admin

[公司名称] | [公司地址]

备忘录管理系统

VC++程序设计

# 摘 要

本报告介绍了一款基于对话框的MFC备忘录管理系统。该系统使用文本文件来存储备忘录信息，每条备忘录包含编号、姓名、日期和文本。该系统实现了增加、删除、修改和查询备忘录信息的功能，通过对话框界面使用户操作更加方便。本系统采用Visual Studio开发，使用MFC框架进行界面设计和程序开发，C++语言实现系统的各项功能。该系统为用户提供了一个简单、高效的备忘录管理工具，可应用于日常生活和工作中。

关键词：MFC C++ 备忘录 管理系统

目录

[摘 要 1](#_Toc132817220)

[**第一章 概述** 2](#_Toc132817221)

[1.1 课程设计目的 2](#_Toc132817222)

[1.2 预备知识 2](#_Toc132817223)

[**第二章 需求分析** 4](#_Toc132817224)

[2.1 系统目标 4](#_Toc132817225)

[2.2 功能分析 4](#_Toc132817226)

[2.3 开发环境 5](#_Toc132817227)

[2.4 用例图 6](#_Toc132817228)

[2.5 用例规约 6](#_Toc132817229)

[**第三章 概要设计** 8](#_Toc132817230)

[**3.1 系统流程图** 8](#_Toc132817231)

[**3.2功能详细分析** 8](#_Toc132817232)

[**第四章 详细设计** 10](#_Toc132817233)

[4.1 登录 10](#_Toc132817234)

[4.2 导入文件 11](#_Toc132817235)

[4.3 添加信息 12](#_Toc132817236)

[4.4 删除信息 13](#_Toc132817237)

[4.5 修改信息 14](#_Toc132817238)

[4.6 查询信息 15](#_Toc132817239)

[**第五章 心得体会** 17](#_Toc132817240)

[参考文献 18](#_Toc132817241)

# 

# **第一章 概述**

## 1.1 课程设计目的

本课程设计的主要目的是帮助学生通过实践掌握MFC框架下对话框应用程序的设计与开发方法，加深对C++编程语言的理解和应用。同时，本课程设计旨在培养学生的分析问题、解决问题的能力，提高学生的编程实践能力和团队合作能力。通过设计一个实用的备忘录管理系统，学生将深入了解软件开发的整个过程，包括需求分析、系统设计、编码实现、测试和维护等方面的知识和技能，提高学生的实际操作能力和实践经验。此外，本课程设计还能够让学生深入了解文本文件的读写操作，以及通过对话框界面实现用户交互的方法，为学生今后的软件开发和实践打下坚实的基础。

## 1.2 预备知识

为了更好地完成本课程设计，学生需要具备以下预备知识：

1. C++编程语言：学生需要具备一定的C++编程语言基础，包括基本语法、流程控制、数组、函数、指针等知识点，能够独立完成简单的C++编程任务。
2. MFC框架：学生需要了解MFC框架的基本概念和应用方法，包括对话框应用程序的设计和实现方法，以及MFC类库中常用类的使用方法。
3. Visual Studio开发环境：学生需要熟练掌握Visual Studio开发环境的使用方法，包括项目的创建和管理、界面设计和布局、调试和测试等方面的知识。
4. 文件读写操作：学生需要了解如何使用C++语言进行文件的读写操作，包括文件的打开、读取和写入等基本操作，以及错误处理和异常处理等技巧。
5. 数据结构：学生需要了解常用的数据结构，如数组、链表、栈、队列等，能够应用适当的数据结构来实现备忘录数据的存储和操作。

# **第二章 需求分析**

## 2.1 系统目标

本备忘录管理系统旨在提供一个高效、易用、可靠的备忘录管理工具，能够帮助用户轻松地记录、查看和编辑备忘录信息，满足用户对备忘录信息的需求。具体目标如下：

1. 提供一个直观、简洁的用户界面，使用户能够快速了解和使用备忘录管理系统的各项功能。
2. 实现备忘录信息的高效管理和维护，包括添加、删除、修改、查询、显示、保存、导入等操作。
3. 提供良好的用户体验，包括友好的提示信息、快速的响应速度、准确的数据处理等方面，确保用户能够愉快地使用备忘录管理系统。
4. 实现备忘录信息的可靠存储和保护，包括数据的加密和备份等措施，以确保备忘录信息的安全和可靠性。

通过以上目标的设定和实现，本备忘录管理系统将能够成为一个高效、实用、可靠的备忘录管理工具，满足用户的各种备忘录信息管理需求。

## 2.2 功能分析

本系统旨在设计一个基于对话框的备忘录管理系统，主要用于用户管理和维护备忘录信息。系统的目标是提供一个简单、高效、易用的备忘录管理工具，满足用户日常生活和工作中对备忘录信息的记录、查看和编辑需求。在系统设计和实现过程中，需要考虑以下功能需求：

1. 添加备忘录：用户能够添加新的备忘录信息，包括编号、姓名、日期和文本等。
2. 删除备忘录：用户能够删除指定的备忘录信息，根据备忘录的编号或姓名进行删除操作。
3. 修改备忘录：用户能够修改指定的备忘录信息，包括编号、姓名、日期和文本等。
4. 查询备忘录：用户能够查询指定的备忘录信息，根据备忘录的编号等条件进行查询操作。
5. 显示备忘录：用户能够查看所有备忘录信息，并以列表的形式展示备忘录的编号、姓名、日期等基本信息。
6. 保存备忘录：系统能够将所有备忘录信息保存在文本文件中，以便用户下次使用时能够读取和维护备忘录信息。
7. 导入备忘录：系统能够从外部文件中导入备忘录信息，并将其添加到当前备忘录列表中。

通过以上功能需求的分析，可以为系统设计和实现提供明确的指导和方向，保证系统能够满足用户的需求，并实现预期的功能目标。

## 2.3 开发环境

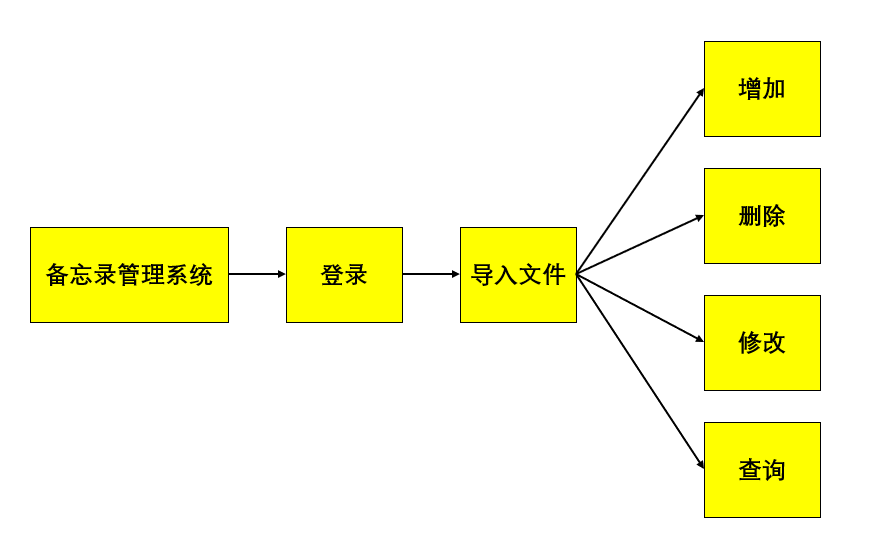
运行环境：windows 11

编译器：Visual Studio 2022

编程语言：C++

框架：MFC

## 2.4 用例图



## 2.5 用例规约

1. 用例名：登录

流程：在登录界面正确输入密码，点击“登录”按钮

2. 用例名：导入备忘录信息

点击“打开备忘录信息”按钮，选择需要导入的文件

3. 用例名：增加备忘录信息

点击“添加备忘录信息”按钮，在弹出的对话框中输入需要添加的信息，点击“确认”按钮

1. 用例名：删除备忘录信息

在列表中选择需要删除的信息，点击“删除备忘录信息”按钮，点击“确认”按钮

1. 用例名：修改备忘录信息

在列表中选择需要修改的信息，点击“修改备忘录信息”按钮，在弹出的对话框中修改信息，点击“确认”按钮

1. 用例名：修改备忘录信息

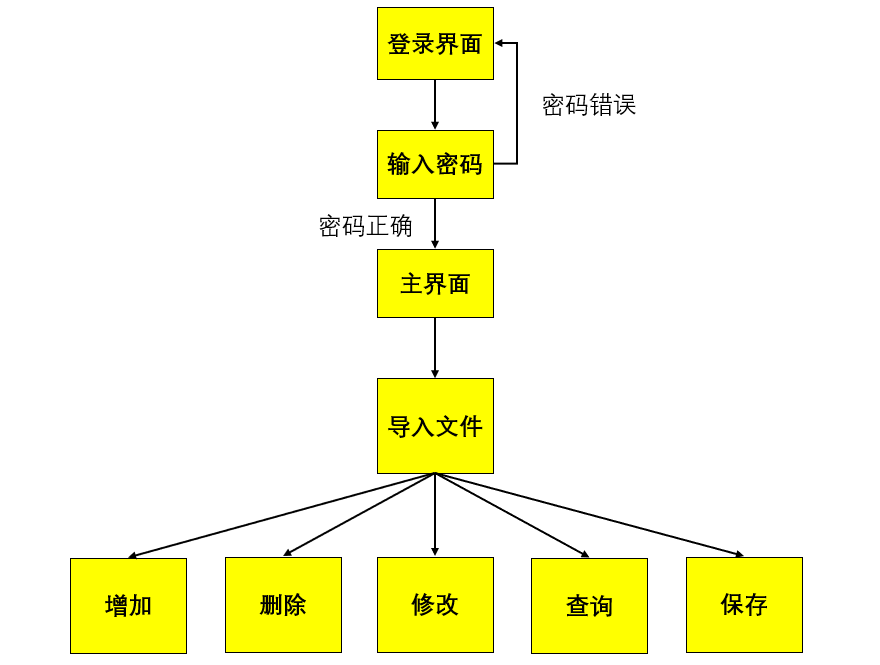
在编辑框中输入编号，点击“查找”按钮

1. 用例名：保存信息

点击“保存”按钮即可

# **第三章 概要设计**

**3.1 系统流程图**



**3.2功能详细分析**

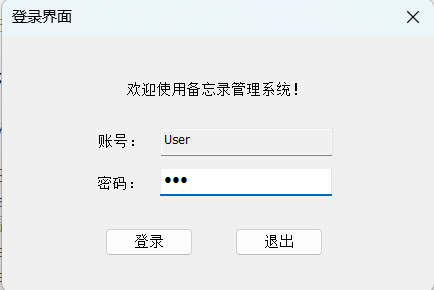
1. 添加备忘录信息：用户可以通过添加功能向备忘录管理系统中添加新的备忘录信息，包括编号、姓氏、名字、日期和文本等信息。
2. 删除备忘录信息：用户可以选择需要删除的备忘录信息，通过删除功能将其从备忘录管理系统中删除。
3. 修改备忘录信息：用户可以选择需要修改的备忘录信息，通过修改功能修改备忘录的编号、姓名、日期和文本等信息。
4. 查询备忘录信息：用户可以通过查询功能按照编号等信息查询备忘录信息，方便用户查找需要的备忘录信息。
5. 显示备忘录信息：备忘录管理系统将备忘录信息以列表的形式显示在程序的主界面上，方便用户查看和管理。
6. 保存备忘录信息：备忘录管理系统将备忘录信息保存在文本文件中，确保备忘录信息的持久存储。
7. 导入备忘录信息：用户可以通过导入功能将其他应用程序中的备忘录信息导入到备忘录管理系统中，方便用户集中管理备忘录信息。

# 

# **第四章 详细设计**

## 4.1登录

（1）功能描述：实现系统的登录功能，正确输入密码后会进入主界面



（2）实现代码：

ifstream infile;

infile.open("code.txt", ios::in);

if (!infile.is\_open())

{

cout << "读取文件失败" << endl;

return;

}

string buf;

infile >> buf;

UpdateData(TRUE);

if (m\_code.GetBuffer() == buf)

{

EndDialog(0);

CMemorandumSystemDlg dlg;

dlg.DoModal();

}

else

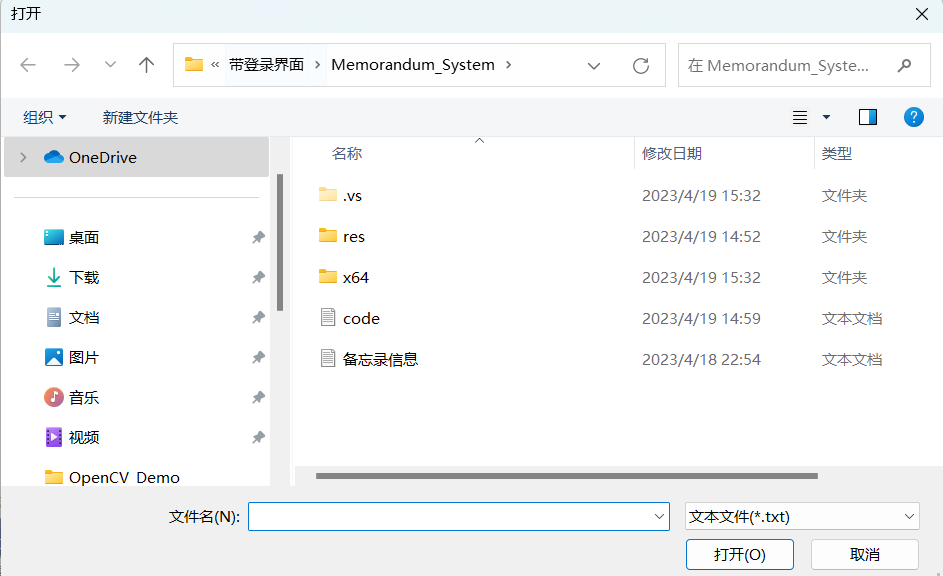
{

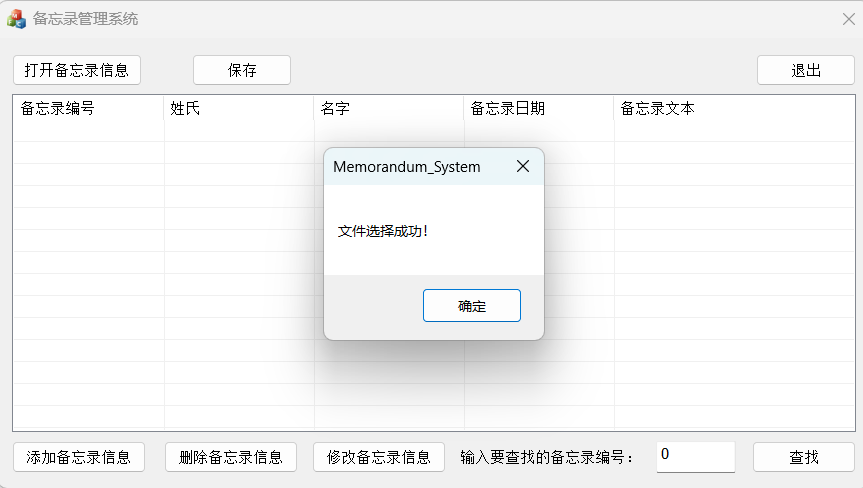
MessageBox(TEXT("密码错误！"));

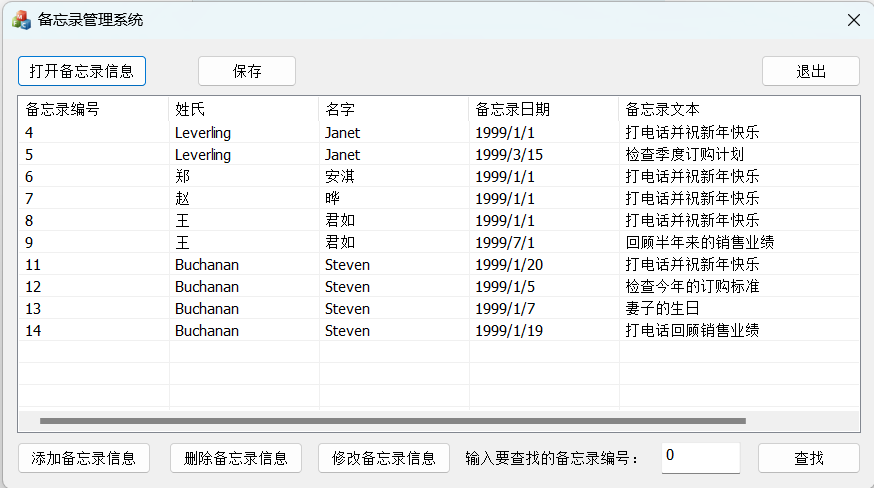
}

## 导入文件

（1）功能描述：实现文件导入功能，弹出对话框选择文件，用户在选择文件后系统将文件中的信息显示在列表上







（2）实现代码：

TCHAR szFilter[] = \_T("文本文件(\*.txt)|\*.txt|所有文件(\*.\*)|\*.\*|");

// 构造打开文件对话框

CFileDialog fileDlg(TRUE, \_T("txt"), NULL, 0, szFilter, this);

// 显示打开文件对话框

if (IDOK == fileDlg.DoModal())

{

// 如果点击了文件对话框上的“打开”按钮，则将选择的文件路径显示到编辑框里

filepath = fileDlg.GetPathName();

choose = 1;

MessageBox(TEXT("文件选择成功！"));

}

if (choose == 1)

{

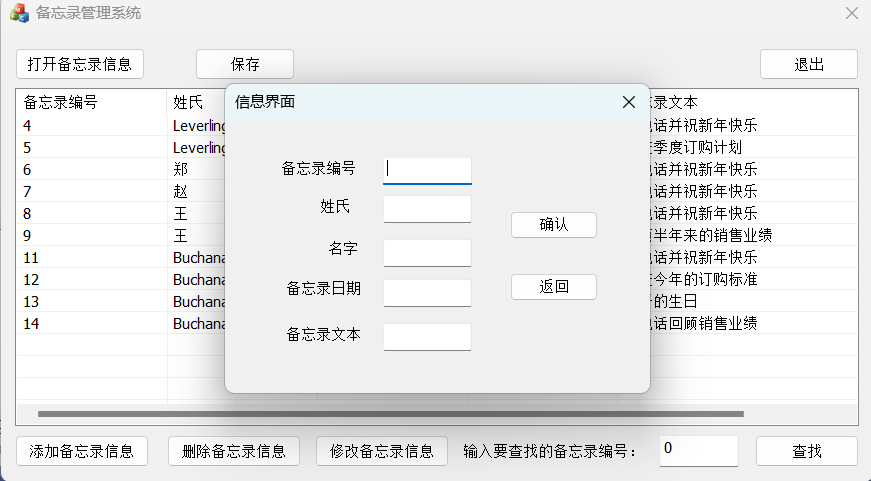
data.Load(filepath);

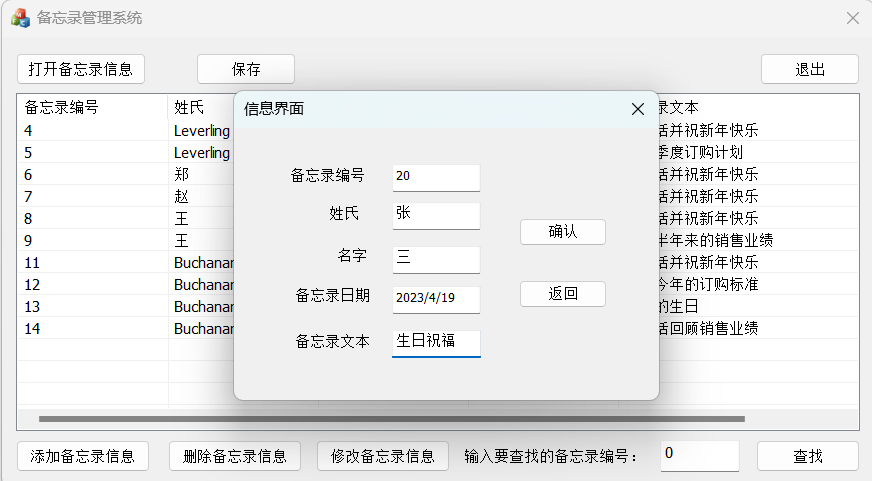
UpdateList();

}

## 添加信息

1. 功能描述：实现添加信息功能，用户在编辑框中输入信息，点击确认即可将添加的信息显示在列表上







（2）实现代码：

if (m\_ID == TEXT("")) m\_ID = TEXT("暂无");

if (m\_LastName == TEXT("")) m\_LastName = TEXT("暂无");

if (m\_FirstName == TEXT("")) m\_FirstName = TEXT("暂无");

if (m\_Date == TEXT("")) m\_Date = TEXT("暂无");

if (m\_text == TEXT("")) m\_text = TEXT("暂无");

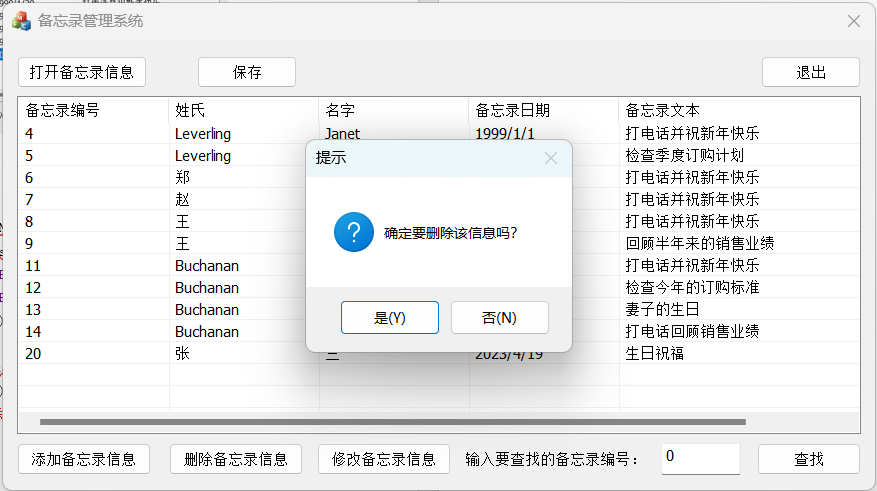
((CMemorandumSystemDlg\*)GetParent())->data.AddInfo(CInfo(m\_ID.GetBuffer(), m\_LastName.GetBuffer(),m\_FirstName.GetBuffer(), m\_Date.GetBuffer(), m\_text.GetBuffer()));

((CMemorandumSystemDlg\*)GetParent())->UpdateList();

MessageBox(TEXT("添加成功"), TEXT("提示"));

## 删除信息

1. 功能描述：实现删除信息功能，用户在列表中选择需要删除的信息，点击确认即可删除



（2）实现代码：

int nIndex = m\_list.GetSelectionMark();

if (nIndex > -1)

{

UINT i;

i = MessageBox(\_T("确定要删除该信息吗？"), \_T("提示"), MB\_YESNO | MB\_ICONQUESTION);

if (i == IDYES)

{

data.DelInfo(nIndex);

MessageBox(TEXT("删除成功！"));

UpdateList();

}

}

else

{

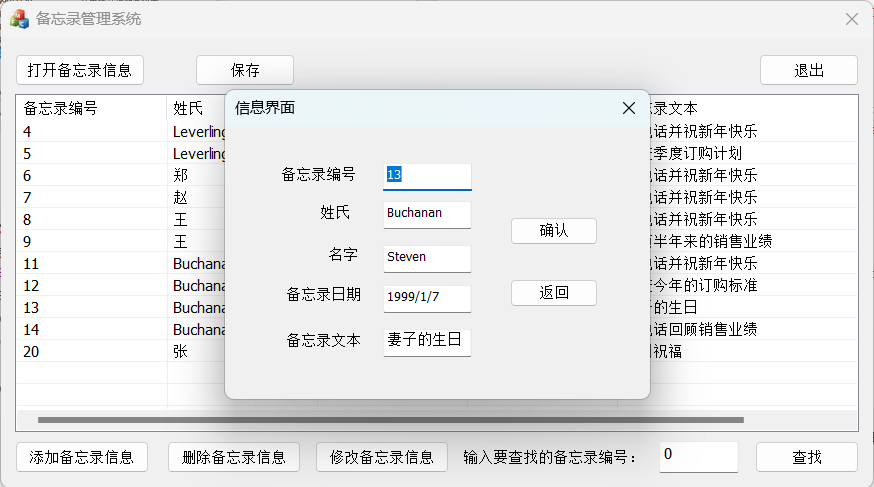
MessageBox(TEXT("点击列表选择要删除的数据！"), TEXT("提示"));

}

}

## 修改信息

（1）功能描述：实现修改信息功能，用户在列表中选择需要修改的信息，点击修改按钮，在对话框中即可修改



（2）实现代码：

int nIndex = m\_list.GetSelectionMark();

if (nIndex > -1)

{

select\_dlg = 2;

Info\_Dlg dlg;

dlg.m\_ID = data.m\_info[nIndex].m\_ID.c\_str();

dlg.m\_LastName = data.m\_info[nIndex].m\_LastName.c\_str();

dlg.m\_Date = data.m\_info[nIndex].m\_Date.c\_str();

dlg.m\_FirstName = data.m\_info[nIndex].m\_FirstName.c\_str();

dlg.m\_text = data.m\_info[nIndex].m\_text.c\_str();

dlg.index = nIndex;

dlg.DoModal();

}

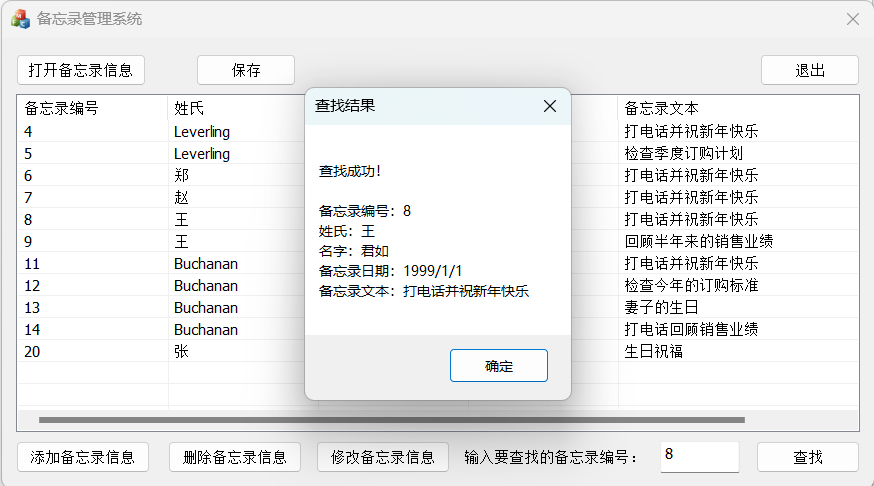
else

{

MessageBox(TEXT("点击列表选择要修改的数据！"), TEXT("提示")); }

## 查询信息

（1）功能描述：实现查询信息功能，用户在编辑框中输入要查询信息的编号，点击查找按钮即可弹出信息



（2）实现代码：

UpdateData(TRUE);

CString str;

str.Format(TEXT("%d"), m\_id);

for (int i = 0; i < data.m\_info.size(); i++)

{

if (data.m\_info[i].m\_ID == str.GetBuffer())

{

str.Format(TEXT("查找成功！\n\n备忘录编号：%s\n姓氏：%s\n名字：%s\n备忘录日期：%s\n备忘录文本：%s"),

data.m\_info[i].m\_ID.c\_str(), data.m\_info[i].m\_LastName.c\_str(),

data.m\_info[i].m\_FirstName.c\_str(), data.m\_info[i].m\_Date.c\_str(),

data.m\_info[i].m\_text.c\_str());

MessageBox(str, TEXT("查找结果"));

return;

}

}

MessageBox(TEXT("查无此人！"), TEXT("查找结果"));

# 

# **第五章 心得体会**

在本次的MFC备忘录管理系统设计中，我深刻体会到了软件开发的艰辛和挑战，同时也收获了许多宝贵的经验和知识。

首先，在需求分析和功能设计阶段，我学会了如何从用户需求出发，全面、系统地分析和设计软件的功能和界面。通过对备忘录管理系统的目标和功能进行深入的研究和思考，我逐渐明确了软件应该具有的功能和特性，并将其转化为可行的、易于实现的程序设计方案。这一过程让我对软件开发的流程和方法有了更为深入的认识和理解。

其次，在编码和调试阶段，我深刻体会到了编程的细节和技巧。在代码实现的过程中，我需要仔细分析每个功能的实现细节，避免因细节疏忽而导致的程序错误和异常。同时，我还需要熟练掌握各种编程技巧和调试方法，快速定位和修复程序中的问题。这一过程让我对MFC框架和C++语言有了更为深入的了解和掌握，也提高了我对编程的技能和能力。

最后，在整个开发过程中，我深刻认识到了软件开发的团队合作和沟通的重要性。通过与同学的交流和合作，我学会了如何协调和分配工作、如何有效地沟通和交流、如何相互学习和提高。这一过程让我认识到，软件开发需要一个高效、协调、合作、开放的团队环境，这是保证软件质量和成功的关键。

总的来说，本次MFC备忘录管理系统的设计，让我收获了许多经验和知识，不仅提高了我对软件开发的认识和能力，也锻炼了我的团队合作和沟通能力。希望在未来的学习和实践中，我能够更好地应用这些知识和经验，不断提高自己的能力和水平。

# 

# 参考文献

[1] 谭浩强.C语言程序设计[M].北京:清华大学出版社,2017

[2] 徐璇,姜明新,黄静,徐晶,李敏.基于MFC的工程软件界面设计[J].电子设计工程,2011,19(21):11-13.DOI:10.14022/j.cnki.dzsjgc.2011.21.040.

[3]孔垂柳.VC++中MFC框架技术探索[J].吉林建筑工程学院学报,2009,26(04):82-84.

[4]周山. 基于MFC和ACCESS数据库的学生信息管理系统[D].吉林大学,2009.