

卓越工程师计划：软件工程专业系列丛书

阶梯式 GIS

软件工程实践系列教程

方芳 杨林 周顺平 编著

数据库篇



科学出版社

卓越工程师计划:软件工程专业系列丛书

阶梯式 GIS 软件工程实践 系列教程——数据库篇

方 芳 杨 林 周顺平 编著

科学出版社

北 京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书是“阶梯式 GIS 软件工程实践系列教程”的数据库篇,目的是通过完成一件事(实现一个小型点线区图形编辑系统)和围绕一个核心(数据库开发),使读者在了解数据库基本概念和基本用法的基础上,掌握运用 C++ 语言通过 SQL 访问和控制数据库的方法。

本书包括实习目的及要求、背景知识概述、系统实现过程和强化编程练习四个部分,通过一系列循序渐进的练习有机地贯穿了数据库基本概念和用法、动态库开发和程序封装、点线区图形结构的定义和数据库存储、C++ 语言和编程工具等方面的内容。

其中系统实现过程这部分按照系统架构设计分别开发“图形数据管理动态库”和“图形编辑系统”,前者完成对关系数据库的封装并提供点线区数据的增删改查接口,后者调用动态库提供的接口完成点线区图形的编辑、查询和显示。整个实现过程已做了细致的划分,各次练习环环相扣,读者依次做循序渐进的练习,在数据库和编程知识运用上越来越深,在功能实现上越来越难,逐步达到强化数据库开发训练的目的。本书各项功能实现过程中,数据库开发不仅涉及常识层面,而且还包括了存储过程和触发器等高级开发,所有这些都被自然地融入到一个个具体的练习中。

“强化编程练习”部分给出了功能要求并说明实现相应功能的思考过程,为有意在功能和数据库开发方面做深度练习的读者提供指引。

本书可作为大专院校师生、数据库或 GIS 基础软件开发人员参考用书,特别是作为大二到大三期间的综合实践用书。

图书在版编目(CIP)数据

阶梯式 GIS 软件工程实践系列教程. 数据库篇/方芳,杨林,周顺平编著. —北京:科学出版社,2015.9

(卓越工程师计划:软件工程专业系列丛书)

ISBN 978-7-03-045722-6

I. ①阶… II. ①方… ②杨… ③周… III. ①地理信息系统—教材
IV. ①P208

中国版本图书馆 CIP 数据核字(2015)第 222550 号

责任编辑:张颖兵 闫 陶/责任校对:肖 婷

责任印制:高 嵘/封面设计:陈明亮

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本:787×1092 1/16

2015 年 9 月第 一 版 印张:11 3/4

2015 年 9 月第一次印刷 字数:259 000

定价:27.00 元

(如有印装质量问题,我社负责调换)

卓越工程师计划:软件工程专业系列丛书

编委会

丛书主编 谢 忠 周顺平 罗忠文

编 委 (按姓氏笔画排序)

万 波 方 芳 叶亚琴 左泽均

向秀桥 孙 明 张剑波 李圣文

杨 林 杨之江 杨林权 周 晔

周顺平 尚建嘎 罗忠文 胡茂胜

袁国斌 龚君芳 龚国清

前言

地理信息系统(Geographic Information System, GIS)是在地理学、测绘科学和计算机科学等学科基础上发展起来的交叉性新兴学科,是研究地理信息采集、存储、管理、分析和应用的技术和工具。随着 GIS 技术的进步, GIS 在测绘、地理、地质、环保、国土、城市、农业、军事等领域得到了越来越广泛的应用,并渐渐发展成为一个产业。

GIS 本质上是利用计算机技术特别是软件技术,对地理信息进行加工并服务于各行各业的一种信息技术。伴随着网络技术和移动技术的推广普及,地理信息的应用范围不断扩大,已从初期的科研和政府部门,扩大到百姓的日常生活和各种商业领域,并从室内延伸到室外;应用深度也不断细化,从地图制作和应用发展到地理信息服务,应用渗透到定位导航、生产调度、国情监测、灾害监测和其他各行各业,并涌现出了寻人、定向广告等各种与地理信息相关的个性化应用。

随着应用领域不断扩大,各种新型的 GIS 应用层出不穷,推动了 GIS 产业的快速发展,使 GIS 成为一个非常具有活力和发展前景的新兴领域,同时也带来了 GIS 应用人才和软件开发人才的强烈需求。

为顺应学科和产业发展的需要,自 20 世纪 90 年代以来,国内越来越多的高校开办了 GIS 本科专业。截至 2010 年,开办 GIS 专业的高等院校已超过 200 所。

各校的 GIS 本科专业大多建立在地学相关专业或计算机专业基础上,培养目标也大致分为两类:面向生产业务和面向软件开发。其中面向生产业务的培养目标,主要是培养能够运用 GIS 知识和工具开展数据处理、制图和行业应用的人才;而面向软件开发的培养目标,主要是培养具有 GIS 知识,能够开展 GIS 相关软件设计和实现的软件工程师。

各高校 GIS 专业在培养 GIS 软件开发人才的过程中,利用各自的领域优势,制定了各具特色的人才培养方案。为国土、地矿、测绘、规划、交通、物流、农业、电信等领域输送了大批具有相应背景知识的软件开发人才。

中国地质大学(武汉)信息工程学院是我国较早建立 GIS 专业的教学单位之一。学院结合自主研究和开发 MapGIS 国产地理信息系统平台软件的经验,以及学校地学背景,在专业定位上以培养面向地学信息化的软件人才为主要培养目标,十多年来为各领域输送了大批 GIS 软件开发人才,学生受到用人单位的普遍欢迎。

GIS 软件开发与其他行业软件开发一样,是一种对动手能力要求极高的工作。众多高校为了强化 GIS 软件开发人才的动手能力,安排了 GIS 软件操作、基础开发、二次开发等不同层次的实践教学课程,实践类型更是丰富多彩,包括课内实习、课程设计、综合实习、产学研、第二课堂、项目实践和企业实习等。

虽然普遍重视实践教学,形式和层次也进行了较为合理的搭配,但 GIS 软件开发人才培养在实践内容方面依然普遍存在以下脱节现象。

(1) 实践内容相互脱节,缺乏主题将各门主要课程和主要实践进行有机联系:①软件开发类课程实习与 GIS 课程脱节,如数据结构实习与 GIS 各种常用结构无关,高级编程语言实习与 GIS 软件开发知识无关;②课程实习与课程实习之间脱节,如高级语言实习、数据结构实习、数据库实习、网络编程实习等相互脱节;③基础实习与综合实习脱节,如数据结构、数据库等课程的课内基础练习与综合实习在练习内容、练习深度上没有很好地关联。

(2) 实践内容与实际需求脱节。各实践课程内容过于传统,与 GIS 基础知识关联度小,实现功能简单与实际应用系统差距较大,在一定程度上降低了学生的学习兴趣。

(3) 各实践课程间缺乏系统性和连贯性,不利于强化和巩固知识点,实践教学质量难于保证。

(4) 综合实习没有标准化,对于要求高、综合性强的题目,对于很多学生在问题和解决方案之间存在巨大的鸿沟,有限的实习时间内难以圆满跨越。因为没有标准化,在综合实习环节也难以利用研究生等辅助教学资源。

上述问题不仅导致学生在软件开发方面的能力参差不齐,而且导致学生的软件开发能力与社会需求严重脱节。因此需要一种将各门课程的主要知识点和技能与 GIS 软件开发有机结合起来的实践,通过一系列由易到难的实践,学生在实现有强烈应用背景的功能中自然而然地运用了各种知识点和技能,从而提高学生 GIS 软件开发的能力。这就是我们希望编写一套阶梯式 GIS 软件工程实践系列教程的初衷。

该阶梯式 GIS 软件工程实践系列教程期望达到下列目标。

(1) 阶梯式。实践难度逐级提高,后面的实践基于前面的实践。

(2) 系统化。考虑到不同年级之间、不同课程之间实践内容的更好衔接;课程实践设置与 GIS 系统挂钩,既关注知识点,也关注综合运用;实践系统化与整体化。

(3) 标准化。不同级别的实践内容标准化,便于教学实施和质量控制;便于授课教师、辅导老师、助研培训与备课;便于学生准备与开展实践活动。

(4) 导向性与挑战性。阶梯式实践教学体系更具导向性,同时也能够满足创新能力强的学生的实践需求。

传统实践教学中,课程内实习是知识点的辅助练习,个性化项目实践和第二课堂则是培养创新能力的环节,该系列教程基于上述目标,旨在有效衔接和补充传统教学环节。

经过万波、叶亚琴、方芳、杨林、左泽均、胡茂盛等老师的努力,终于形成了阶梯式 GIS 软件工程实践系列教程的基础篇、数据库篇和网络篇。基础篇面向大一到大二的学生,重点训练学生的 GIS 软件开发基础技能,包括基础知识、编程语言、编程工具三位一体的训练;数据库篇面向大二到大三阶段的学生,重点训练 GIS 软件开发专业技能,工程、系统和专业方向三位一体的训练;网络篇面向大三到大四阶段的学生,重点是 GIS 应用软件开发训练,特别是基于网络和地图服务的训练。

对于期望从事 GIS 基础软件开发的学生,从基础篇开始练习是不错的选择,然后选择数据库篇以加强数据库开发技能,最后再选择网络篇。

本书是数据库篇,对于喜好数据库开发的学生,也可独立选择本书作为实践指导书。

本书通过开发一个简单的图形编辑器,使学生掌握点、线、区等基本矢量数据的组织方式和 Windows 图形编程方法;通过将点、线、区等图形数据存储到关系数据库中,使学生通过大量 SQL 脚本的编写,加深对关系数据库的理解,提高数据库开发能力;通过将点、线、区数据存储相关的功能封装到一个动态库中,使学生能够反复体会封装等概念和多层体系架构。

该系列实践教程是中国地质大学(武汉)信息工程学院十多年 GIS 和软件工程专业实践教学经验的总结,出版教程既为了便于开展教学,也为了与兄弟院校分享经验,衷心期望广大师生对该系列教程提出宝贵意见,以便充实改进。

感谢宴四方、熊军、廖婧、刘超群、冯庄等研究生协助完成初稿并测试所有的上机指南。

编者

2015 年 4 月

目 录

第 1 章 实践目的及要求	1
1.1 实践目的	1
1.2 实践目标	2
1.2.1 数据库开发	2
1.2.2 C++语言	3
1.2.3 图形绘制	3
1.2.4 编程工具和框架	3
1.2.5 程序调试	5
1.2.6 编程规范化	6
1.3 实践要求	6
1.3.1 对学生的要求	6
1.3.2 对老师的要求	6
第 2 章 背景知识概述	7
2.1 几何图形及其结构	7
2.1.1 点	7
2.1.2 线	7
2.1.3 区	8
2.2 Windows 图形编程	9
2.2.1 图形绘制方法	9
2.2.2 数据坐标系与窗口坐标系	10
2.3 数据库基本概念	11
2.4 系统功能与设计说明	12
2.4.1 软件架构设计	12
2.4.2 存储层“数据库物理结构”设计	12
2.4.3 逻辑层“数据管理接口”设计	15
2.4.4 表现层“功能及菜单”设计	15
第 3 章 系统实现过程	16
练习 1:创建图形数据管理层动态库工程	16
练习 2:在数据库中手动创建所需数据表	16
练习 3:建立“连接数据库”和“断开数据库”接口	22
练习 4:实现“初始化图形数据库”接口	27
练习 5:实现访问图形数据总表的接口	29
练习 6:实现“添加点”接口	44
练习 7:创建图形编辑系统应用工程	47

练习 8:完成“连接数据库”和“断开数据库”功能	54
练习 9:完成“初始化图形数据库”功能	58
练习 10:完成“造点”功能	59
练习 11:实现“查找点”接口	62
练习 12:实现“删除点”接口	64
练习 13:实现“删除点”功能	65
练习 14:实现“修改点”接口	69
练习 15:实现“移动点”功能	71
练习 16:实现“添加线”接口	74
练习 17:实现“添加线”功能	79
练习 18:实现“查找线”接口	84
练习 19:实现“删除线”接口	88
练习 20:实现“删除线”功能	89
练习 21:实现“修改线”接口	94
练习 22:实现“移动线”功能	97
练习 23:实现“添加区”接口	101
练习 24:实现“添加区”功能	105
练习 25:实现“查找区”接口	110
练习 26:实现“删除区”接口	113
练习 27:实现“删除区”功能	114
练习 28:实现“修改区”接口	119
练习 29:实现“移动区”功能	122
练习 30:窗口重绘	126
练习 31:实现“计算图形准确外包络矩形”功能	128
练习 32:编写“计算图形准确外包络矩形”存储过程	133
练习 33:用存储过程优化“计算图形准确外包络矩形”功能	136
练习 34:增加“更新点、线、区最大 ID 号”触发器	139
第 4 章 强化编程练习	143
练习 35:增加“更新点、线、区逻辑数”触发器	143
练习 36:增加“恢复点”接口	143
练习 37:实现“恢复点”功能	143
练习 38:实现“修改点参数”功能	144
参考文献	145
附录 1 C++编码规范	146
附录 2 优秀程序员的基本修炼	169

第 1 章 实践目的及要求

1.1 实践目的

本教程引导学生开发一个简单的图形编辑系统,并将图形数据存储到关系数据库中。通过该图形系统的实现,在编程语言、复杂数据结构、图形学、Windows 绘图方法、编程工具和调试环境、数据库编程、概念封装、接口设计等方面得到循序渐进的综合性训练。

数据库作为数据存储和管理最重要的平台和工具,在各种应用系统中起着基础性的作用。数据库已成为普及计算机知识、涉足信息化或软件开发的必修课。在网络已经普及并进入大数据时代的今天,理解数据库基本原理、掌握数据库基本开发技能是每一个应用软件开发人员必须修炼的基本功。

因此,本教程的重心是数据库开发,但所设计的数据库开发不再局限于常规字段和表格的操作,而是对矢量图形这样一种不定长数据进行存储、管理和操作,并且在实现的功能中,自然地运用了存储过程和触发器等数据库的特性。存储过程和触发器在常规的数据库实习中很少涉及,但在大型数据库应用系统开发中又非常有用。通过上述实践,学生的数据库开发能力有实质性的提升。

这就是本书命名为数据库篇的原因。

可视化是当今软件开发的又一个重要特征。无论科学计算、流程管理,还是地理信息表达、游戏涉及和地层模拟,都强调可视化。可视化的基本手段是使用计算机图形技术实现直观的再现和互动操作。

本教程通过图形编辑系统的开发,反复练习基于 Windows 的图形显示和操作方法,理解和掌握 Windows 图形编程技术。

总而言之,本教程主要实现以下三个目的:①加深对数据库的理解,提高数据库开发技能;②在关系数据库基础上封装出“点”、“线”、“区”等概念及其操作接口,理解和掌握封装概念;③提高 C++语言、数据结构、图形学和 Windows 编程的运用能力。

Windows 编程和程序调试不再是本书强调的内容,所以建议对 C++语言、数据结构、图形学、Windows 编程以及编程工具和调试环境缺乏基础的学生,先完成“基础篇”中的练习,再开展本书所列的练习。

本教程各项练习安排在完成了 C++语言、数据结构、数据库等课程学习后实施,可作为一项完整的课程设计。通过该项实习,达到巩固已有知识、学习新知识和技能、综合运用多门课程知识的目的。

设计本书的目的之一是强化关系数据库开发,因此点、线、区数据存储未使用 SQL Server 数据库的 geography 和 geometry 这两种针对地理对象和几何对象的数据类型。感兴趣的学生可以在完成本书的练习后,使用 geometry 类型字段重新设计点表(PNT_Table)、线表(LIN_Table)和区表(REG_Table),重构数据管理层动态库 DataBaseDLL

中各接口函数的代码实现,并学习和思考 geography 和 geometry 的区别和用途。
最终实现的软件系统如图 1.1 所示。

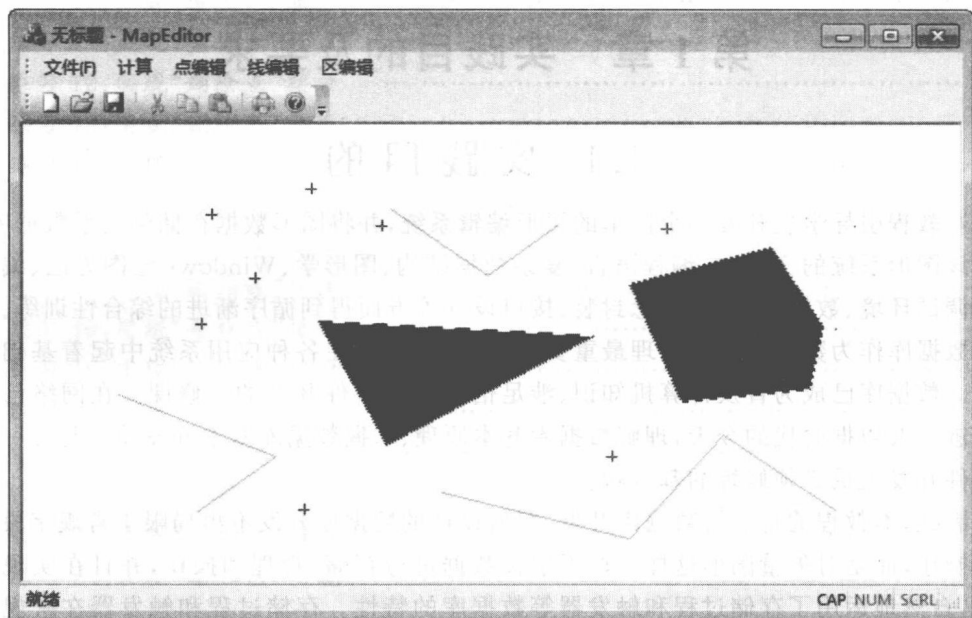


图 1.1 实习成果软件界面(参考)

1.2 实践目标

掌握数据库开发方法是本教程的重点。另外通过实习,在 C++ 语言、数据结构、图形绘制、编程工具和框架、程序调试和编程规范化方面得到进一步强化。

1.2.1 数据库开发

要求理解数据库基本概念,掌握数据库开发方法。

需要理解和掌握的数据库基本概念如下。

- (1) 数据库服务器和相关概念,如数据源、服务器类型、名称、登录用户和权限等。
- (2) 数据库和相关概念,如数据库文件、连接数据库、断开数据库等。
- (3) 表格和视图,以及相关的索引、依赖、触发器等。
- (4) 存储过程等。

需要掌握的数据库开发基本方法如下。

- (1) 服务器引擎创建。
- (2) 利用数据库工具手动创建数据库、表格、视图、触发器、存储过程。
- (3) 利用数据库工具手动编写 SQL 语句,执行并查看结果,借此验证 SQL 的正确性。
- (4) 掌握 C++ 语言中嵌入 SQL 语句的方法。
- (5) 掌握 ODBC 封装类 CDataBase 和 CRecordSet 的调用方法。

1.2.2 C++语言

该部分要求与基础篇基本相同,目的在于强化。

要求掌握 C++ 语言的核心内容,能够熟练运用各种概念和方法。

应该掌握的内容如下。

(1) C 语言的基本部分,如字符集、关键字、标识符和操作符、变量和常量、表达式、语句、过程控制。

(2) 函数的定义方法和调用方法。

(3) 数组、指针、结构的定义和使用。

(4) 类的定义和使用方法,理解和掌握成员变量和成员函数的定义和使用方法。通过微软基础类库(Microsoft Foundation Classes, MFC)的调用,充分理解类的概念,熟练掌握类的调用方法,特别是基类成员变量和成员函数,以及 this 指针的使用方法。

(5) 理解动态链接库(Dynamic Link Library, DLL)的概念及其作用,掌握创建和调用方法。

1.2.3 图形绘制

该部分要求与基础篇相同,目的在于强化。

要求理解 Windows 绘图原理,掌握 Windows 绘图方法。

本次实习是在 MFC 环境下完成的, MFC 将 Windows 的绘图方法封装成设备描述、画笔、刷子、绘图模式等 C++ 类和函数。

通过实现点、线、区等几何图形的交互式编辑、缩放、移动等功能,一方面充分理解数据坐标到窗口坐标之间的映射关系;另一方面理解 Windows 绘图原理,熟练掌握 CClientDC、CPen、CBrush 等类的使用方法,通过实现橡皮线等功能理解 DC 的绘图模式。

1.2.4 编程工具和框架

该部分要求与基础篇相同,目的在于强化。

要求掌握编程工具的基本用法,理解 Visual Studio 应用程序框架。

本书全部基于 Visual Studio 2010 进行编写,希望学生使用该版本进行练习,使用其他版本可能存在少量界面和功能的不一致。

Visual Studio 是一套集成开发环境(IDE)的开发工具,除了常规的针对代码和资源进行编辑、编译和运行所需的常规功能(如文件、工具、编辑、生成、调试等),还包括大量用于查看代码和资源静态状态的功能和视图(如解决方案资源管理器、类视图、属性管理器、资源视图),以及查看代码运行时资源状态和运行结果的功能和视图(如断点设置、逐语句跟踪、输出窗口、局部变量窗口、监视窗口、调用堆栈窗口和即时窗口等)。熟悉和掌握这些功能、工具和视图的用法,是掌握 Visual Studio 集成工具的基础,也是本书希望学生掌握的基本技能。在此基础上,学生可自学 Visual Studio 提供的“体系结构”、“测试”、“分析”等高级功能。

“解决方案资源管理器”提供项目及其文件的有组织的视图,并且提供对项目 and 文件相关命令的便捷访问。“类视图”用于显示正在开发的应用程序中定义、引用或调用的符号,阐明代码中的符号结构,并且提供对符号的便捷访问。“资源视图”用于显示工程中用

到的所有非编程部件资源,并且提供对资源的快捷编辑。

Windows 是基于消息循环机制的操作系统,Windows 所有的程序都是由消息驱动的。Windows 收集和管理各类事件(如单击菜单或按钮、鼠标移动、键盘按下等)产生的消息,并将消息发送给与消息相关的应用程序,应用程序在接收到消息后根据消息类型执行相应的操作。例如,当用户单击某菜单项时,Windows 首先捕获到该事件,产生一个 WM_COMMAND 消息并发送到用户单击的应用程序的消息队列中,应用程序逐个处理消息队列中的消息,在处理 WM_COMMAND 消息时,调用相应的消息处理函数。

Visual Studio 将 Windows 操作系统的各种应用程序接口(Application Programming Interface,API)函数封装成 C++ 类库,统称为 MFC,MFC 中封装的类如 CWnd、CButton、CFile、CDialog 等,MFC 还将消息处理机制封装成应用程序框架。在 MFC 应用程序框架基础上开发应用程序,程序员就可以大大简化“接收消息—调用消息处理函数”这一复杂过程,而将思维集中在文件处理、视窗操作、数据对象的设计与实现上来。因此理解 MFC 并掌握应用程序框架,对使用 Visual Studio 开发应用程序非常重要。

MFC 应用程序框架由“应用程序类”(CWinApp)→“主框架类”(CMainFrame)→“视图类”(CView)→“文档类”(CDocument)构成。一般一个应用程序有一个应用程序类、一个主框架类、多个视图类、多个文档类,其关系如箭头所示。

MFC 应用程序框架各组成部分及其关系如图 1.2 所示。

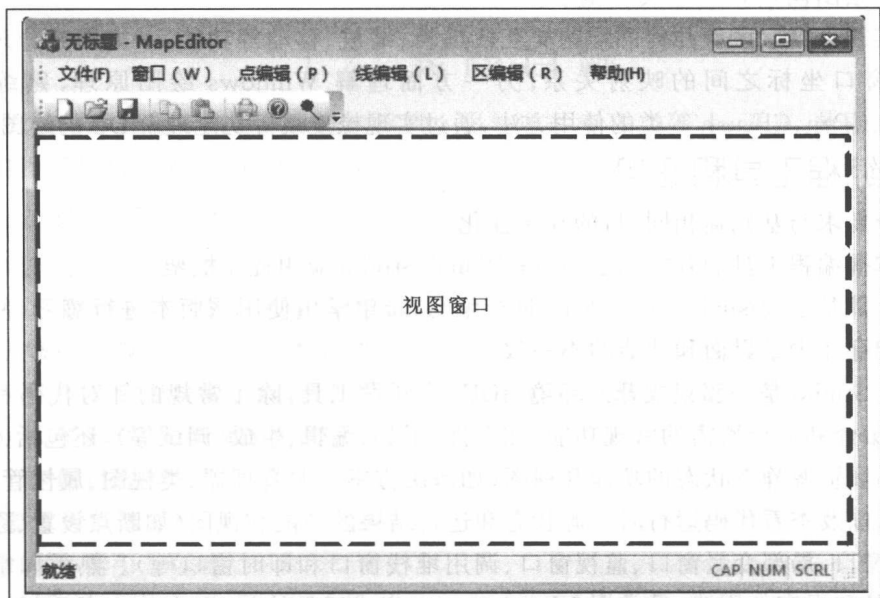


图 1.2 MFC 应用程序框架

图 1.2 中,“主框架窗口”是整个应用程序的窗口,即图 1.2 中实线框内的部分。“视图窗口”是主框架窗口的一个子窗口,即图 1.2 中虚线框内的部分。主框架窗口对应的类是主框架类 CMainFrame,视图窗口对应的类是视图类 CView。

MFC 提供了一个文档/视图结构,文档指的是文档类 CDocument,视图指的是视图

类 CView。数据的存储和加载可以由文档类完成,数据的显示和修改则由视窗类完成。对于以 MapEditor 命名的程序,文档类指的是 CMapEditorDoc,派生于 CDocument。视窗类指的是 CMapEditorView,派生于 CView。

本书为了更好地实践数据结构相关知识和文件操作,没有使用 CDocument 类,而是使用 CFile 类进行数据存储和管理。

对于以 MapEditor 命名的程序,应用程序类指 CMapEditorApp,派生于 CWinApp。在程序启动时首先会通过 CMapEditorApp 类完成一些初始化工作,包括窗口类的设计、注册以及窗口的创建、显示和更新。然后再进入消息循环中,通过消息映射机制来处理各种消息。

MFC 中的所有类均派生于 CObject 类,各种派生类如图 1.3 所示。

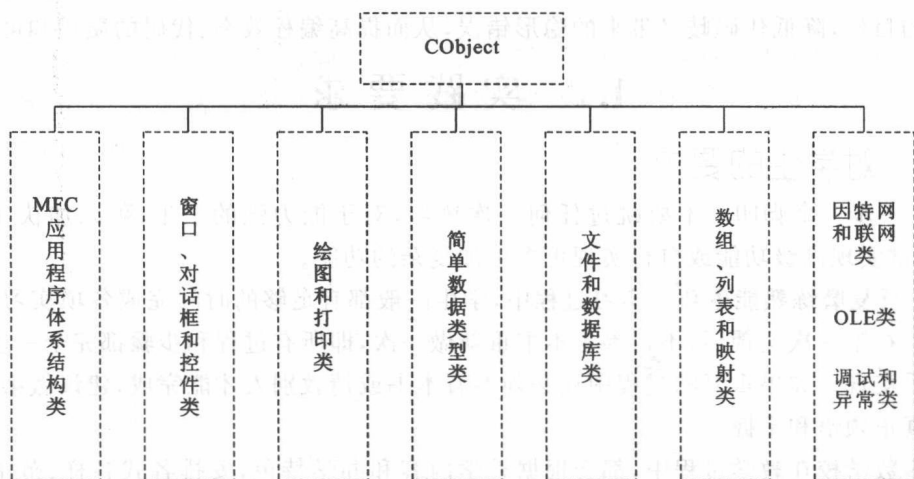


图 1.3 MFC 类汇总

1.2.5 程序调试

该部分要求与基础篇相同,目的在于强化。

要求理解调试对于提高软件开发效率和质量的重要性,掌握程序调试方法。

掌握断点设置、单步跟踪、变量查看等调试方法,习惯使用跟踪手段检查和优化程序。

程序调试是在程序正式发布之前,程序员借助集成调试环境,用手工方式逐行或逐过程走查代码,依据各语句的执行结果,确认代码与设计的一致性,以及实际性能与需求或设计的符合程度。通过跟踪检查,发现代码算法错误或性能偏差,进而改正错误或优化代码。

程序调试本质上是一种白盒测试,是原始代码经过不断修改快速摆动到最优的有效方法,是优秀程序员必须掌握的基本技能之一。

通过本次实习,要求掌握在 Visual Studio 2010 环境下,设置和取消断点、逐语句或逐过程跟踪程序的每一步,查看各步实际运行结果,验证结果的正确性等方法。要求熟练掌握这些方法对应的按键使用方法。

通过本次实习,希望学生能够灵活运用编译器、调试器等工具和代码浏览等手段实现高效编程。

1.2.6 编程规范化

该部分要求与基础篇相同,目的在于强化。

要求理解规范化编程对提高程序质量的重要性,掌握规范编程基本方法,形成良好的编程习惯。

软件规范的目的是统一软件的规范风格,提高软件源代码的可读性、可靠性和重用性,提高代码的质量和可维护性,减少软件维护成本。良好的编程规范可以改善软件质量,缩短软件开发时间,提升团队效率,简化维护工作。所以,掌握基本编程规范,形成良好的编程习惯对优秀程序员尤为重要。

通过本实习,要求了解附录 1 所列编程规范的基本要求,并在实习中反复认真应用,逐步形成使用简洁的语句编写代码、使用准确的语言编写注释的良好习惯,达到程序易读易理解的目标,降低代码歧义带来的隐形错误,从而提高编程效率、代码的质量和可靠性。

1.3 实践要求

1.3.1 对学生的要求

(1) 一步一个脚印。不要跳过任何一次练习,对于能力强的学生,可以加快实习进度,并尝试实现高级功能或自行实现更多更加复杂的功能。

(2) 反复磨练熟能生巧。实习过程中,学生一般都有足够的时间完成各项实习任务,建议学生在第一次实现后,不再参考本书重新做一次,即所有过程和步骤都完全一边回忆一边动手实现。如果重复的过程中还必须参看本书或请教别人才能完成,建议做第 3 次,以达到真正领悟和掌握。

大多数学校在教学过程中,都会根据教学内容和办学特色,安排各式各样、范围宽泛的练习,对学生动手能力的提高有利无害,但问题是很多学生并未真正掌握,没有达到实习目标甚至没有完成。

因此,理解每一项练习背后的知识,掌握练习要求的工具、过程和方法,就非常重要。要做到这一点,仅跟随老师或教程走一趟显然不够,学生必须重复练习内容直至掌握,课内时间如果不够学生课外要花时间,做到真理解,也就是不仅说得清、还写得出、做得对。

(3) 挑战自己多多益善。该系列练习中的空间数据管理分为基础和提高两部分,常规实习可以只完成基础部分,但建议学生能够完成提高部分,以便更好地理解 and 掌握数据库更多的知识和开发技能。

1.3.2 对老师的要求

(1) 记录过程。要求学生在实习过程中,填写实习记录,作为实习成绩评定的部分依据。

(2) 抽查代码。指导老师在实习过程中,要与学生交流,检查学生的实习情况,检查学生所写的代码,并抽取代码由学生解释其含义和用法。

(3) 检查结果。通过成果演示、答辩、代码抽查、部分功能重做等形式检查和评定学生掌握的程度。

第 2 章 背景知识概述

2.1 几何图形及其结构

2.1.1 点

点是几何图形最基本的单元,是空间中只有位置、没有大小的图形。在一个平面上,通常用一个有序坐标对 (x,y) 来表示一个点,其中 x 习惯上表示水平位置, y 表示竖直位置,如图 2.1 所示。

虽然一个有序坐标对可以确定一个点的位置,但由于点是现实世界中点状地物(如电杆、灯塔、泉水、水文站、气象观测点等)的抽象,种类繁多多样,所以除了空间位置,点还有一些属性,如种类、颜色等。

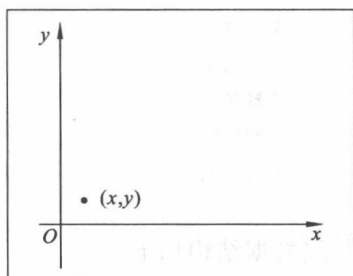


图 2.1 点坐标

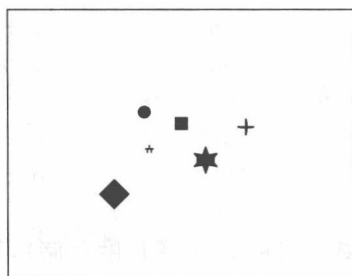


图 2.2 点状图形

在计算机中,为了记录和显示不同的点,通常每个点还会给一个唯一的编号,通常称为 ID;为了显示不同的图案以表达不同的含义,还要记录图案号。所以本教程点图形的结构如下。

```
struct{
    long      ID;           //点编号
    double    x;            //点位坐标 X
    double    y;            //点位坐标 Y
    COLORREF  color;        //点颜色
    int        pattern;     //点图案号
    char       isDel;        //是否被删除
}PNT_STRU;
```

2.1.2 线

线是现实世界中线状地物(如道路、河流、航线、电力线等)的抽象。当要记录一条线时,把所有的点记录下来显然是不切实际的,仅需要记录线上的一些“节点”就可以描述整条线。这些“节点”就是线的端点与转折点。所以,在计算机中,一条线用有限多个有序坐标点来表示,如图 2.3 所示。

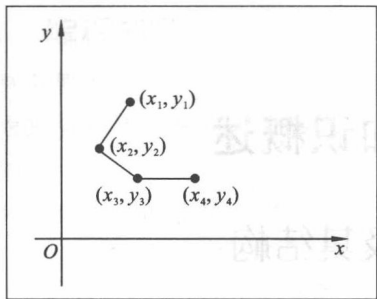


图 2.3 线坐标

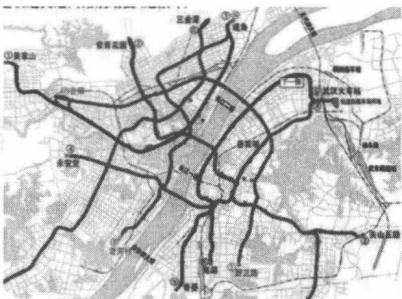


图 2.4 线状图形(武汉市道路图)

与点类似,线的种类也是多种多样的,除了节点序列,还有颜色、线型、种类等更丰富的属性。为了区分不同的线,每条线同样要分配一个唯一的 ID。

因为不同的线节点数不同,所以为了提高存储和检索效率,将每条线分两部分存储,一部分是长度固定的索引,另一部分是长度变化的“节点数组”。

线索索引结构如下。

```
struct {
    long        ID;           //线编号
    char        isDel;        //是否被删除
    COLORREF    color;        //线颜色
    int         pattern;      //线型(号)
    long        dotNum;       //线节点数
}LIN_NDX_STRU;
```

线节点数据:DOT(可变长度存储),其中单个节点数据结构如下。

```
struct {
    double      x;            //节点 x 坐标
    double      y;            //节点 y 坐标
}D_DOT;
```

2.1.3 区

区是现实世界中面状地物(如地块、湖泊、行政区等)的抽象,在计算机中,区是由平面上三个和三个以上的节点连接而成的封闭图形,即通过有序描述区边界的节点来描述一个最简单的区(因为有孔的区结构过于复杂,本书不讨论此类区),这样,最简单的区就是一个有限多个有序坐标点,如图 2.5 所示。

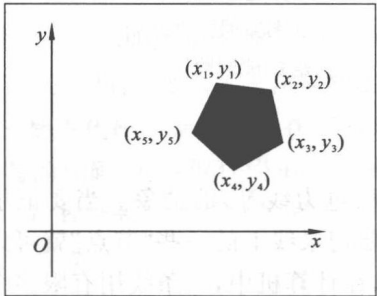


图 2.5 区坐标

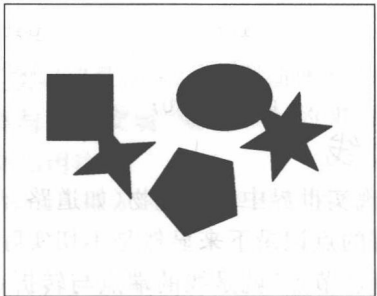


图 2.6 面状图形