

信息科学与技术丛书

程序设计系列

Visual C# 2005

智能设备程序设计

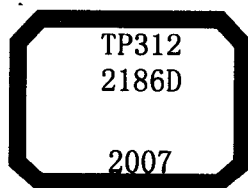
姜波 宁峰 等编著

- ◎ Visual Studio 2005 开发工具的使用
- ◎ C# 编程语言的实际运用
- ◎ 智能设备程序设计的详细过程
- ◎ “连连看”游戏程序的完整代码展示
- ◎ 动物管理员(Zookeeper)及“连连看”游戏程序的代码详解



机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书
程序设计系列



Visual C# 2005 智能设备程序设计

姜波 宁峰 张馨月 等编著



机械工业出版社

本书是智能设备 (Smart Device) 及 Visual C# 2005 开发和学习的指南。

全书共 3 个部分。第 1 部分介绍了 C# 语言的基本语法、Visual Studio 2005 的平台特性以及面向对象程序设计的特点。第 2 部分讲解了如何利用面向对象的程序设计方式, 使用 Visual C# 2005 进行智能设备程序开发的过程。还介绍了如何通过循序渐进的开发方法, 逐步实现“连连看”游戏程序功能。第 3 部分是在“连连看”游戏的基础上, 介绍了如何利用面向对象程序设计的思想, 进行 Zookeeper (动物管理员) 游戏程序实例的开发。

本书适合 C# 及 Smart Device 的初、中级程序员学习, 也可作为各高校或培训机构的授课教材。

图书在版编目 (CIP) 数据

Visual C# 2005 智能设备程序设计 / 姜波等编著. —北京: 机械工业出版社, 2006.10

(信息科学与技术丛书. 程序设计系列)

ISBN 7-111-20082-9

I. V... II. 姜... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 124069 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 罗子超

责任印制: 李 妍

北京地质印刷厂印刷

2007 年 1 月第 1 版·第 1 次印刷

184mm×260mm·17.75 印张·438 千字

0001—5000 册

定价: 32.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68326294

编辑热线电话 (010) 88379739

封面无防伪标均为盗版

出版说明

随着信息科学技术的迅速发展，人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书，来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术受到不断挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着因特网延伸到人类世界的层层面面，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容，注重理论与实践相结合，内容实用，层次分明，语言流畅，是信息科学与技术领域专业人员不可或缺的图书。

现今，信息科学技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

如果说 20 世纪被称为 PC 时代的话,那么 21 世纪则已经进入到了“后 PC 时代”。在这个时代当中,无所不在的嵌入式技术无疑是最具代表性的技术,而智能设备又是与我们关系最为紧密的产品之一。从应用在工业领域的红外线测温仪,到人们日常使用的手机,我们几乎每天都在和智能设备打着交道。作为一名程序设计人员,又该如何利用现有的技术进行智能设备程序设计呢?这就是本书要解决的问题。

选择 C#作为程序设计语言,将 Visual Studio 2005 作为程序开发环境,是因为 Visual Studio 2005 是微软公司于 2005 年 12 月推出的全新的 Windows 开发工具。它继承了 Visual Studio 开发工具在使用上的便利与快捷,同时还增加了许多新的特性,尤其是加强了对智能设备的支持。而 C#作为一种面向对象的编程语言,不但实现了程序的快速开发,跨平台部署,与 Web 平台的紧密结合等,其对智能设备的良好支持也使得它受到了越来越多程序开发人员的欢迎。

本书的特点

1. 本书打破了以往以 C#语法为主的讲解模式,而是采用了“实例+代码讲解+原理分析”的方式,力求从实际开发的角度全面展现使用 C#进行智能设备程序设计的过程。
2. 本书不同于其他书籍中介绍的简单棋类游戏,或者简单的计算器程序实例,而是介绍了目前网络上非常流行的“连连看”游戏和“动物管理员 (Zookeeper)”游戏实例。
3. 为了使读者能够更好地掌握书中的内容,本书还对涉及到的基本知识点进行了详细的介绍。比如, C#的基本语法、面向对象的程序设计思想等。
4. 在“动物管理员 (Zookeeper)”程序实例中,采用了“引导教学”的讲解方式,使读者不但可以学习程序设计的过程,加深对程序的理解,还可以极大地提高读者的实际动手能力。
5. 本书通俗易懂、内容实用、层次分明、语言流畅、注意理论与实践相结合,是程序员们学习的好帮手。
6. 随书附赠的光盘中包含了各个版本“连连看”程序的完整工程文件,同时配有完整的视频演示。

本书适合的读者

作为一本程序设计类图书,本书适合于各大高校的在校学生、从事游戏程序设计的开发人员、从事智能设备程序设计的开发人员以及热衷于 C#的读者。本书可作为各高校和培训机构的授课教材。

本书内容的组织

根据讲授内容的难易程度，本书主要包括以下几个部分：

1. 基础理论部分：这一部分包括第 1~4 章，主要为读者介绍了 Visual Studio 2005 的特点、智能设备程序设计的特点、C#的基本语法及面向对象程序设计的基本思路。
2. 初级程序实例部分：这一部分包括第 5~12 章，主要为读者介绍了采用 C#作为程序设计语言，在 Visual Studio 2005 环境下，开发 Pocket PC 版“连连看”的过程。
3. 高级程序实例部分：这一部分为第 13 章，主要为读者介绍了如何在“连连看”的程序框架下，构建出一个新的智能设备程序“Zookeeper”。

致谢

本书主要由姜波、宁峰、张馨月编写，参加本书编写工作的还有李淑云、马一宁、宁立君、曹淦参、王志斌、魏冰、赵雪峰、陈秀丽、于文、北京博创兴业科技有限公司的邹莹、杨亮、陶永以及北京.NET 俱乐部副主席葛涵涛。

书中难免出现不足之处，请各位读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 Visual Studio 2005 简介 1

1.1 微软.NET 战略的提出 1

1.1.1 .NET 战略提出的背景 1

1.1.2 .NET 战略带来的影响 2

1.2 Visual Studio 版本的演化
及其特点 2

1.2.1 Visual Studio.NET 2002 和
2003 简介 2

1.2.2 Visual Studio 2005 简介 4

第 2 章 智能设备程序设计简介 8

2.1 智能设备简介 8

2.1.1 Windows CE 简介 8

2.1.2 Pocket PC 和 Pocket PC Phone
Edition 简介 9

2.1.3 Smart Phone 简介 10

2.2 智能设备程序的特点 11

2.2.1 API 函数和 MFC 11

2.2.2 硬件特性导致的程序设计上的
差异 11

2.2.3 程序的移植 12

2.3 智能设备版“Hello World” 12

2.3.1 程序的构建 12

2.3.2 添加代码并运行该程序 14

第 3 章 C#语法简介 16

3.1 Visual C# 2005 简介 16

3.2 数据类型 18

3.2.1 预定义类型中的值类型 18

3.2.2 预定义类型中的引用类型 21

3.2.3 复合类型中的值类型 23

3.2.4 复合类型中的引用类型 27

3.2.5 类型转换 36

3.3 变量 40

3.4 变量的范围 41

3.5 变量修饰符 43

3.5.1 访问修饰符 43

3.5.2 静态变量修饰符和实例变量
修饰符 44

3.5.3 只读变量修饰符 45

3.6 变量与参数 45

3.7 常量 47

3.8 运算符 47

3.8.1 赋值运算符 48

3.8.2 条件运算符 49

3.8.3 checked 和 unchecked 运算符 49

3.8.4 is 运算符 49

3.8.5 typeof 运算符 49

3.8.6 sizeof 运算符 50

3.8.7 运算符的简化操作与优先级 50

3.9 程序控制语句 51

3.9.1 if 语句 51

3.9.2 switch 语句 52

3.9.3 while 和 do...while 语句 54

3.9.4 for 语句 54

3.9.5 foreach 语句 55

3.9.6 跳转语句 55

3.9.7 try 语句 57

3.10 程序的结构 57

3.10.1 类 57

3.10.2 命名空间 59

3.10.3 Main 方法 60

第 4 章 面向对象程序设计简介 61

4.1 面向对象概述 61

4.1.1 传统软件设计方法存在的问题 61

4.1.2 面向对象程序方法的特点 62

4.1.3 面向对象程序设计方法中涉及的
基本概念 62

4.2 对象和类 63

4.2.1 对象的概念 63

4.2.2 面向对象中的类	64	6.4.2 游戏类	114
4.3 面向对象中的继承	64	第7章 在“连连看”中添加卡片	
第5章 游戏实例——“连连看”	66	选中标示	119
5.1 游戏介绍	66	7.1 新添加的选中标示	119
5.2 “连连看”游戏程序的		7.2 选中标示类的添加所引起的	
设计步骤	66	代码变化	119
5.2.1 程序画面的实现	66	7.2.1 变化后的卡片集合类	120
5.2.2 程序代码的添加	68	7.2.2 变化后的游戏类	126
5.3 基类源代码详解	79	7.3 新添加的代码模块	130
5.3.1 成员变量的含义	80	7.4 代码变化的类及新添加的模块	
5.3.2 基类的构造函数	80	代码详解	131
5.3.3 基类中其余函数代码的解释	81	7.4.1 选中标示类	131
5.4 卡片类源代码详解	83	7.4.2 卡片集合类	133
5.4.1 成员变量的含义	84	7.4.3 游戏类	138
5.4.2 卡片类的构造函数	84	第8章 在“连连看”中添加计时	
5.4.3 卡片类中其余函数代码的解释	85	功能	139
5.5 卡片集合类源代码详解	85	8.1 程序计时功能简介	139
5.5.1 成员变量的含义	85	8.2 计时器类的添加	139
5.5.2 卡片集合类的构造函数	86	8.3 代码变化的类及新添加的模块	
5.5.3 与卡片处理、绘制、重置相关的		代码详解	141
函数	86	8.3.1 计时器类	141
5.5.4 游戏程序中对屏幕单击的处理	88	8.3.2 游戏类	143
5.5.5 判断两张卡片之间是否可以连通	89	8.4 计时功能中时间显示方式的	
5.6 游戏类源代码详解	95	改变	144
5.6.1 成员变量的含义	95	8.4.1 变化后的时间显示方式	144
5.6.2 游戏程序的初始化	95	8.4.2 时间显示方式的代码变化	144
5.6.3 游戏对象的处理	97	8.4.3 代码变化后的计时器类代码	
5.6.4 游戏程序的主函数	98	详解	146
第6章 游戏设计中游戏状态的		8.4.4 代码变化后的游戏类代码详解	148
添加	100	第9章 卡片闪烁、飞散及旋转标示	
6.1 游戏状态概述	100	的实现	149
6.2 “连连看”游戏程序状态的		9.1 游戏特效概述	149
介绍	101	9.2 特效添加所引起的代码变化	150
6.3 程序状态的添加所引起的代码		9.2.1 卡片类	151
变化	102	9.2.2 卡片集合类	153
6.3.1 卡片集合类	102	9.3 变化后的卡片类源代码详解	160
6.3.2 游戏类	108	9.3.1 新添加的全局变量和常量	160
6.4 变化部分代码详解	113	9.3.2 函数的代码变化	160
6.4.1 卡片集合类	113	9.3.3 新添加的函数	161

9.4 变化后的卡片集合类源代码 详解	163	13.1 游戏介绍	219
9.5 变化后的选中标示类源代码 详解	165	13.1.1 游戏画面展示	219
9.6 变化后的计时器类源代码 详解	166	13.1.2 Zookeeper 的游戏规则	220
9.7 变化后的游戏类源代码 详细解	167	13.2 Zookeeper 游戏程序的框架 结构	222
第 10 章 连接类的抽象	168	13.2.1 Zookeeper 游戏程序所 包含的类	222
10.1 抽象出的连接类	168	13.2.2 和“连连看”中相应类的对应 关系	224
10.2 连接类的抽象所引起的代码 变化	169	13.3 基类关键源代码详解	224
10.2.1 卡片集合类	169	13.3.1 基类中的变量	225
10.2.2 新添加的连接类	174	13.3.2 基类中的相关函数	225
10.2.3 新添加的连接集合类	179	13.4 动物类关键源代码详解	227
10.3 新添加的连接类源代码详解	181	13.4.1 全局变量和常量	227
10.3.1 Link 类中全局量的含义	181	13.4.2 构造函数、动物类型和状态的 读取及设置	228
10.3.2 Link 类的构造函数	182	13.4.3 处理对象的函数	230
10.3.3 Draw 函数	183	13.4.4 其余相关函数	233
10.3.4 判断卡片连接相关的函数	183	13.5 动物集合类关键源代码详解	235
10.4 新添加的连接集合类源代码 详解	187	13.5.1 全局变量和常量	235
10.5 代码变化的类源代码详解	191	13.5.2 构造函数	236
10.5.1 卡片集合类	191	13.5.3 下一状态设置, 动物对象数组 处理、重置及绘制函数	237
10.5.2 游戏类	195	13.5.4 处理智能设备屏幕单击的 函数	239
第 11 章 “梦幻闪电”特效的实现	197	13.5.5 状态切换函数	241
11.1 “梦幻闪电”特效概述	197	13.5.6 实现动物卡片位置交换的 函数	243
11.2 特效添加所引起的代码变化	198	13.5.7 实现动物卡片填充的函数	245
11.3 代码变化的类源代码详解	203	13.5.8 实现动物卡片移除的函数	246
11.3.1 基类	203	13.6 动物计数器类关键源代码 详解	250
11.3.2 卡片类	204	13.6.1 构造函数及相关属性的实现	251
11.3.3 连接类	205	13.6.2 动物卡片计数、绘制和重置 函数	252
第 12 章 “飞舞碰撞”特效的实现	209	13.7 数字类关键源代码详解	254
12.1 “飞舞碰撞”特效概述	209	13.7.1 数字类中的构造函数和返回/设置 数值函数	254
12.2 特效添加所引起的代码变化	210	13.7.2 Draw 函数	255
12.3 代码变化的类源代码详解	216		
12.3.1 卡片类	216		
12.3.2 连接类	216		
第 13 章 游戏实例——Zookeeper	219		

13.7.3 重置和增加函数	256	13.12.2 构造函数及返回计时器的值 ...	261
13.8 游戏程序目标类关键源代码		13.12.3 与时间处理相关的函数	261
详解	256	13.12.4 时间条的绘制	262
13.9 游戏程序等级类关键源代码		13.12.5 重置及实现计时时间延长的	
详解	257	函数	263
13.10 游戏程序分数类关键源代码		13.13 游戏类关键源代码详解	264
详解	258	13.13.1 全局变量	264
13.11 选中标示类关键源代码		13.13.2 完整源代码	265
详解	258	13.13.3 构造函数及属性方法	271
13.11.1 选中标示类中用到的属性		13.13.4 初始化函数	271
方法	259	13.13.5 游戏的主循环	272
13.11.2 选中标示类中的重置、选中/		13.13.6 初始化游戏程序和处理所有	
取消选中函数	259	对象	273
13.12 计时器类关键源代码详解	260	13.13.7 绘制及重置所有游戏对象	273
13.12.1 全局常量和变量	260		

第1章 Visual Studio 2005 简介

1.1 微软.NET 战略的提出

提到微软的.NET 战略,那些较早关注 IT 动态的读者一定会非常熟悉。尽管微软现在已经闭口不提当年极力推广的.NET 战略,但是作为 IT 发展史上一个重要的里程碑来说,正是由于该战略的推出,直接导致了后来被广大程序设计人员广泛使用的 Visual Studio.NET 集成开发环境的诞生。本书的全部内容都是在 Visual Studio 平台下进行程序设计的,因此在学习具体的程序设计之前,先来回顾一下这段发展历程,简单了解一下微软的.NET 战略。

1.1.1 .NET 战略提出的背景

当年美国总统克林顿提出信息高速公路概念的时候,或许还没有想到互联网会发展到今天这个样子。网络不仅仅使人们获得信息的途径大大增加,更为便利,也在一定程度上影响了每一个人的生活方式。那些经常使用电脑的读者一定会有这样的感受,当电脑可以连接到网络的时候,我们会觉得有好多事情可以做,一旦网络失去连接,我们就会觉得自己的工作无从下手。正是在这种背景下,微软推出了.NET 战略。

微软推出该战略的目的就是希望帮助用户,能够在任何时候、任何地方、利用任何工具都可以获得网络上的信息,并享受网络通信所带来的快乐。微软.NET 平台的基本思想是:

- 侧重点从连接到互联网的单一网站或设备上,转移到计算机、设备和服务群组上,使其通力合作,提供更广泛、更丰富的解决方案。
- 用户将能够控制信息的传送方式、时间和内容。计算机、设备和服务将能够相辅相成,从而提供丰富的服务。
- 企业可以提供一种方式,允许用户将它们的产品和服务无缝地嵌入到自己的电子构架中。

.NET 平台的突破性改进在于以下几点:

- 使用统一的 Internet 标准(如 XML)将不同的系统对接。
- 这是 Internet 上首个大规模的高度分布式应用服务架构。
- 使用了一个管理程序,全面地管理平台中运行的各种服务,并且为它们提供强大的安全保护后台。

最初发布的.NET 平台包含了以下一些核心组件:

- 一组用于创建互联网操作系统的构建块,其中包括 Passport.NET(用于用户认证)以及用于文件存储的服务、用户首选项管理、日历管理以及众多的其他任务。
- 构建和管理新一代服务的基本结构和工具,包括 Visual Studio.NET、.NET 企业服务器、.NET 框架和 Windows.NET。

- 能够启用新型智能互联网设备的.NET 设备软件。

正是由于这些特性,加上微软的极力推广,使得.NET 战略在初期得到了很多公司和程序设计人员的追捧。

1.1.2 .NET 战略带来的影响

我们来看一下.NET 战略对开发人员、IT 专业人员,以及企业应用的巨大影响。

对于开发人员来说,.NET 的策略是将互联网本身作为构建新一代操作系统的基础,对互联网和操作系统的设计思想进行合理延伸。这样,开发人员必将创建出摆脱设备硬件束缚的应用程序,以便轻松实现互联网连接。.NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

这不但改变了开发人员开发应用程序的方式,而且使得开发人员能创建出全新的应用程序。新型开发范例的核心是 Web 服务概念的引入。Web 服务是一种通过简单对象访问协议(SOAP),在互联网上展现其功能性的、极为公开的服务。SOAP 是一种基于可扩展标记语言(XML)制定的协议。

在过去,开发人员通过集成本地系统服务来构建应用程序。在这种模型下,开发人员可以访问丰富的开发资源,并能严格控制应用程序的行为。而在.NET 战略推出以后,开发人员在很大程度上挣脱了这种模型的束缚,致力于构建具有复杂结构的 n 层化系统。这种系统能将网络上众多的应用程序一并进行集成,大大提升了应用程序的价值。这样,开发人员便可以把精力集中在充分挖掘软件独特的商业价值上,而不是构建基本结构上。

从理论上讲,开发人员可通过调用 Web 应用编程接口(API),将 Web 服务集成到应用程序中。其调用方法与调用本地服务类似,不同的是,Web API 调用可通过互联网发送给位于远程系统中的某一服务。例如,Passport 服务使得开发人员能够对应用程序进行认证。通过对 Passport 服务编程,开发人员可以充分利用 Passport 的基本结构,通过运行 Passport 来维护用户数据库,以确保其正常运行、定期备份等。

.NET 平台还可以从根本上改善计算机和用户之间进行交互的方式,最大限度地发挥电子商务中计算技术的重要作用。它还可以保证完全消除当今计算技术中的缺陷,能够实现确保用户从任何地点、任何设备都可访问其个人数据和应用程序。除此之外,.NET 技术还可实现多个应用程序在逻辑上的松散耦合链接和紧密耦合链接。用户可以通过手写、语音和图像技术与其个人数据进行交互。这些数据将安全地存放在互联网上,用户通过 PC、移动电话、寻呼机或 PDA 等设备访问这些数据。应用程序可进行灵活的功能调整,以适应用户所用设备的功能状况。应用程序可根据用户预定义的选项集和指令集,完全代替用户自动执行相应的操作。

1.2 Visual Studio 版本的演化及其特点

为了配合微软.NET 战略的推出,微软公司推出了一套全新的集成开发环境,那就是 Visual Studio.NET。下面介绍一下.NET 集成开发环境的演化过程。

1.2.1 Visual Studio.NET 2002 和 2003 简介

2002 年 3 月 22 日,微软发布了微软 Visual Studio.NET 2002 和 .NET Framework。它是微

软发布的第一款基于.NET 战略的集成开发环境。

.NET 最核心的部分是处于 .NET Framework 底层的通用语言 (Common Language Runtime, CLR), 它是 20 多种语言的统一运行环境, 提供了内存管理、例外处理、安全性、版本控制和组件通信等系统服务。Visual Studio.NET 充分发挥了 CLR 的潜力, 为开发者提供了一个统一的集成开发环境和调试器。由于采用统一类型定义和共享类库, 各语言不仅在运行时, 而且在设计时就可以实现对象级的交互。

统一的运行环境及共享类库使.NET 支持的两种主要应用: ASP.net 和 Windows Form 应用可以共享大部分代码; 而 VS.NET 和 CLR 让开发人员摆脱系统底层细节, 留给开发者的是充分发挥自己的想象空间。完整的.NET 平台还包括服务器、基础服务、客户设备支持和用户体验。VS.NET 提供的是一个集成的环境, 通过整合这些资源和服务, 实现最终的用户体验。

在.NET 平台中, Web Services 已经成为微软.NET 的核心。微软定义的下一代分布式计算主要由 3 个重要步骤组成: 所有应用都是 Web Services; 汇集整合所有 Web Services; 为用户提供简单而有深度的 Web 体验。

微软声称: “Visual Studio.NET 是目前最快、最好的 Web Services 开发工具”。一方面是由于微软在开发.NET 及 VB.NET 时就内置了 Web Services 支持; 另一方面 VS.NET 不仅提供 Web Services 开发工具, 而且提供从 Web Services 的开发到发布、注册、整合的全过程支持。

为创建和部署 Web Services, .NET 平台采用一系列标准的互联网协议, 如 XML、SOAP、WSDL 和 UDDI 等, 其中, XML 是基础和核心。VS.NET 对 XML 支持全面而直接。在 VS.NET 中, 开发人员能“即见即所得”地创建 XML 文档; 并通过 XML 文档实例来自动生成 XML Schema、解析 XML 文件。

在组件尤其是中间件的市场, 长期以来一直是 Java 的天下, “一次编写, 到处运行”的承诺使基于 J2EE 和 EJB 的组件技术成为市场主导产品。而微软.NET 则试图进入面向组件的开发。无论是面向组件的语言 (如 C#、VB.NET 和 C++等), 还是传统的语言 (如 COBOL 等), 在.NET 平台中, 都可以产生基于微软中间语言 (MSIL) 的组件, 而这些组件在运行时具有对象级的交互能力。

同时, Visual Studio.NET 充分利用了底层的 CLR 组件支持, 并提供了一整套开发工具使组件开发简单而高效, 用户只需关心组件的商业逻辑, 而由 VS.NET 来完成其余复杂的工作。C#面向组件开发, 内置专门针对组件开发的多种特性, 如对属性、委托、声明的支持等, 使开发基于.NET 组件得心应手。

.NET 组件技术最重要的特点可能就是其组件具有“即插即用”的能力。支持直接开发具有自描述性的组件, 使组件部署无需人工干预。微软试图借助 Visual Studio.NET 2002 在组件市场重新定位。

软件人员最关心的还是开发效率的提高, Visual Studio.NET 2002 的人性化界面和众多工具将成倍提高开发效率。无论针对传统的 Windows 桌面开发, 还是 Web Services, 开发人员都不必再为不同的应用重写代码, 不必为不同的客户设备定制不同的界面。

为了进一步巩固.NET 战略, 同时修正前一代产品 (Visual Studio.NET 2002) 中的一些 Bug, 微软公司在 2003 年又推出了其升级版 Visual Studio.NET 2003。同时.NET Framework 的版本也从 1.0 升级到了 1.1。和前一个版本相比, VS.NET 2003 并没有大的变动。

如图 1-1 所示为 Visual Studio.NET 2003 的开发环境。

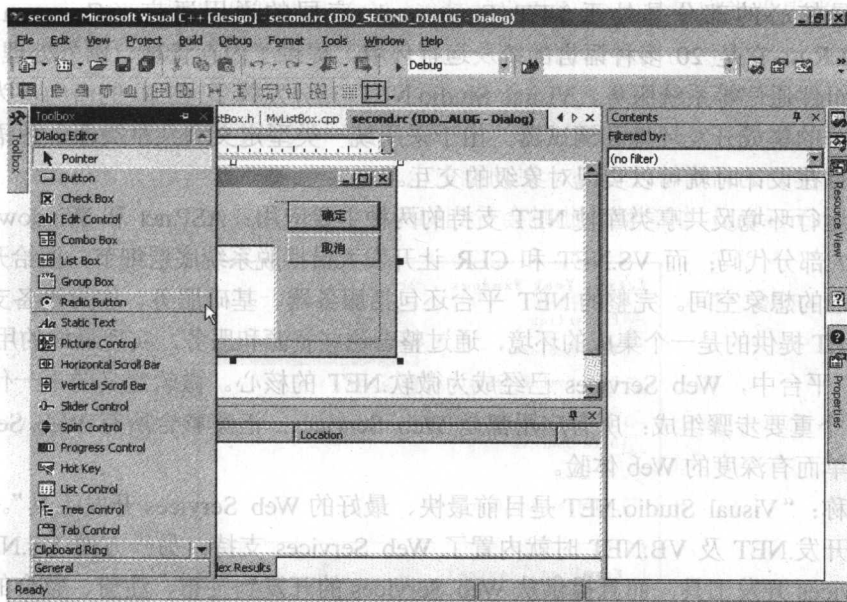


图 1-1 Visual Studio.NET 2003 的开发环境

1.2.2 Visual Studio 2005 简介

Visual Studio 2005 是微软公司在 2005 年推出的一款集成开发环境。同 Visual Studio.NET 2003 相比, Visual Studio 2005 是基于 .NET Framework 2.0 的, 而前面的版本则是支持 .NET Framework 1.1 的。 .NET Framework 2.0 主要新增了以下一些特性:

- 加入了对 64 位平台的支持。
- ADO.NET 支持用户定义类型 (UDT)、异步数据库操作、XML 数据类型、大值类型和快照隔离等。
- ASP.NET 新增了很多有用的控件, 改善代码隐藏模型, 缓存功能提供了多种新方式, 如在 SQL SVR 数据库的表上生成缓存依赖项的能力。
- 配置文件属性使 ASP.NET 能自动跟踪单个用户的属性值, 使用简单控件添加导航菜单。
- 增强了与 COM 的互操作性。
- 增加 Console 控制。
- 在 System.NET 命名空间中, 添加了对 FTP、HTTP 资源缓存, Web 请求处理等相关增强的功能。
- 应用程序可使用新的 NegotiateStream 和 SslStream 类进行身份验证, 并帮助保护在客户端和服务端之间传输的信息的安全。
- System.IO.Compression 命名空间中的类以 GZIP 压缩解压标准读写数据。
- System.NET.NetworkInformation 命名空间中的类, 应用程序可以访问 IP、IPv4、IPv6、TCP 和 UDP 网络流量的统计信息及类似 ipconfig 命令返回的信息。
- NET work Change 类通过在检测到地址更改时引发事件来提供地址更改通知。

- System.NET.Cache 命名空间中的类，可以使用各种缓存策略。
- Ping 类提供类似于 ping 命令提供的功能，允许同步/异步调用。
- SerialPort 类提供访问计算机串口及与串口 I/O 设备进行通信的能力。
- System.NET.Mail/System.NET.Mime 下的类可以方便发送电子邮件。
- 泛型和泛型集合，包括反射对泛型的支持，使得代码更灵活、可重用性更高。

除了上面提到的新增特性以外，在功能上也作了一些调整。当启动 Visual Studio 2005 以后，可以按〈Ctrl+Tab〉组合键，此时会看到如图 1-2 所示的界面。

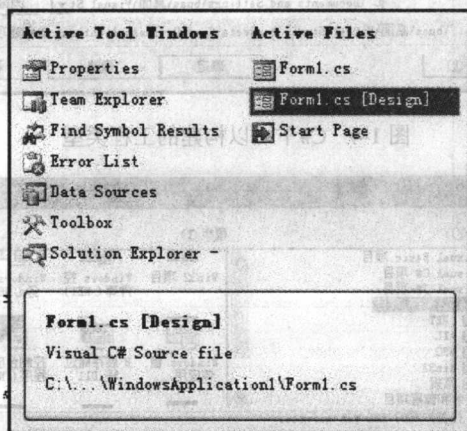


图 1-2 功能切换界面

在按住〈Ctrl+Tab〉组合键的同时，可以通过方向键实现光标选项的移动，来实现对不同功能和板块的查看。

本书的程序实例运行在 Smart Device 中的 Pocket PC 上，下面介绍对智能设备的支持上的变化。

在 Visual Studio.NET 2003 中，用户会看到只有 VB 和 C# 两种编程语言支持智能设备，如图 1-3 所示为 VB 下可以构建的工程类型；如图 1-4 所示为 C# 下可以构建的工程类型；如图 1-5 所示为 C++ 可以构建的工程类型。

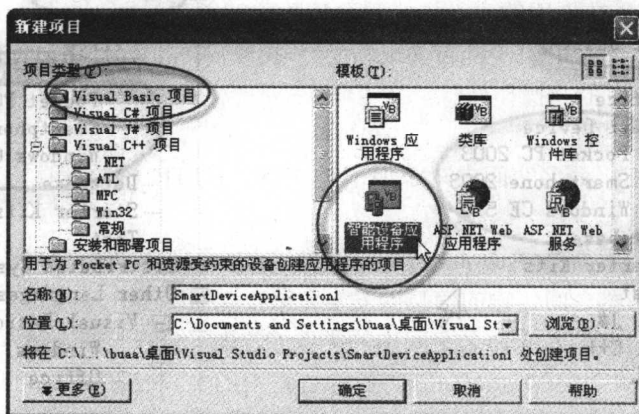


图 1-3 VB 下可以构建的工程类型

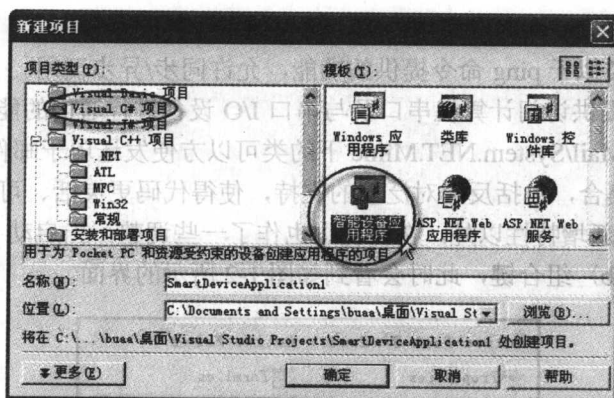


图 1-4 C#下可以构建的工程类型

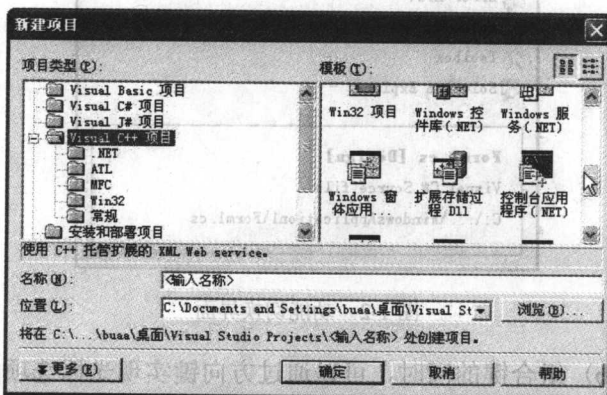


图 1-5 C++下可以构建的工程类型

而在 Visual Studio 2005 中，除了 C#和 VB 以外，C++中也加入了智能设备的支持。如图 1-6 所示为 Visual Studio 2005 下 VB 可以构建的工程类型；如图 1-7 所示为 Visual Studio 2005 中 C#可以构建的工程类型；如图 1-8 所示为 Visual Studio 2005 中 C++可以构建的工程类型。

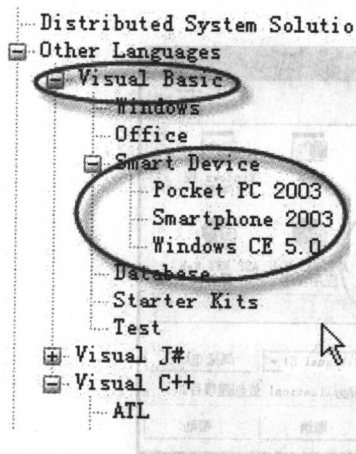


图 1-6 VB 下可以构建的工程类型

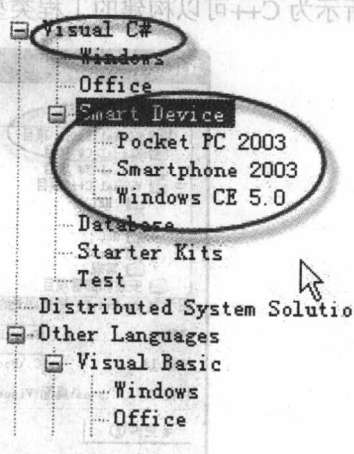


图 1-7 C#下可以构建的工程类型

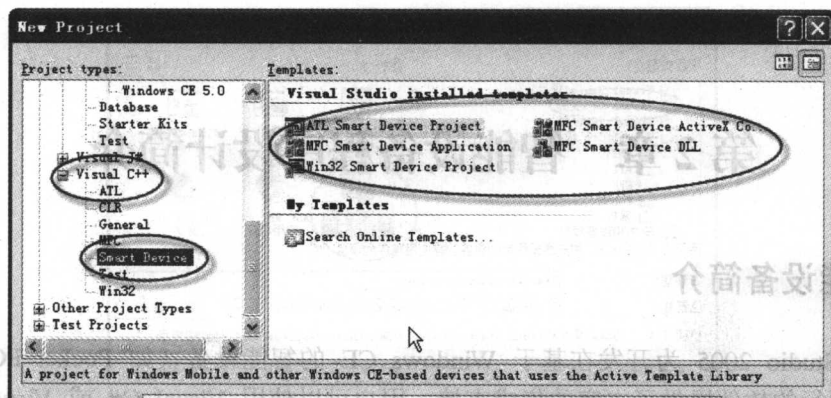


图 1-8 C++下可以构建的工程类型

如果用户在 Visual Studio 2005 的“智能设备”模板库中选择一个模板，那么将使用这个模板来创建项目。在“智能设备”模板库中，用户可以找到各种类型的模板，包括“ATL Smart Device Project”、“MFC Smart Device Application”和“Win32 Smart Device Project”。这些模板都是为 Windows CE 5.0 设计的，可以用于开发各种类型的 Windows CE 应用程序。在“智能设备”模板库中，用户还可以找到一些其他的模板，如“ATL”、“CLR”、“General”和“Win32”。这些模板都是为 Windows CE 5.0 设计的，可以用于开发各种类型的 Windows CE 应用程序。

这里提到的 Smart Device 主要包括 Windows CE、Pocket PC 和 Smart Phone。下面介绍各种产品的特点。

2.1.1 Windows CE 简介

Windows CE 是一个典型的嵌入式操作系统，它最初是为掌上型设备、移动应用、消费电子产品和嵌入式系统设计的。它最初是由微软公司开发的，后来被其他公司广泛采用。

“CE”这个名字的来源已无从考证，但有些专家认为，CE 中的 C 代表了“Consumer”（消费者），E 代表了“Embedded”（嵌入式）。如果把 CE 看作是“Computer Embedded”（嵌入式计算机）的缩写，那么这个名字就很有意义了。

嵌入式操作系统是一种用于控制工业过程、仪器、设备等的操作系统。它通常具有实时性、可靠性、小型化等特点。在嵌入式系统中，操作系统需要与硬件紧密配合，以实现各种功能。Windows CE 是一种典型的嵌入式操作系统，它广泛应用于各种嵌入式设备中。随着嵌入式系统的不断发展，Windows CE 也在不断地进行更新和改进，以满足用户的需求。

Windows CE 的设计目标是：提供高性能、实时性、可靠性、小型化等特点。它支持多种 CPU 等。从操作系统的角度来看，Windows CE 具有灵活的内存管理机制、低功耗、体积小等特点。它支持多种文件系统（如 FAT、NTFS 等），并提供了丰富的 API 接口。此外，Windows CE 还支持多种网络协议（如 TCP/IP、Modem 等），并提供了强大的图形用户界面（GUI）。这些特点使得 Windows CE 成为了一种非常优秀的嵌入式操作系统。

Windows CE 良好的通信能力主要来自于它对各种通信硬件的支持。它支持多种通信协议（如 RS-232、RS-485、CAN 等），并提供了丰富的 API 接口。此外，Windows CE 还支持多种网络协议（如 TCP/IP、Modem 等），并提供了强大的图形用户界面（GUI）。这些特点使得 Windows CE 成为了一种非常优秀的嵌入式操作系统。