

1+1 数据库混合开发技术丛书

Visual C++ + SQL Server

数据库开发与实例

李 涛 刘凯奎 王永皎 编著



本书数据库开发案例

- 回 客户资源管理系统
- 回 企业备忘录管理系统
- 回 图书馆管理系统
- 回 设备管理系统
- 回 员工培训管理系统
- 回 公司贸易管理系统
- 回 餐饮管理系统
- 回 宾馆管理系统



清华大学出版社

TP312
2063D

1+1 数据库混合开发技术丛书

Visual C++ + SQL Server

数据库开发与实例

李 涛 刘凯奎 王永皎 编著

清华大学出版社

北 京

内 容 简 介

Visual C++ 6.0 是 Microsoft Visual Studio 开发组件中最为强大的编程工具, 由于其强大的功能、友好的开发环境, 一经推出便广受好评, 直到今天仍然是 Windows 平台上应用程序开发的利器; 而其与 SQL Server 2000 的结合更是程序开发的黄金组合。

本书共 12 章, 系统地介绍了 Visual C++ 6.0 与 SQL Server 2000 的基本操作, 并详细介绍了使用 Visual C++ 开发数据库应用程序的方法, 具体包括 Visual C++ 6.0 开发环境、数据库基本原理、SQL Server 2000 操作规范、数据库访问接口概述、应用程序开发流程、数据库及应用程序的优化以及发布等内容。本书的光盘中附有全部实例的源代码和安装方法介绍, 读者可以根据需要在这些示例代码的基础上进行二次开发。

本书作为使用 Visual C++ 进行数据库开发的实例教程, 其重点明确、结构合理、语言简明, 既可作为高等院校 Visual C++ 程序设计课程的教材, 也是工程技术人员进行数据库开发的优秀参考书。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual C++ + SQL Server 数据库开发与实例 / 李涛, 刘凯奎, 王永皎 编著. —北京: 清华大学出版社, 2006.7
(1+1 数据库混合开发技术丛书)

ISBN 7-302-13217-8

I. V … II. ①李 … ②刘… ③王… III. ①C 语言—程序设计 ②关系数据库—数据库管理系统, SQL Server—程序设计 IV. ①TP312 ②TP311.138

中国版本图书馆 CIP 数据核字(2006)第 064850 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 王 定

文稿编辑: 鲍 芳

封面设计: 久久度文化

版式设计: 康 博

印 刷 者: 北京四季青印刷厂

装 订 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 195×260 印张: 27 字数: 657 千字

版 次: 2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷

书 号: ISBN 7-302-13217-8/TP·8356

印 数: 1~5000

定 价: 48.00 元(含光盘)

前言

Visual C++是 Microsoft Visual Studio 开发组件中最为强大的编程工具。在数据库领域,其表现也异常突出,它不仅提供了多种数据库开发技术,支持几乎所有的数据库系统,如 SQL Server, DB2, Sybase, FoxPro, Access 等,还提供了良好的开发环境和丰富的文档支持。

SQL Server 2000 是在 SQL Server 7.0 的基础上扩展升级的产品,是一款面向高端的数据库管理系统,它不仅继承了 SQL Server 7.0 版的稳定性、便于管理和易于开发等优势,同时又具有许多新特性,赢得了众多客户的喜爱。

本书面向初、中级用户,由浅入深地讲解了使用 Visual C++和 SQL Server 共同开发数据库系统的技术。全书共 12 章,内容包括 Visual C++及数据库访问基础的介绍,数据库基础及 SQL 语言的讲解,以及各种实例的具体应用。

第 1 章讲解了 Visual C++及数据库访问基础。首先讲解了面向对象的程序设计,以及 Visual C++的基本知识,然后讲解了使用 Visual C++开发数据库的优势和这些开发技术的种类,最后通过实例展示了 ADO 的数据库控件的使用方法。

第 2 章讲解了 SQL Server 2000 数据库管理。主要讲述了 SQL Server 2000 关系数据库系统的有关知识,包括数据库服务器配置、数据库对象管理、数据库维护等内容。在后面章节中,将通过使用 Visual C++进行的实例开发来帮助读者掌握和巩固 SQL Server 的使用方法和技巧。

第 3 章讲解了数据库基础及 SQL 语言。从回顾数据库的发展历程开始,简单介绍了数据库的分类和操作方式,并对数据库操作语言 SQL 进行了详细介绍。

第 4~11 章按照软件工程的应用步骤阐述了数据库系统的开发过程和技巧。各个数据库应用案例如下:

- 客户资源管理系统
- 企业备忘录管理系统
- 图书馆管理系统
- 设备管理系统
- 员工培训管理系统

- 公司贸易管理系统
- 餐饮管理系统
- 宾馆管理系统

在这些系统的讲述过程中,首先阐述了系统功能概述、功能模块的划分和系统流程的分析,然后讲述了数据库需求分析、数据库概念结构设计和数据库逻辑结构设计,接下来介绍了连接到数据库的方法,最后讲解了系统中界面的设计和代码的实现方法。其中前5个案例详细介绍了利用MFC ODBC开发数据库管理系统的方法,使读者可以学会利用CDatabase数据库类和CRecordset记录集类操作数据库,以及CDBException异常类处理数据库异常的方法。后3个案例详细介绍了利用ADO开发数据库管理系统的方法,使读者可以利用ADO的数据库连接对象(Connection Object)、命令对象(Command Object)和记录集对象(RecordSet Object)来操作数据库,以及利用记录集对象实现数据库记录的添加、修改和删除等操作。

第12章介绍了数据库应用程序的优化与发布。本章主要从优化和制作安装程序两个角度介绍了软件从开发环境到生产环境所需要进行的实用化工作。

本书光盘中包含所有实例的源代码,为了运行这些代码,读者需要配置一个运行环境,安装Visual C++ 6.0 sp6 中文企业版和SQL Server 2000 简体中文版,详细的配置请参考本书具体章节的介绍。

本书由李涛、刘凯奎、王永皎编著,参与本书制作的还有郝荣福、李大宇、付鹏程、高翔、朱丽云、崔凌、张巧玲、钟华、高海峰、王建胜、张浩和刘湘等同志。由于本书涉及的范围比较广泛,作者的经验有限,书中不足之处在所难免,敬请广大读者、专家提出宝贵意见。

编 者

目 录

第 1 章 Visual C++及数据库访问基础	1
1.1 Visual C++基础	1
1.1.1 面向对象程序设计	1
1.1.2 开发环境	6
1.1.3 MFC 编程	15
1.1.4 消息映射机制	18
1.2 访问 SQL Server 数据库	20
1.2.1 MFC ODBC 数据库开发技术	21
1.2.2 ADO 数据库开发技术	28
1.3 数据库访问控件	32
1.3.1 创建一个对话框程序	33
1.3.2 添加 ADO 数据控件	34
1.3.3 添加 DataGrid 控件	36
1.3.4 添加 DataCombo 控件	37
1.3.5 添加 DataList 控件	38
第 2 章 SQL Server 2000 数据库管理	41
2.1 SQL Server 2000 概述	41
2.1.1 SQL Server 2000 简介	41
2.1.2 SQL Server 2000 功能与特点	41
2.2 SQL Server 服务器配置	42
2.2.1 服务器注册	42
2.2.2 服务器启动、暂停和停止	44

2.2.3 服务器连接与断开	46
2.3 数据库操作	46
2.3.1 数据库管理	46
2.3.2 数据表操作	53
2.3.3 视图操作	63
2.3.4 触发器管理	66
2.3.5 存储过程管理	69
2.3.6 索引管理	71
2.3.7 数据库的维护	74
2.4 系统开发工具	76
2.4.1 配置 ODBC 数据源	76
2.4.2 绘制 E-R 图	78
2.5 Transact-SQL 语言	81
第 3 章 数据库理论与 SQL 语言	83
3.1 数据库理论基础	83
3.1.1 数据库的产生和发展	83
3.1.2 数据库的信息表示和建模	86
3.1.3 关系数据库基础	88
3.2 SQL 语言简介	91
3.2.1 SQL 与过程化程序设计语言的 区别	92
3.2.2 SQL 语法分类	92

3.2.3 数据库操作.....	93	5.3 工程框架设计.....	167
3.2.4 表操作.....	93	5.3.1 创建工程并设计主界面.....	167
3.2.5 视图操作.....	95	5.3.2 连接数据库.....	171
3.2.6 数据查询.....	100	5.3.3 显示数据到界面上.....	172
3.2.7 数据修改.....	106	5.4 工程模块设计.....	178
第4章 客户资源管理系统.....	111	5.4.1 登录模块设计.....	178
4.1 系统设计.....	111	5.4.2 成员信息模块设计.....	178
4.1.1 系统功能描述.....	111	5.4.3 类型信息模块设计.....	182
4.1.2 功能模块划分.....	112	5.4.4 备忘录管理模块设计.....	184
4.1.3 系统流程分析.....	112	5.4.5 查询模块设计.....	191
4.2 数据库设计.....	113	5.5 实例演示.....	197
4.2.1 数据库需求设计.....	113	第6章 图书馆管理系统.....	201
4.2.2 数据库逻辑结构设计.....	113	6.1 系统设计.....	201
4.2.3 创建数据库.....	116	6.1.1 系统功能描述.....	201
4.2.4 创建表.....	117	6.1.2 功能模块划分.....	202
4.2.5 创建存储过程.....	122	6.1.3 系统流程分析.....	202
4.3 工程框架设计.....	123	6.2 数据库设计.....	203
4.3.1 创建工程并设计主界面.....	123	6.2.1 数据库需求设计.....	203
4.3.2 连接数据库.....	128	6.2.2 数据库逻辑结构设计.....	203
4.3.3 显示数据到主界面上.....	129	6.2.3 创建表的脚本文件.....	205
4.4 工程模块设计.....	136	6.3 工程框架设计.....	208
4.4.1 登录模块设计.....	136	6.3.1 创建工程并设计主界面.....	208
4.4.2 配置信息模块设计.....	137	6.3.2 连接数据库.....	210
4.4.3 客户信息模块设计.....	140	6.4 工程模块设计.....	212
4.4.4 客户查询模块设计.....	147	6.4.1 登录模块设计.....	212
4.5 实例演示.....	157	6.4.2 图书管理模块设计.....	213
第5章 企业备忘录管理系统.....	161	6.4.3 借阅证管理模块设计.....	232
5.1 系统设计.....	161	6.4.4 系统及其他模块设计.....	237
5.1.1 系统功能描述.....	161	6.5 实例演示.....	237
5.1.2 功能模块划分.....	162	第7章 设备管理系统.....	239
5.1.3 系统流程分析.....	162	7.1 系统设计.....	239
5.2 数据库设计.....	163	7.1.1 系统功能描述.....	239
5.2.1 数据库需求设计.....	163	7.1.2 功能模块划分.....	240
5.2.2 数据库逻辑结构设计.....	163	7.1.3 系统流程分析.....	241
5.2.3 创建表的脚本文件.....	165	7.2 数据库设计与实现.....	241

7.2.1 数据库需求设计	241	第 9 章 公司贸易管理系统	307
7.2.2 数据库逻辑结构设计	242	9.1 系统设计	307
7.2.3 创建表的脚本文件	243	9.1.1 系统功能描述	307
7.2.4 创建存储过程	246	9.1.2 功能模块划分	308
7.3 工程框架设计	246	9.1.3 系统流程分析	309
7.3.1 创建工程并设计主界面	246	9.2 数据库设计	309
7.3.2 连接数据库	250	9.2.1 数据库需求设计	309
7.3.3 显示数据到界面上	252	9.2.2 数据库逻辑结构设计	310
7.4 工程模块设计	255	9.2.3 创建表的脚本文件	313
7.4.1 用户资料管理	256	9.3 工程框架设计	315
7.4.2 设备库存管理	258	9.3.1 创建工程并设计主界面	315
7.4.3 设备借出归还管理	261	9.3.2 连接数据库	317
7.4.4 统计信息管理	267	9.4 工程模块设计	319
7.5 实例演示	271	9.4.1 供应商信息管理	319
第 8 章 员工培训管理系统	275	9.4.2 产品信息管理	327
8.1 系统设计	275	9.4.3 进货信息管理	331
8.1.1 系统功能描述	275	9.5 实例演示	340
8.1.2 功能模块划分	276	第 10 章 餐饮管理系统	343
8.1.3 系统流程分析	276	10.1 系统设计	343
8.2 数据库设计	277	10.1.1 系统功能描述	343
8.2.1 数据库需求设计	277	10.1.2 功能模块划分	344
8.2.2 数据库逻辑结构设计	278	10.1.3 系统流程分析	345
8.2.3 创建表的脚本文件	282	10.2 数据库设计	345
8.3 工程框架设计	286	10.2.1 数据库需求设计	346
8.3.1 创建工程并设计主界面	286	10.2.2 数据库逻辑结构设计	346
8.3.2 连接数据库	287	10.2.3 创建表的脚本文件	350
8.4 工程模块设计	288	10.3 工程框架设计	353
8.4.1 学员信息管理	289	10.3.1 创建工程并设计主界面	353
8.4.2 培训课程管理	291	10.3.2 连接数据库	357
8.4.3 岗位部门管理	296	10.4 工程模块设计	358
8.4.4 培训计划管理	299	10.4.1 登录模块设计	359
8.4.5 培训教材管理	299	10.4.2 前台销售管理模块设计	361
8.4.6 培训资源管理	300	10.4.3 统计模块设计	367
8.4.7 培训教员管理	301	10.4.4 基础资料模块设计	369
8.4.8 用户管理	302	10.5 实例演示	375
8.5 实例演示	303		

第 11 章 宾馆管理系统.....	379	11.5 实例演示.....	400
11.1 系统设计.....	379	第 12 章 数据库应用程序的优化与发布.....	403
11.1.1 系统功能描述.....	379	12.1 应用程序的优化.....	403
11.1.2 功能模块划分.....	380	12.1.1 应用程序的界面优化.....	403
11.1.3 系统流程分析.....	380	12.1.2 数据库设计优化.....	404
11.2 数据库设计.....	381	12.1.3 操作系统相关优化.....	406
11.2.1 数据库需求设计.....	381	12.2 数据库应用程序的安全.....	407
11.2.2 数据库逻辑结构设计.....	381	12.2.1 数据库的安全.....	407
11.2.3 创建表的脚本文件.....	383	12.2.2 程序权限控制.....	409
11.3 工程框架设计.....	385	12.2.3 应用程序日志管理.....	409
11.3.1 创建工程并设计主界面.....	385	12.3 帮助文件的创建.....	410
11.3.2 连接数据库.....	386	12.3.1 制作 CHM 帮助文件.....	410
11.4 工程模块设计.....	387	12.3.2 应用程序同帮助文件的关联.....	418
11.4.1 登录模块设计.....	387	12.4 制作安装程序.....	418
11.4.2 系统管理模块设计.....	387	12.4.1 安装程序的制作.....	419
11.4.3 房间信息管理模块设计.....	388	12.4.2 应用程序安装过程.....	423
11.4.4 订房信息管理模块设计.....	397		
11.4.5 结算信息管理模块设计.....	400		

Visual C++及数据库访问基础

Visual C++是 Microsoft Visual Studio 开发组件中最为强大的编程工具。一方面，它是当今最为流行的系统级开发语言，另一方面，它能够和 Microsoft 的操作系统无缝结合，开发出高性能的 Windows 应用程序。

在数据库领域，Visual C++的表现也异常抢眼，它提供了多种数据库开发技术，支持几乎所有的数据库系统，如 SQL Server, DB2, Sybase, FoxPro, Access 等；而且还提供了良好的开发环境和丰富的文档支持。

总之，利用 Visual C++可以开发出功能强大、性能优良和界面友好的数据库应用程序，这也正是许多公司长期以来采用 Visual C++开发数据库系统的重要原因。

1.1 Visual C++基础

Visual C++不仅仅是一个编译器，更是一个全面的应用程序开发环境，从而可以充分利用具有面向对象特性的 C++来开发出专业级的 Windows 应用程序。为了能充分利用这些特性，首先必须理解 C++程序设计语言和 Microsoft 基本类库(MFC)的层次结构。MFC 包容了 Windows API 中的用户界面部分，并适用于目前几乎所有版本的 Windows。

1.1.1 面向对象程序设计

面向对象技术是目前流行的系统设计开发技术，包括面向对象分析(OOA, Object Oriented Analysis)和面向对象程序设计(OOD, Object Oriented Design)，它的提出主要是为了解决传统程序设计方法，即结构化程序设计方法所不能解决的代码重用问题。

从工程角度来说，面向对象技术是一种围绕真实世界的概念来组织模型的程序设计方法，并采用对象来描述问题空间的实体。众所周知，客观世界的问题都是由客观世界中的实体及其

相互关系构成的,在面向对象的设计中,可以将客观世界中的实体抽象为问题空间中的对象(Object)。

其次,在解决实际问题时,由于研究的问题不同,面向的对象也就不同,一个人、一张桌子、一间房子都可以看作是一个对象。下面以桌子为例,对这一对象进行考察。一张桌子应该具有这样一些属性(attributes),例如价格、尺寸、重量、位置和颜色等,这些属性的值就代表了这张桌子的状态,如价格为300元,颜色为棕色等。对这张桌子可以用许多方式来操作,可以买、卖,进行物理变更(例如把这把桌子漆成红色),或从一个地方挪到另一个地方。每一种操作(operation)或方法(method)会改变对象的一个或更多个属性的值。要完整、准确地描述一张桌子,就必须定义这些属性及属性之上的操作。桌子状态的改变只能通过这些操作来进行。换个角度说,桌子这一对象封装了数据(定义桌子的属性值)和操作(可用来改变桌子属性的动作)。

从这个例子中可以得到对象和类两个直观的定义。

- 对象: 就是一个封装了数据和操作的实体。对象的结构特征由属性表示,属性数据描述了对应的状态,操作可改变属性数据从而改变对象的状态。
- 类: 就是具有相同操作功能和相同属性(数据格式)的对象的抽象集合。类是对象集合的一种抽象,它规定了这些对象的公共属性和方法;对象则是类的一个实例。香蕉是一个类,而某一个具体的香蕉则是一个对象。

类作为面向对象程序设计的核心概念,在工程中有很多的应用特征,体现在如下几个方面。

- 继承性(inheritance)

软件的可重用性、程序成分的可重用性都是通过继承类中的属性和操作来实现的。因为重用就意味着利用已有的定义、设计和实现,缺少这种继承的手段是无法做到的。在定义一个新的对象时,利用继承性只需指明它具有哪些类定义以外的新的特性,即说明其个性,而不必定义新对象的全部特性。这就大大减少重复定义,充分利用了前人的劳动成果,同时也使定义的系统结构更加清晰、易于理解和维护。

类的对象之间有继承性,类与类之间也可以有继承性从而构成层次结构,相对上层的是超类(superclass)或称基类,相对下层的是子类(subclass)或称派生类,子类在继承超类的属性和操作的同时可以拥有自己特有的属性和操作。

- 封装性(encapsulation)

在面向对象的领域内,把对象定义为封装了数据和操作的实体。这个定义将对象的各种独立的外部特征与内部细节分开,亦即对象的具体数据结构和各种操作实现的细节对于对象外的一切是隐藏的。例如要知道一张桌子的重量时,只需向这个对象发出一条获得重量的请求,然后可以得到其重量,这便是对象外部所需做的一切,而完全不必要了解给桌子称重的具体细节。对象将其实现细节隐藏在其内部,因此无论是对象功能的完善扩充,还是对象实现的修改,影响仅限于该对象内部,而不会对外界产生影响,这就保证了面向对象软件的可构造性和易维护性。

- 多态性(polymorphism)

多态性是指允许不同类的对象对同一消息作出响应。例如同样的加法,把两个时间加在一起和把两个整数加在一起肯定完全不同。又如,同样的选择编辑、粘贴操作,在字处理程序和绘图程序中有不同的效果。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势,很好地解决了应用程序函数同名问题。多态性在类的虚函数中有很好的体现。

在 C++ 中,对象的声明与基本数据类型的变量声明一样,要声明一个类的对象,只需用类名声明即可。例如 `MyClass myObject` 声明了一个类对象 `myObject`,该对象是 `MyClass` 类的一个实例,`MyClass` 是类的名字。

定义或声明一个类的语法格式如下。

```
class class_name
{
    private:
        //私有的成员变量和成员函数
    protected:
        //受保护的成员变量和成员函数
    public:
        //公有的成员变量和成员函数
};
```

注意:类声明的结束处要加分号,成员的声明不能使用 `extern`, `auto` 和 `register` 关键字修饰。

如果定义时不写关键字 `private`, `protected` 或 `public`,则系统默认为 `private`,这 3 个关键字出现的顺序和次数可以是任意的。它们是访问权限修饰符,访问处于修饰部分之内的成员时会受到不同的限制。

- 私有部分(private)

从关键字 `private` 开始一直到其他访问权限修饰符或“`}`”符号处,这部分的成员变量和成员函数只有类的成员函数可以访问,类的对象、类的派生类都无权直接访问私有成员。

- 受保护的部分(protected)

从关键字 `protected` 开始一直到其他访问权限修饰符或“`}`”符号处,这部分的成员只有类的成员函数或派生类的成员函数可以访问,类的对象无权直接访问。

- 公有部分(public)

从关键字 `public` 开始一直到其他访问权限修饰符或“`}`”符号处,这部分的成员是开放的,类的成员函数、类的对象,派生类的对象和成员函数都可以直接访问。

例如:

```
class Point{
    int x;
    int y;
```

```
public:
    int m_iPos;
    int GetX();
    int GetY();
};
```

上面的 C++ 代码声明了一个 Point 类，它有两个私有成员变量 x 和 y，一个公有成员变量 m_iPos 和两个公有成员函数 GetX 和 GetY，这两个成员函数可以访问成员变量 x 和 y，但是用 Point mypoint; 语句声明的对象 mypoint 不能直接访问成员变量 x 和 y，而 m_iPos 则是成员函数和对象都可以直接访问。成员函数对成员变量的访问是在函数体中直接使用变量名，对象对成员变量的访问使用 “.” 操作符，例如：

```
int z = mypoint.m_iPos;    //将 m_iPos 的值赋给 z
```

一般情况下，在一个 .h 文件中进行类声明，若声明中对成员函数只进行了声明而没有实现，那么在一个 .cpp 文件中定义这些成员函数的功能实现。定义成员函数的一般形式如下。

```
返回类型 类名::成员函数名(形式参数表)
{函数体}
```

其中，“::”是作用域运算符，“类名::”表示其后的成员函数属于该类，在成员函数的函数体中就可以直接使用类的成员变量和成员函数。

以上介绍了类的声明和成员函数的实现，类还具有继承性与派生性的特点。众所周知，在现实生活中也存在继承和派生关系的例子。例如，自行车、汽车、飞机和轮船都是交通工具，其中自行车和飞机又属于陆地交通工具。这种从属关系可以用图 1-1 表示。

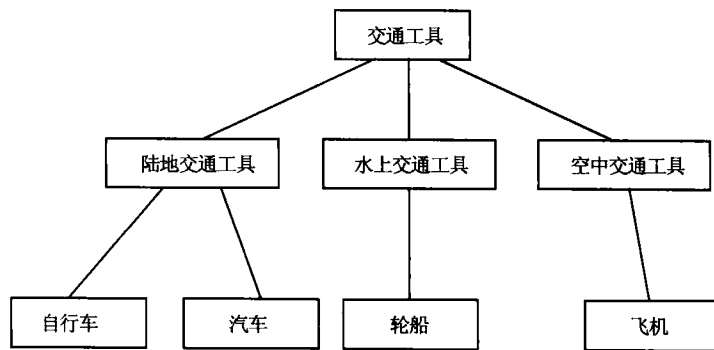


图 1-1 交通工具类层次

图 1-1 中位于上方的是基础类别，位于下方的是派生类别。派生出来的类别和基础类别之间的关系是“是一种”(Is A Kind Of)的关系，派生类别具有基类的所有特征，还可以有其区别于基类的更多特征，其中可能包括新的成员变量和成员函数。

派生类定义的语法格式如下。

```
class <派生类> : <访问权限修饰符> <基类 1> [, [访问权限修饰符] <基类 2>, ... ]
```

其中,“派生类”为要派生的类的名称,访问权限修饰符可以是 `public`, `protected` 或者 `private`。“基类”为派生类所继承的基类名称,一个派生类可以继承多个基类,一个派生类还可以再派生出另一个类。

例如,可以用 C++ 语言来简单描述上述“交通工具”类层次图中的部分“类”,代码如下。

```
class CVehicle // 交通工具类
{
private:
    int m_nWeight;
public:
    void Start(); // 工具启动
    void Move(); // 工具移动
    void Stop(); // 工具停止
    SetWeight(int nWeight) { m_nWeight = nWeight; } //设置重量
};
class CTerrVehicle : public CVehicle // 陆地交通工具类从交通工具类继承而来
{
protected:
    int m_nWheel; // 陆地交通工具都有轮子
public:
    void SetWheel(int nWheel) { m_nWheel = nWheel; } //设定车轮数量
};
class CWaterVehicle : public CVehicle // 水上交通工具类从交通工具类继承而来
{
public:
    GetTonnage() {...}; // 得到排水量
};

class CAirVehicle : public CVehicle // 空中交通工具类从交通工具类继承而来
{
public:
    GetFlyingTime() {...}; // 得到飞行时间
};

class CBike : public CTerrVehicle // 自行车是一种陆地交通工具
{
public:
    void Brake() {...}; // 刹车
};
```

```
class CCar : public CTerrVehicle // 汽车是一种陆地交通工具
{
public:
    void FillGas()          {...}; // 汽车需要加油
    void Brake()           {...}; // 刹车
};
```

对于上面的例子需要说明的是：派生类通过继承隐含地具有了基类所定义的成员变量和成员方法，例如 CCar 类就继承了 CTerrVehicle 类的 m_nWheel 和 SetWheel()，以及 CCVehicle 类的 m_nWeight 和 Start()，Stop()，Move()和 SetWeight()。所以对于类 CCar，其定义实际上相当于如下代码。

```
class CCar
{
private:
    int m_nWeight;
protected:
    int m_nWheel;
public:
    void FillGas()          {...}; // 汽车需要加油
    void Brake()           {...}; // 刹车
    void SetWheel(int nWheel) { m_nWheel = nWheel; }
    void Start();
    void Move();
    void Stop();
    SetWeight(int nWeight) { m_nWeight = nWeight; }
};
```

1.1.2 开发环境

Visual C++作为一种程序设计语言，同时也是一个集成开发工具。它不但提供了软件代码自动生成的功能，同时也可以对资源文件进行可视化的编辑。

1. Visual C++的文件类型

在使用 Visual C++开发应用程序的过程中，系统自动生成了大量的各种类型的文件，不同类型的文件在 Visual C++工程中起着不一样的功效。

扩展名为 dsw 的文件在 Visual C++中是级别最高的，称为 Workspace 文件。在 Visual C++中，应用程序是以 Project 的形式存在的，Project 文件以.dsp 为扩展名，在 Workspace 文件中可以包含多个 Project，由 Workspace 文件对它们进行统一的协调和管理。



与 dsw 类型的 Workspace 文件相配合的一个重要的文件类型是以 opt 为扩展名的文件, 这个文件中所包含的, 是在 Workspace 文件中要用到的, 本地计算机的有关配置信息, 所以这个文件不能在不同的计算机上共享, 当打开一个 Workspace 文件时, 如果系统找不到需要的 opt 类型文件, 就会自动创建一个与之配合的包含本地计算机信息的 opt 文件。

Project 文件的扩展名是 dsp, 该文件中存放的是一个特定的工程, 也就是特定的应用程序的有关信息, 每个工程都对应有一个 dsp 类型的文件。

以 clw 为扩展名的文件用来存放应用程序中用到的类和资源的信息, 这些信息是 Visual C++ 中的 ClassWizard 工具管理和使用类的信息来源。

对应每个应用程序有一个 readme.txt 文件, 该文件中列出了应用程序中用到的所有文件信息, 打开并查看其中的内容就可以对应用程序的文件结构有一个基本的认识。

在应用程序中大量应用的是以 h 和 cpp 为扩展名的文件, 以 h 为扩展名的文件称为头文件。以 cpp 为扩展名的文件称为实现文件。一般来说以 h 为扩展名的文件与以 cpp 为扩展名的文件是一一对应配合使用的, 在以 h 为扩展名的文件中主要包含的是类的定义, 而在以 cpp 为扩展名的文件中主要包含的是类成员函数的实现代码。

Visual C++ 中以 rc 为扩展名的文件称为资源文件, 其中包含了应用程序中用到的所有的 Windows 资源。注意, rc 文件可以直接在 Visual C++ 集成环境中以可视化的方法进行编辑和修改。

以 rc2 为扩展名的文件也是资源文件, 但该文件中的资源不能在 Visual C++ 的集成环境下直接进行编辑和修改, 而是根据需要手工地编辑这个文件。

对于以 ico, bmp 等为扩展名的文件是具体的资源, 产生这种资源的途径很多。使用 rc 资源文件的目的是为了对程序中用到的大量的资源进行统一的管理。

2. Visual C++ 工具介绍

Visual C++ 提供了各种向导和工具来简化 Windows 程序的开发过程, 在一定程度上实现了软件的自动生成和可视化编程。下面介绍 Visual C++ 集成环境中几个最主要的开发和辅助工具。

首先要介绍的是 Appwizard 工具, 其作用是帮助程序员一步步地生成一个新的应用程序, 并且自动生成应用程序所需的基本代码, 具体操作步骤如下。

(1) 选择“文件”|“新建”命令, 打开“新建”对话框, 并选择所要创建的文件类型。如下图所示。这里的文件分成了“文件”、“工程”、“工作区”、“其他文档”4 大类型, 每一类型下面又包含许多具体的文件类型, 选中“工程”选项卡, 在列表框中列出的是各种不同的应用程序类型, 例如 dll 类型的动态链接库, exe 类型的可执行程序等, 这里选中 MFC Appwizard(exe)选项, 表示要创建的是一个使用 MFC 基本类库进行编程的可执行程序。

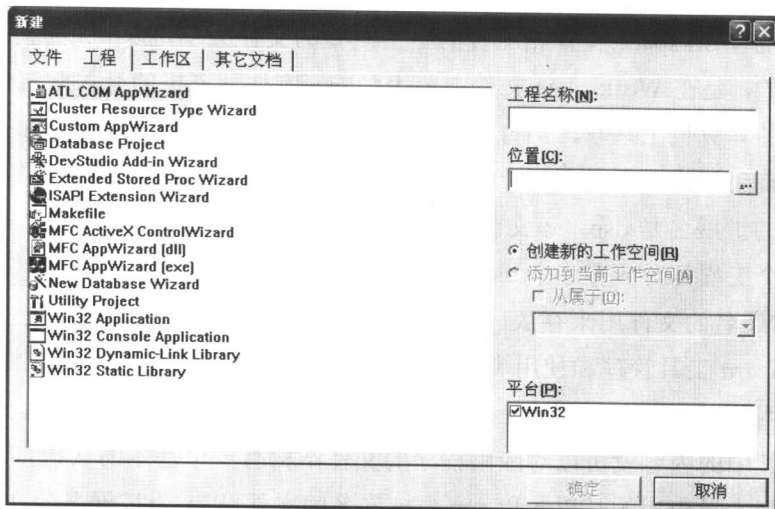


图 1-2 新建 MFC 应用程序

(2) 在“工程名称”文本框中输入 `test`，在“位置”文本框中输入程序定义文件存放的目录(对话框右下角的平台一栏中的 `Win32` 项表示要创建的程序是建立在 32 位的 Windows 平台上)，如图 1-3 所示。单击“确定”按钮，就启动了使用 MFC 方式开发应用程序的“应用程序向导”功能。

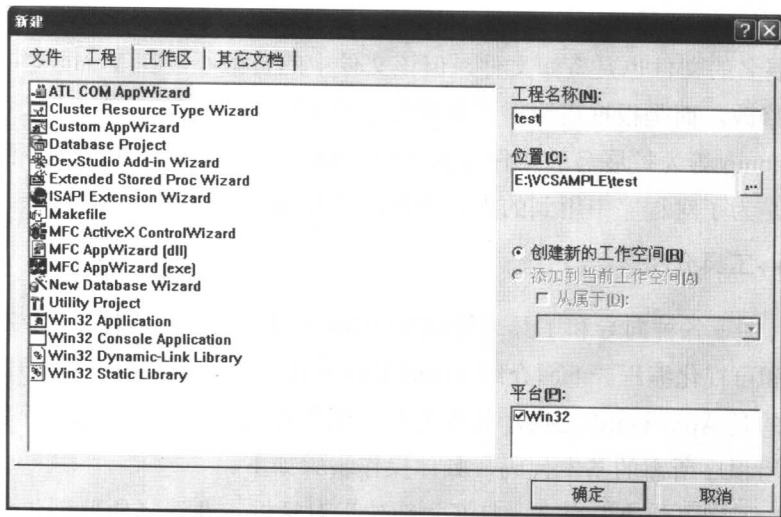


图 1-3 新建 test 应用程序

(3) 选择程序的类型为“单文档”、资源所用的语种为“中文”，如图 1-4 所示。

(4) 单击“下一步”按钮，在图 1-5 所示的步骤中可以选择程序是否需要数据库方面的支持，在这里选择“否”单选按钮。