3.5.3.3 Global事件 3.6 小结 3.7 作业第4章 ASP.NET类库 4.1 .NET类库概述 4.2 任务1：小学生算术测验 4.2.1 要求和目的 4.2.2 操作步骤 4.2.3 相关知识 4.2.3.1 System名字空间 4.2.3.2 Math类 4.2.3.3 Random类 4.2.3.4 DateTime结构 4.2.3.5 String类 4.2.3.6 Convert类 4.2.3.7 DayOfWeek枚举 4.3 任务2：上传文件到指定目录 4.3.1 要求和目的 4.3.2 操作步骤 4.3.3 相关知识 4.3.3.1 System.IO名字空间 4.3.3.2 FileUpload类 4.3.3.3 File类 4.3.3.4 Directory类 4.3.3.5 StreamReader类和StreamWriter类 4.3.3.6 “～”和“@”运算符 4.4 任务3：绘制花瓣图案 4.4.1 要求和目的 4.4.2 操作步骤 4.4.3 相关知识 4.4.3.1 System.Drawing名字空间 4.4.3.2 Point结构 4.4.3.3 Color结构 4.4.3.4 Pen类 4.4.3.5 Bitmap类 4.4.3.6 Graphics类 4.5 任务4：简单端口扫描器 4.5.1 要求和目的 4.5.2 操作步骤 4.5.3 相关知识 4.5.3.1 System.Net名字空间 4.5.3.2 System.Net.Sockets名字空间 4.5.3.3 TcpClient类 4.5.3.4 Dns类 4.6 任务5：简单邮件发送系统 4.6.1 要求和目的 4.6.2 操作步骤 4.6.3 相关知识 4.6.3.1 System.Web.Mail名字空间 4.6.3.2 MailMessage类 4.6.3.3 MailAttachment类 4.6.3.4 SmtpMail类 4.6.3.5 System.Net.Mail名

字空间 4.6.3.6 SmtpClient类 4.7 任务6：多线程应用 4.7.1 要求和目的 4.7.2 操作步骤 4.7.3 相关知识 4.7.3.1 多线程的基本概念 4.7.3.2 System.Threading名字空间 4.7.3.3 Thread类 4.7.3.4 线程的状态 4.7.3.5 线程的优先级 4.8 任务7：文件加密 4.8.1 要求和目的 4.8.2 操作步骤 4.8.3 相关知识 4.8.3.1 数据加密的基本概念 4.8.3.2 主要加密算法 4.8.3.3 System.Security.Cryptography 名字空间 4.8.3.4 DES类 4.8.3.5 CryptoStream类 4.9 小结 4.10 作业第5章 SQL Server数据库基础 5.1 数据库概述 5.1.1 基本概念 5.1.2 SQL Server概述 5.2 任务1：安装SQL Server 2005 5.2.1 要求和目的 5.2.2 操作步骤 5.2.3 相关知识 5.2.3.1 SQL Server数据库对象 5.2.3.2 SQL Server 系统数据库 5.3 任务2：建立学生信息数据库 5.3.1 要求和目的 5.3.2 操作步骤 5.3.3 相关知识 5.3.3.1 表的概念 5.3.3.2 字段和字段类型 5.3.3.3 数据完整性 5.3.3.4 约束 5.3.3.5 索引 5.4 任务3：建立学生选课数据库 5.4.1 要求和目的 5.4.2 操作步骤 5.4.3 相关知识 5.4.3.1 单表数据库问题分析 5.4.3.2 多表 5.4.3.3 外键和参照完整性 5.5 任务4：查询学生选课数据库 5.5.1 要求和目的 5.5.2 操作步骤 5.5.3 相关知识 5.5.3.1 多表数据库问题分析 5.5.3.2 视图的概念 5.5.3.3 关系图 5.5.3.4 分组和筛选 5.6 任务5：备份和恢复数据库 5.6.1 要求和目的 5.6.2 操作步骤 5.7 小结 5.8 作业第6章 SQL查询语言 6.1 概述 6.2 任务1：建立学生选课数据库 6.2.1 要求和目的 6.2.2 操作步骤 6.2.3 相关知识 6.2.3.1 SQL语法基本规则 6.2.3.2 CREATE语句 6.2.3.3 DROP语句 6.2.3.4 ALTER语句 6.3 任务2：查询学生选课数据库 6.3.1 要求和目的 6.3.2 操作步骤 6.3.3 相关知识 6.3.3.1 INSERT语句 6.3.3.2 SELECT语句 6.3.3.3 DELETE语句 6.3.3.4 UPDATE语句 6.4 任务3：使用存储过程 6.4.1 要求和目的 6.4.2 操作步骤 6.4.3 相关知识 6.4.3.1 存储过程 6.4.3.2 T-SQL的运算符 6.4.3.3 T-SQL的函数 6.4.3.4 T-SQL的变量 6.4.3.5 T-SQL的语句 6.5 任务4：使用触发器和游标 6.5.1 要求和目的 6.5.2 操作步骤 6.5.3 相关知识 6.5.3.1 触发器 6.5.3.2 游标 6.6 小结 6.7 作业第7章 使用ADO.NET访问数据库 7.1 ADO.NET概述 7.2 任务1：在ASP.NET中操作数据库中的数据（Ⅰ） 7.2.1 要求和目的 7.2.2 操作步骤 7.2.3 相关知识 7.2.3.1 数据库相关的名字空间 7.2.3.2 SqlConnect类 7.2.3.3 SqlCommand类 7.2.3.4 SqlDataReader类 7.3 任务2：在ASP.NET中操作数据库中的数据（Ⅱ） 7.3.1 要求和目的 7.3.2 操作步骤 7.3.3 相关知识 7.3.3.1 SqlDataAdapter类 7.3.3.2 DataSet 7.3.3.3 数据绑定 7.4 任务3：显示选课数据库的数据（Ⅰ） 7.4.1 要求和目的 7.4.2 操作步骤 7.4.3 相关知识 7.4.3.1 SqlDataSource控件 7.4.3.2 GridView控件 7.4.3.3 模板的使用 7.5 任务4：显示选课数据库的数据（Ⅱ） 7.5.1 要求和目的 7.5.2 操作步骤 7.5.3 相关知识 7.5.3.1 DataList控件 7.5.3.2 DetailsView控件 7.5.3.3 FormView控件 7.6 任务5：以报表的方式输出选课数据库 7.6.1 要求和目的 7.6.2 操作步骤 7.6.3 相关知识 7.6.3.1 报表 7.6.3.2 ReportViewer控件 7.6.3.3 Repeater控件 7.7 小结 7.8 作业第8章 在ASP.NET中使用XML处理数据 8.1 任务1：创建XML文档 8.1.1 要求和目的 8.1.2 操作步骤 8.1.3 相关知识 8.1.3.1 XML和HTML的异同 8.1.3.2 XML文档的结构 8.1.3.3 创建XML文档 8.2 任务2：显示XML文档 8.2.1 要求和目的 8.2.2 操作步骤 8.2.3 相关知识 8.3 任务3：在Web页面中读写XML文档 8.3.1 要求和目的 8.3.2 操作步骤 8.3.3 相关知识 8.3.3.1 XML控件 8.3.3.2 DataSet对象中与XML相关的方法 8.4 小结 8.5 作业附录A C#程序设计语言基础 A.1 C#概述 A.2 任务1：创建简单的C#程序 A.2.1 要求和目的 A.2.2 操作步骤 A.2.3 相关知识 A.2.3.1 C#的基本语法规则 A.2.3.2 变量和赋值语句 A.2.3.3 数据类型 A.2.3.4 表达式 A.3 任务2：有分支的C#程序 A.3.1 要求和目的 A.3.2 操作步骤 A.3.3 相关知识 A.3.3.1 if语句 A.3.3.2 问号表达式 A.3.3.3 switch语句 A.3.3.4 break语句 A.4 任务3：有循环的C#程序 A.4.1 要求和目的 A.4.2 操作步骤 A.4.3 相关知识 A.4.3.1 for语句 A.4.3.2 自加和自减运算 A.4.3.3 while语句 A.4.3.4 do while语句 A.4.3.5 continue语句 A.5 任务4：使用数组的C#程序 A.5.1 要求和目的 A.5.2 操作步骤 A.5.3 相关知识 A.5.3.1 数组的定义 A.5.3.2 数组的引用 A.6 任务5：C#面向对象程序 A.6.1 要求和目的 A.6.2 操作步骤 A.6.3 相关知识 A.6.3.1 对

象 A.6.3.2 类 A.6.3.3 方法的重载 A.6.3.4 类的继承 A.7 任务6：C#异常处理 A.7.1 要求和目的 A.7.2 操作步骤 A.7.3 相关知识 A.7.3.1 try-catch语句 A.7.3.2 Exception类 A.8 小结 A.9 作业附录B HTML概要 B.1 HTML概述 B.2 任务1：创建简单的HTML文档 B.2.1 要求和目的 B.2.2 操作步骤 B.2.3 相关知识 B.2.3.1 HTML的基本结构 B.2.3.2 标签 B.2.3.3 常见标签 B.2.3.4 字符实体 B.3 任务2：创建带有表格的HTML文档 B.3.1 要求和目的 B.3.2 操作步骤 B.3.3 相关知识 B.3.3.1 图形标签 B.3.3.2 超链接标签 B.3.3.3 表格标签 B.3.3.4 项目符号标签 B.3.3.5 meta标签 B.4 小结 B.5 作业附录C 建立Web服务器 C.1 概述 C.2 任务1：建立Web服务器 C.2.1 要求和目的 C.2.2 操作步骤 C.3 任务2：Web站点设置 C.3.1 要求和目的 C.3.2 操作步骤 C.4 任务3：多Web站点配置 C.4.1 要求和目的 C.4.2 操作步骤 C.5 小结 C.6 作业参考文献

章节摘录

第1章 ASP.NET概述学习提示本章以3个任务为线索，介绍了ASENET所涉及的基本概念、程序结构和开发环境，涉及的问题和概念非常多，在本章中，建议读者重点掌握简单的ASENET程序结构和开发方法，其他细节问题可留待后面的章节解决。
在本书的叙述中会大量涉及具体操作，在描述具体操作的过程时做如下约定：将涉及的操作对象用“【】”符号括住，如单击【开始】按钮打开Windows开始菜单，若在符号“【】”前没有特别说明表示“单击”，即【开始】等同于单击【开始】菜单。
用符号“→”表示下一步，如【开始】→【程序】。
ASP.NET是一种功能强大的用于创建动态Web页面的服务器端技术，是目前三种主要的动态Web页面设计技术之一，另外两种分别是JSP（Java Server Pages）和PHP（Hypertext Preprocessor），三种技术各具特色。
ASP.NET是Microsoft公司推出的新一代软件开发平台.NET的组件之一，它集中体现了Microsoft软件产品一贯的功能强大、用户界面友好、相关资源丰富的特点。
1.1 基本概念1.1.1 B/S和C/S结构所有基于网络的应用系统都会涉及服务器端和客户端两方面的程序设计工作，依据客户端使用方式的不同，可以将网络应用系统分为两大体系，即B/S结构体系和C/S结构体系。
B/S结构是浏览器/服务器Browser/Server）结构，C/S结构是客户机/服务器（Client/Server）结构。
C/S结构体系是为服务器端和客户端分别开发一个应用程序，在使用时，用户需要将客户端程序首先安装到客户端上，用户要访问服务器端的程序和资源时，必须要通过客户端程序。
目前广泛使用的QQ软件就是一个典型的C/S结构的网络应用系统。
B/S结构体系是仅为服务器端开发一个应用程序，在客户端用浏览器作为客户端程序。
显然，由于使用浏览器作为客户端应用程序，使用者无须下载和安装客户端软件，也无须学习新软件的使用方法就可以使用基于B/S结构的应用系统，这给开发者和使用者都带来了极大的方便。
C/S结构和B/S结构各具特点，C/S结构可以使应用系统功能更强大，效率更高；而B/S结构使用更方便，两者的开发方法和开发工具也不尽相同，就目前的应用情况看，B/S结构体系是应用的主流。
本书介绍的ASRNET+SQL Server组合是一种典型的B/S结构体系的开发工具。
1.1.2 静态网页和动态网页1.1.2.1 静态网页网页可以分为静态网页和动态网页两类，静态页面完全是由HTML（Hyper Text Markup Language，超文本标记语言）代码构成的，在浏览者访问静态网页之前，网页的创建者已经用HTML代码完全确定了页面要显示的内容。
网页创建者可以通过各种文本编辑器，来编写HTML代码，如Windows中的记事本，以扩展名.htm或.html来保存文件即可生成静态网页。
HTML语言的语法规则非常简单，具体规则请参见附录8。
静态网页也有专门的制作工具，如Micromedia公司的Dreamweaver和Microsoft公司的FrontPage等，Dreamweaver和FrontPage都被视为是所见即所得的、功能强大的HTML代码生成器。
1.1.2.2 动态网页动态网页是指浏览者所访问的网页根据客户端请求的情况即时动态地生成所需网页。
在动态网页中，通常会包含HTML代码和指令代码。
动态网页也可以分为两种模式，即所谓的基于客户端的动态网页和基于服务器的动态网页。
1. 客户端模式在客户端模式中，Web服务器根据客户端的请求将相应的HTML和指令代码发送到客户端，附加在客户端浏览器上的指令处理模块处理这些HTML和指令代码，并将其转换为纯HTML代码，交由浏览器显示。
支持客户端模式的主要技术有JavaScript、VBScript、Java小应用程序以及Flash等。
在客户端模式中，能使用何种技术要取决于浏览器上所附加的指令处理模块。
JavaScript、VBScript都是脚本语言，Netscaps公司的Navigator浏览器和3.0版本以后的IE浏览器均支持JavaScript。
VBScript是Microsoft公司推出的脚本语言，能被IE浏览器很好地支持。
IE和Navigator浏览器均通过Java虚拟机（Java Virtual Machine，JVM）的方式来支持Java小应用程序。

Flash也有比较强的交换能力，其使用的Action Script语言和JavaScript非常类似，IE和Navigator浏览器都是通过附加Flash插件的方式实现对Flash的支持。
由于工作机理所限，客户端模式存在着一些固有缺陷，主要表现在两个方面：第一，在设计网页时需要考虑客户端浏览器的情况；第二，当涉及使用服务器资源的代码时，由于这些代码要在客户端被解释执行，会暴露给客户端用户，存在安全隐患。
2. 服务器端模式在服务器端模式中，Web服务器根据客户端的请求处理动态网页所包含的HTML和指令代码，生成纯HTML代码，然后将纯HTML代码发送到客户端，由客户端浏览器显示。
显然该模式从根本上解决了客户端模式下存在的问题，该模式也成为动态网页设计的主流模式。
目前，支持服务器端模式的主要技术有ASP、JSP、PHP等。
这三种技术各具特点，ASP.NET可视为ASP的最新版本，是Microsoft公司.NET阵营中的主要成员之一；JSP是SUN公司J2EE阵营中的一个主要成员。
上述两个阵营代表程序设计技术发展的主要方向，AS P.NET和JSP也成为动态网页设计的主流。
PHP是一种免费的、源代码开放的、具有很好跨平台特性的动态网页设计技术。
PHP+Linux+MvSQL可以构成完全免费而且非常稳定的应用开发平台，该组合一度非常流行，但也正是由于其是免费的，没有很好的技术支持，更新速度比较慢，现在已渐渐淡出开发领域。
1.1.3 Web服务器的概念在基于B/S结构的应用系统中，所有程序和数据都被置于Web服务器之上，显然Web服务器的性能有着举足轻重的作用。
一般讲Web服务器有两重含义，第一是指在网络上提供Web服务的计算机，此时Web服务器的概念包括硬件和软件两部分；第二是指处于网络操作系统之上的，专门用于提供Web服务的软件。
本书中提到的Web服务器是指后者。
目前市场上可以选择的Web服务器产品有十几种之多，其中比较著名的有Apache和IIS Web服务器。
Apache是一款使用广泛、品质卓越的Web服务器，在Web服务器领域处于领导地位，是UNIX和Linux平台下首选的Web服务器。
IBM和HP公司的操作系统都捆绑了Apache作为其W曲服务器组件。
IIS（Internet Information Server）是Windows操作系统的组件之一，理所当然地成为Windows平台下首选的Web服务器，IIS提供了强大的Internet和Intranet服务功能，IIS除可提供Web服务外，还可以提供诸如FTP服务、NNTP服务、SMTP服务等其他一些服务。
本书使用Windows 2000作为操作系统平台，以IIs作为Web服务器。
1.1.4 HTTP协议HTTP（Hypertext Transfer Protoc01）协议即超文本传输协议，是Web服务的基础。
该协议属于TCP/IP协议族中的应用层协议，主要用于Web浏览器和Web服务器之间的超文本传输，是使用最广泛的应用层协议之一。
HTTP于1990年提出，经历了不断的使用与发展，目前广泛使用的是HTTP1.0的第六版。
HTTP 1.1的规范工作也在进行之中，部分Web服务器和浏览器也开始支持HTTP1.1，如IIS就支持HTTP1.1协议。
HTTP协议使用TCP连接，是一个支持客户机/服务器模式，具有简单高效、灵活性好、无连接、无状态等特点的应用层协议。
在HTTP协议环境下，客户端向服务器请求服务时只需传送请求方法和路径。
常用的请求方法有GET、HEAD、POST，多种方法的使用使客户端与服务器间可以用不同的方式联系。
由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。
HTTP允许传输任意类型的数据对象，所传输的数据类型由Content—Type加以标记。
HTTP每次连接只处理一个请求，服务器处理完客户的请求并收到客户的应答后即断开连接，这种方式称为无连接，采用无连接方式可以节省传输时间。
HTTP是一个无状态协议，它不对已处理的事务进行记忆。
在服务器不需要前面信息的情况下，无状态协议的应答较快，但如果后续处理需要前面的信息则必须重传，这样可能导致每次连接传送的数据量增大。
有关HTTP协议的详细内容读者可以参阅RFC 1945或文献[1]。

编辑推荐

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