

**课程设计报告**

**职工工作量统计系统设计**

|  |  |
| --- | --- |
| 学 院 | 计算机与通信工程学院 |
| 专 业 | 计算机类 |
| 班级序号 | 200127 |
| 学 号 | **202012182** |
| 姓 名 | 姚辉 |
| 指导教师 | 柏禄一 李国瑞 |
| 验收日期 | 2020年12月25日 |

核心知识点清单，由学生确认

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数组 | 链表 | 指针 | 文件读写 | 默认参数 | 函数模板 | 多文件 | 类 | 派生 | 虚函数 | 友元函数 | 重载 | 多继承 |
|  | √ | √ | √ | √ |  | √ | √ |  |  |  |  |  |

以下为教师评分表，学生不可填写

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 程序质量  （60%） | 课程设计报告（20%） | 答辩效果  （20%） | 总分 | 等级 |
|  |  |  |  |  |

评分标准与说明：

1. 程序质量（60%）包含程序正确性与所用知识点数量（40%），代码可读性（10%）与界面友好性（10%）。
2. 课程设计报告（20%）要求排版规范，模块设计有文字说明，图、表、代码清单要有序号和名字。
3. 现场答辩（20%）要求根据学生制作的PPT、讲述清晰、回答问题等情况综合评分。
4. 收齐所有纸质报告的同时，要求学委收集所有学生的代码工程、报告电子版和答辩PPT以备存档。

目录

1 课程设计要求 3

2 题目分析 3

3 类结构设计 3

4 流程图 4

5工程结构组织 6

6核心代码解释 9

6.1核心代码1：文件保存 9

6.2核心代码2：文件读取 10

6.3核心代码3：添加信息 11

6.4核心代码4：删除信息 13

6.5核心代码5：排序算法 14

6.6核心代码6：查询显示方式 15

6.7核心代码7：登录设置 16

7 运行主要界面 17

8 课程设计所用知识点总结 **27**

9 项目总结 **27**

10 参考文献 **27**

【课程设计要求】

题目：职工工作量统计系统设计

**1、问题描述**

职工包括姓名、职工号、性别、年龄、所在部门、联系方式等信息。 工作量包括职工号、完成的产品数量等信息。 该设计系统能够对职工的工作量进行统计，并排出名次。注意，一个职工的工作量是可以多次输入的。

**2、功能要求**

（1）添加功能：程序能够添加职工的记录和工作量，可提供选择界面供用户选择所要添加的类别， 要求职工的编号要唯一，如果添加了重复职工号的职工记录时，则提示数据添加重复并取消添加。

（2）查询功能：可根据职工号、姓名等信息对已添加的职工记录进行查询。还可以按职工号对工 作量进行查询。如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。

（3）显示功能：可显示当前系统中所有职工记录和工作量统计信息，每条记录占据一行。

（4）编辑功能：可根据查询结果对相应的记录进行修改，修改时注意编号的唯一性。

（5） 删除功能：主要实现对已添加的人员记录进行删除和工作量进行删除。如果当前系统中没有 相应的人员记录，则提示“记录为空！”并返回操作；否则，输入要删除的人员的编号，根据所输入的 信息删除该人员记录，如果没有找到该人员信息，则提示相应的记录不存在。

（6）统计功能：对职工可按工作量进行统计，并排名次。

（7）保存功能：可将当前系统中各类人员记录存入文件中，存入方式任意。

（8）读取功能：可将保存在文件中的人员信息读入到当前系统中，供用户进行使用**。**

**3、问题的解决方案**

**根据系统功能要求，可以将问题解决分为以下步骤：**

**（1）应用系统分析，建立该系统的功能模块框图以及界面的组织和设计；**

**（2）分析系统中的各个实体及它们之间的关系；**

**（3）根据问题描述，设计系统的类层次；**

**（4）完成类层次中各个类的描述；**

**（5）完成类中各个成员函数的定义；**

**（6）完成系统的应用模块；**

**（7）功能调试；**

**（8）完成系统总结报告。**

【题目分析】

此题我选择采用链表的思路，设置了Date类作为数据模板用来存放数据，开了一个核心头指针head并且应用于全局，所有功能的实现围绕head展开。并且应用了mfc使得界面可视化，增加了用户友好性。

【类结构设计】

根据题目分析。主要运用了Date类，如图1所示

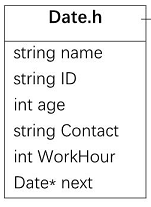


图1 Date类图

因为运用了MFC进行界面设计，所以类比较多。其中MFC为我自动提供了一些支持程序的类如targetver.h，pch.h，MFCApplication3.h，afxdialogex.h，framework.h，MFCApplication3Dlg.h，Resource.h。其中主菜单类是SystemDlg.h，我在其中定义了一些按键接口如afx\_msg void OnBnClickedButton1();在cpp实现文件中通过DoModal()弹窗函数实现调用。其中数据类是Date.h，我在其中定义了数据类型，因为是用链表的思路，所以也定义了链表指针。其中每一个功能我都单独开了一个类，如添加功能就是Add.h实现。具体各类之间的关系如图2所示例

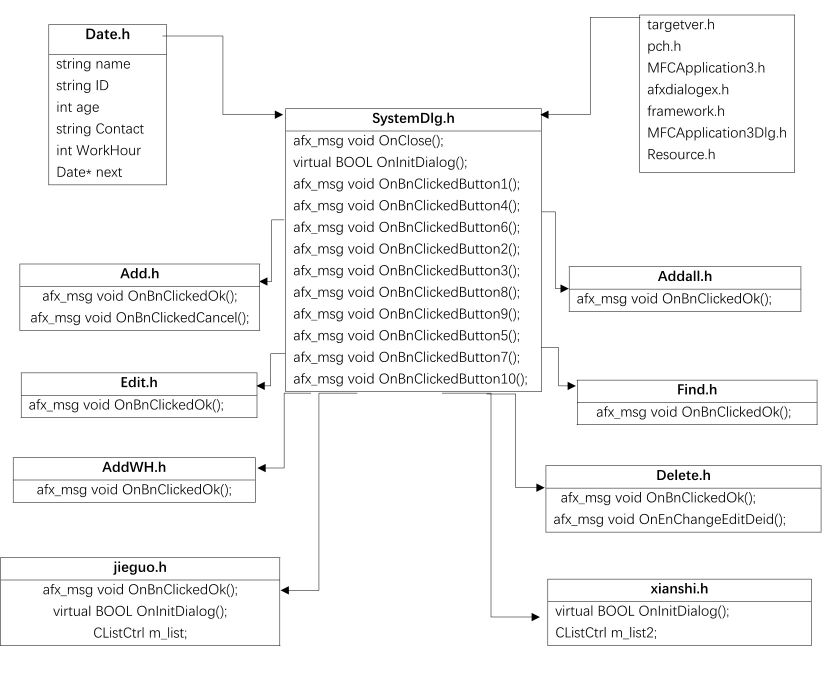
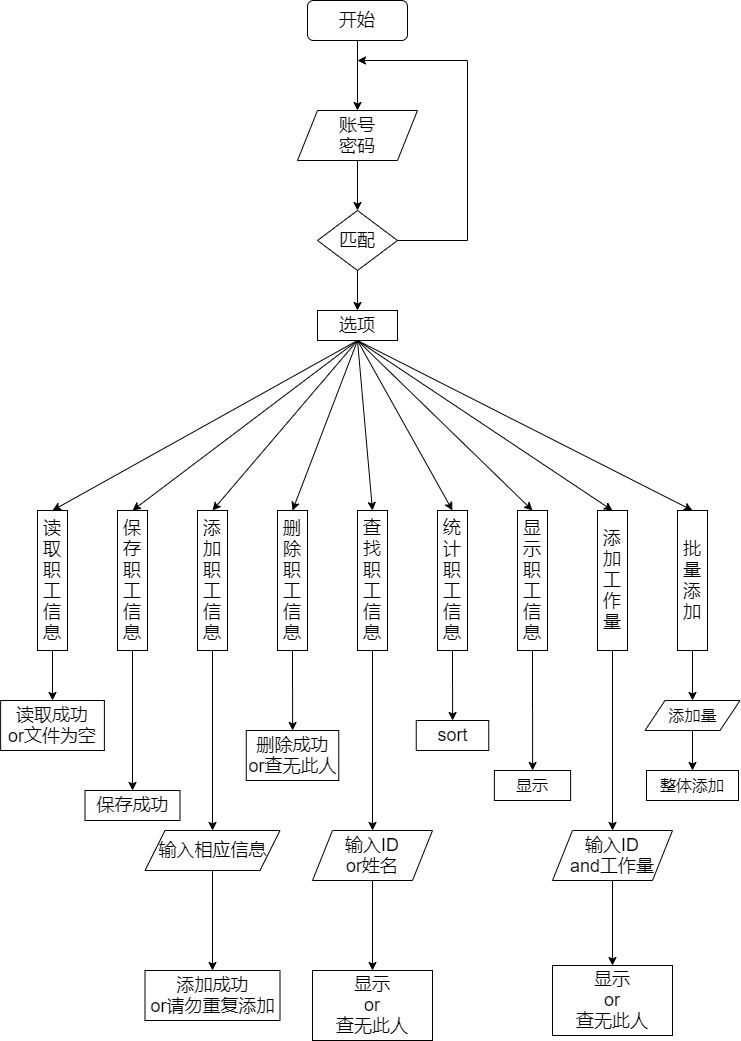


图2 类之间的关系

【流程图】



【工程结构组织】

本项目使用Visual Studio实现，工程文件结构图如图3所示。

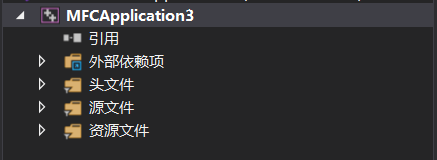


图3 工程文件结构图

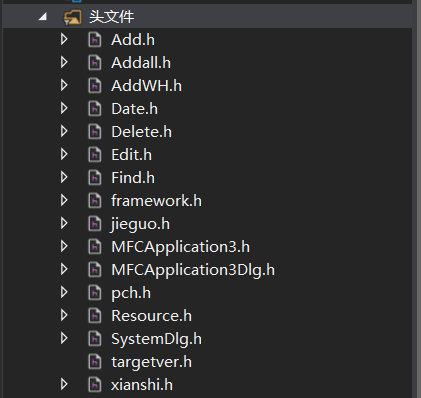


图4 头文件结构图

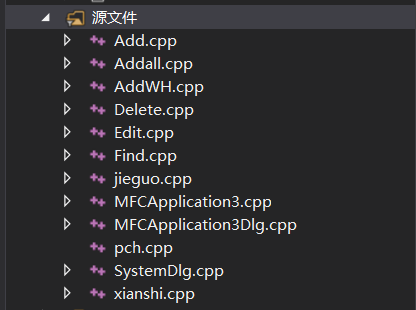


图5 源文件结构图

本项目主要由28个文件组成（不包括外部依赖项和资源文件），各文件的功能如下表（表1）所示。

表1. 本项目文件功能列表

|  |  |  |
| --- | --- | --- |
| 序号 | 文件名 | 说明 |
| 1 | Date.h | 数据类定义文件 |
| 2 | targetver.h | MFC系统头文件 |
| 3 | pch.h | MFC系统头文件 |
| 4 | pch.cpp | Pch实现文件 |
| 5 | afxdialogex.h | MFC系统头文件 |
| 6 | framework.h | MFC系统头文件 |
| 7 | Resource.h | MFC系统头文件 |
| 8 | MFCApplication3Dlg.h | 登录类定义头文件，包含账号密码 |
| 9 | MFCApplication3Dlg.cpp | 登录类定义实现文件 |
| 10 | MFCApplication3.h | MFC系统头文件 |
| 11 | SystemDlg.h | 菜单类定义头文件 |
| 12 | SystemDlg.cpp | 菜单类实现文件，在其中嵌入了文件读写以及排序功能 |
| 13 | Add.h | 添加功能类定义头文件 |
| 14 | Add.cpp | 添加功能类实现文件 |
| 15 | Addall.h | 批量添加功能类定义头文件 |
| 16 | Addall.cpp | 批量添加功能类实现文件 |
| 17 | AddWH.h | 添加工作量功能定义头文件 |
| 18 | AddWH.cpp | 添加工作量功能实现文件 |
| 19 | Delete.h | 删除功能类定义头文件 |
| 20 | Delete.cpp | 删除功能类实现文件 |
| 21 | Edit.h | 编辑功能类定义头文件 |
| 22 | Edit.cpp | 编辑功能类实现文件 |
| 23 | Find.h | 查询功能类定义头文件 |
| 24 | Find.cpp | 查询功能类实现文件 |
| 25 | jieguo.h | 结果显示功能定义头文件 |
| 26 | jieguo.cpp | 结果显示功能类实现文件 |
| 27 | xianshi.h | 显示功能定义头文件 |
| 28 | xianshi.cpp | 显示功能实现文件 |

【核心代码解释】

核心代码1：文件保存

如图6运用输入输出流进行文件保存，包含头文件<fstream>后定义一个ofstream的对象save，调用open函数以ios::out方式打开文件，如果文件不存在将会创建一个数据库的txt文件来保存信息，因为采用ios::out的方式，每次保存信息会覆盖之前的信息，避免了重复保存。保存方式是定义一个工作指针p，接收应用于全局的头指针head，遍历一遍链表通过重载运算符<<依次保存。保存成功后，调用close函数关闭文件，调用MFC的一个MessageBox函数进行一个保存成功的弹窗提示。

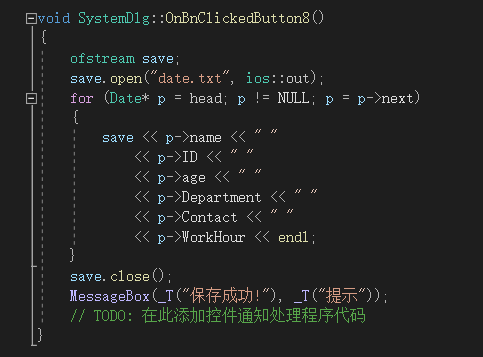


图6 文件保存

核心代码2：文件读取

如图7运用输入输出流进行文件保存，包含头文件<fstream>后定义一个ifstream的对象read，调用open函数打开文件，调用is\_open函数判断文件是否为空，如果文件不存在或者文件为空，将会调用MessageBox函数进行一个文件为空的弹窗提示；如果文件存在，则会通过while循环进行链表创建，其中定义一个工作指针node并为其开辟内存空间，通过read对象运用重载运算符>>初始化node的信息，然后判断node是头节点还是普通节点，循环创建，当node循环到文件末尾或者node里面的age为非正常值时候跳出循环。读入成功后，将会调用MessageBox函数进行一个读入成功的弹窗提示。创建链表的代码会再下一部分核心代码详细解释。

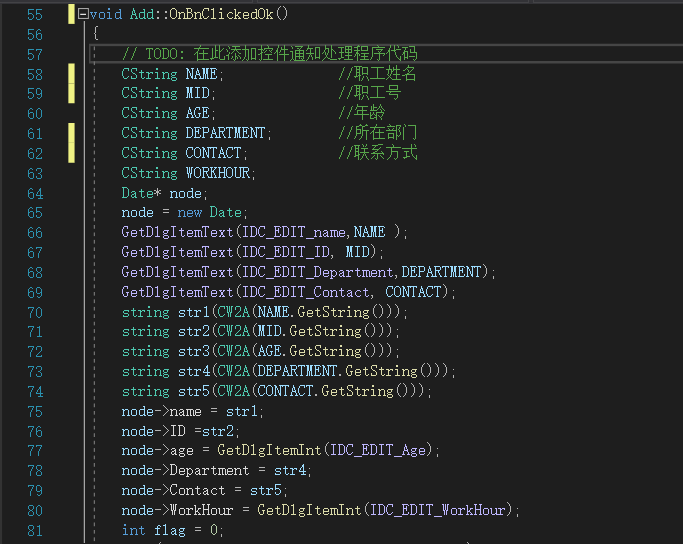


图7 文件读取

核心代码3：添加信息

如图8定义CString类型的数个变量与Date里面的数据对应，调用GetDlgItemText函数用于接收输入框内的文本类型的数值（其中变量是输入框的ID和接收变量的变量名），再调用CW2A函数进行强制类型转换，将CString类型的变量通过GetString函数转换成string类型存入工作指针node，调用GetDlgItemInt函数将输入框中的数据类型的变量转换为int型存入工作指针node。

图8 接收信息



如图9 node指针已被成功定义，在添加信息之前创建一个工作指针进行链表遍历，对新添加信息和原有信息的ID进行比较，如果重复将会调用MessageBox函数进行一个请勿重复添加的添加失败的提示。如果没有重复，则会进入创建链表的函数，先判断node是否为头节点，如果是则插入头节点，然后使next置空形成尾节点；如果是普通节点，则开一个尾指针tail，通过尾指针的特性next为空循环查找尾节点，找到后使node的next置空形成新的尾节点，再将tail的指针指向新的尾节点。通过flag的值判断是否创建成功，创建成功后将会调用MessageBox函数进行一个添加成功的弹窗提示

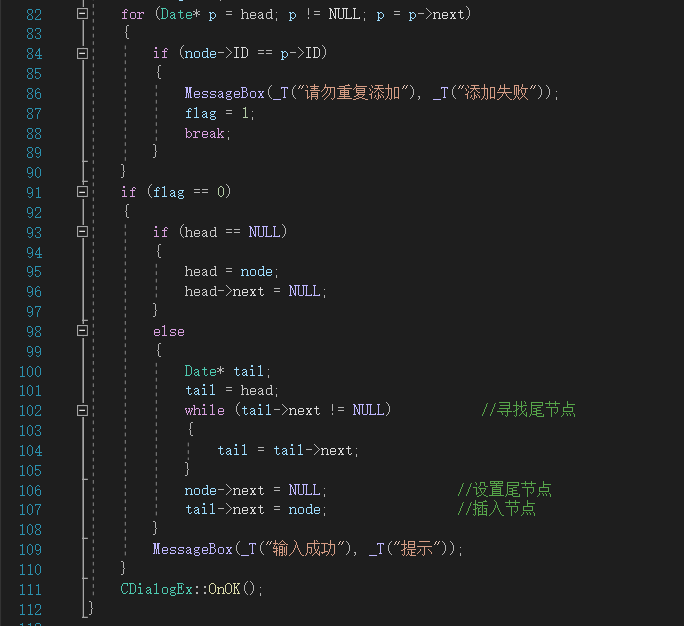


图9 插入链表

核心代码4：删除信息

同样的思路获取输入框中的值，循环查找要删除的节点，依次判断节点类型，分头节点，普通节点，尾节点三类删除，通过flag的值判断是否删除成功，后调用MessageBox函数进行一个添加成功的弹窗提示

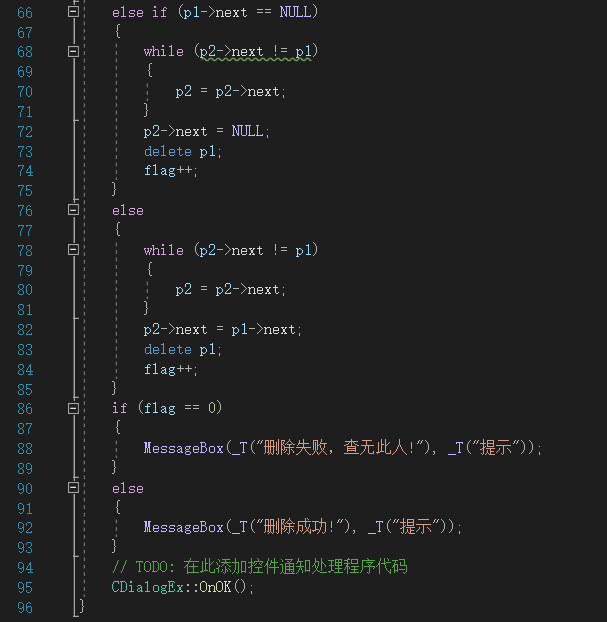
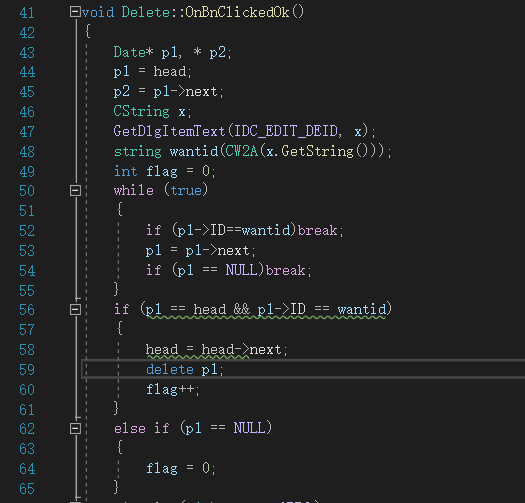


图10 删除信息

核心代码5：排序算法

运用选择排序的思想，对链表进行选择排序，其中不改变链表的指向，只交换其数据，实现从小到大排序（因为后期显示运用MFC的链表函数会逆序输出，所以这里排序用了从小到大）。

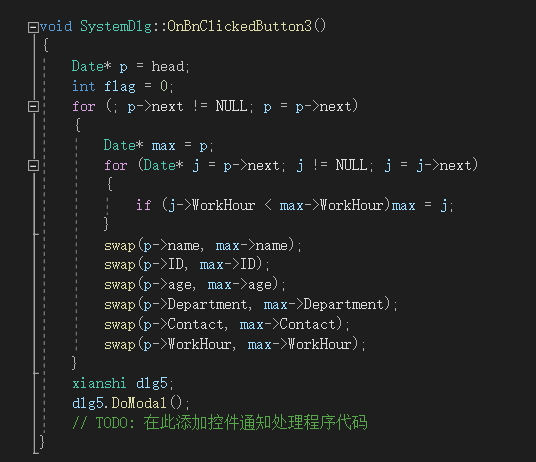


图11 排序算法

核心代码6：查询显示方式

如图12，设置一个指向父窗口的指针，用MFC内函数GetDlgItemtExt函数获取父窗口内输入框的值并将其转换成string类型，遍历链表进行比较，找到后运用之前设置过的列表，先用OnInitDialog对其进行初始化，然后调用InsertColumn函数进行输出，四个参数分别为列数，名称，行数，宽度。这里还是运用到了CString和string的相互转换。

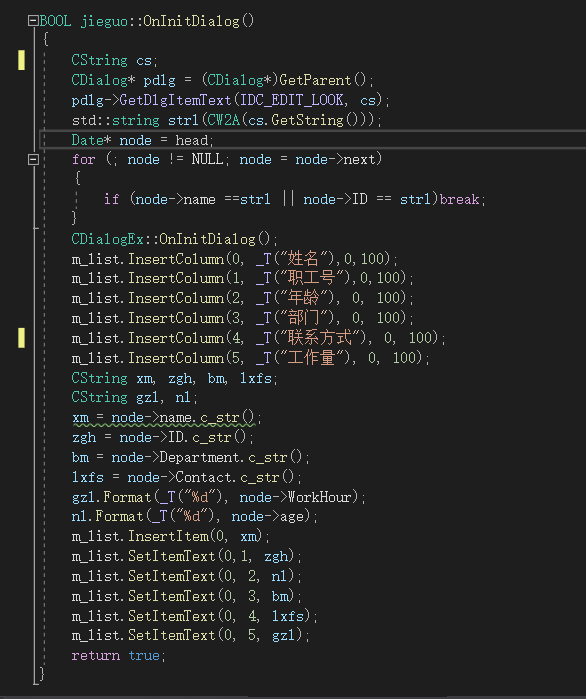


图12 查询及显示

核心代码7：登录设置

如图13定义两个CString数据类型的变量来存储输入框中的用户名和密码，运用CompareNoCase函数进行忽略大小写的比较，比较成功则可以通过DoModal弹窗进入菜单界面，比较失败则调用MessageBox函数进行一个用户名或密码正确的登录失败弹窗提示。

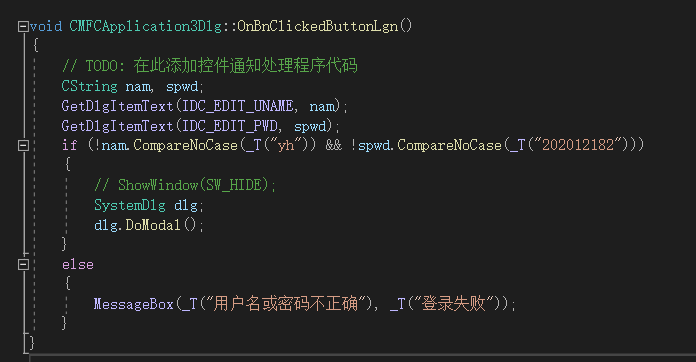


图13 登录设置

【运行主要界面】



图14 登录界面

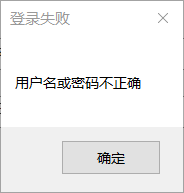


图15 登录失败提示

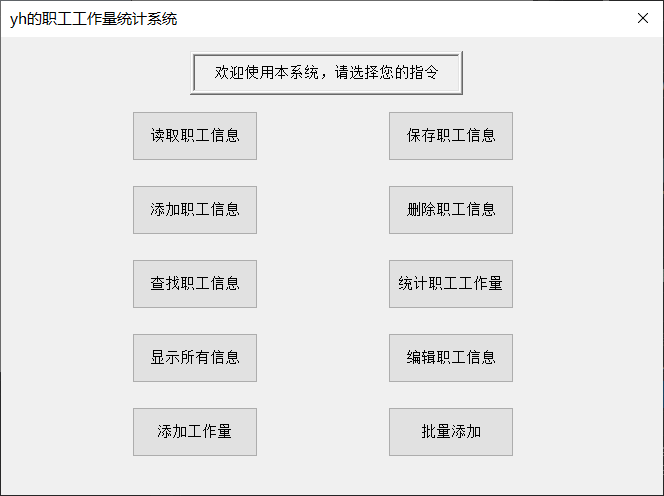


图16 菜单界面



图17文件读入提示



图18 文件保存提示



图19 所有信息显示

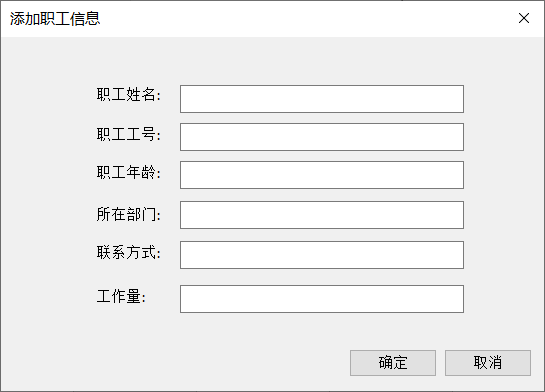


图20 添加职工工作量

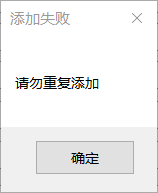


图21 添加失败提示

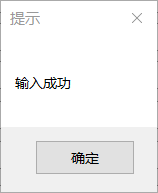


图22 添加成功提示

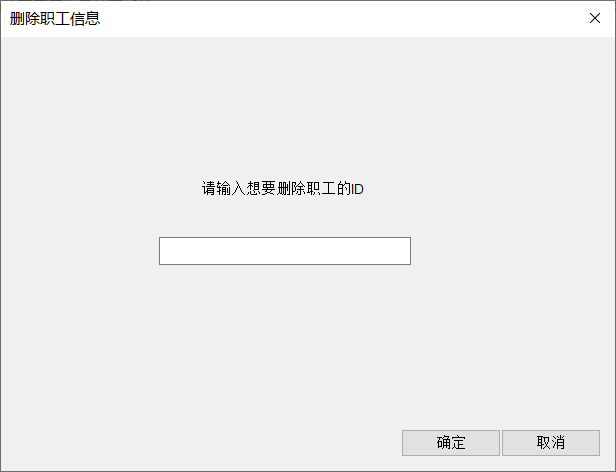


图23 删除职工信息

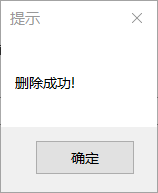


图24 删除成功提示

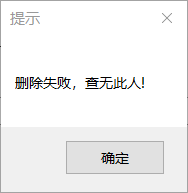


图25 删除失败提示

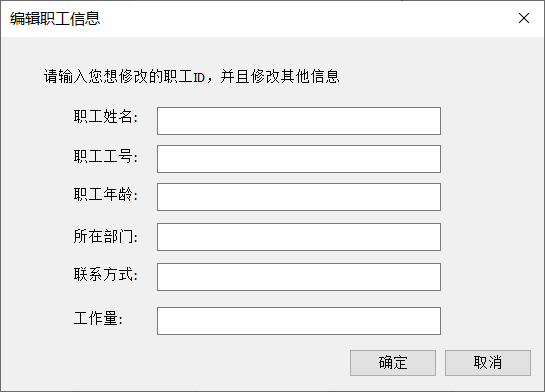


图26 编辑职工信息

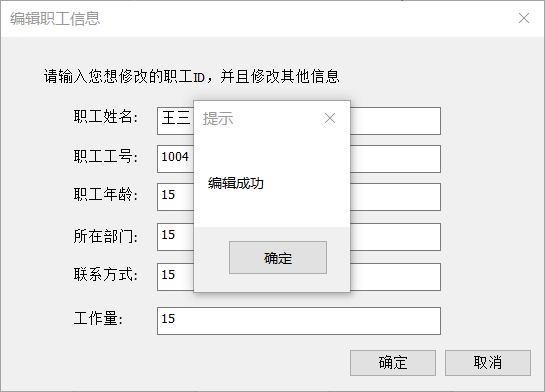


图27 编辑成功提示

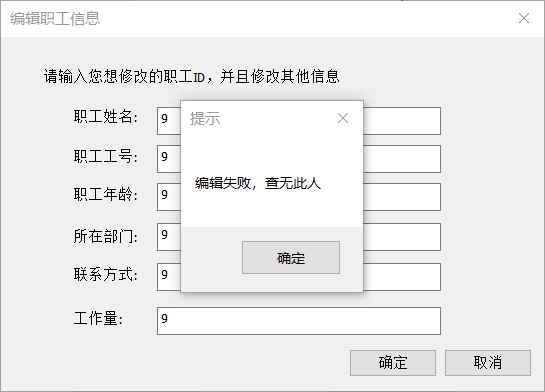


图28 编辑失败提示

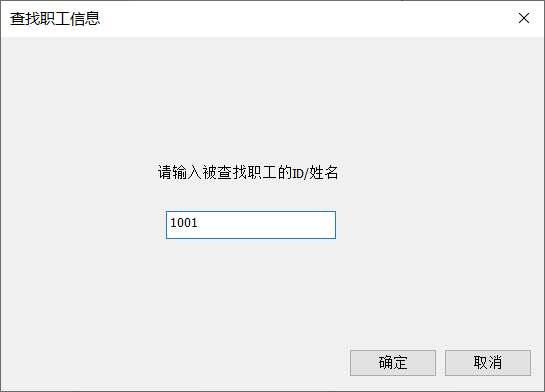


图29 查找职工信息

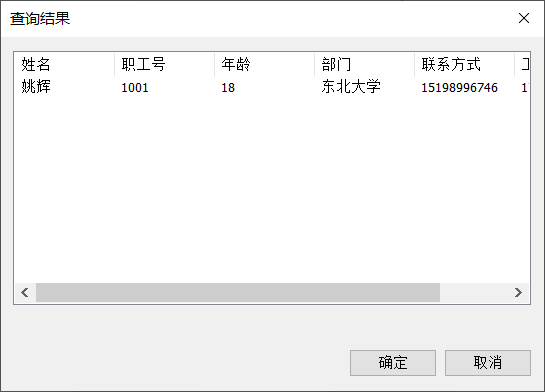


图30 查找成功提示

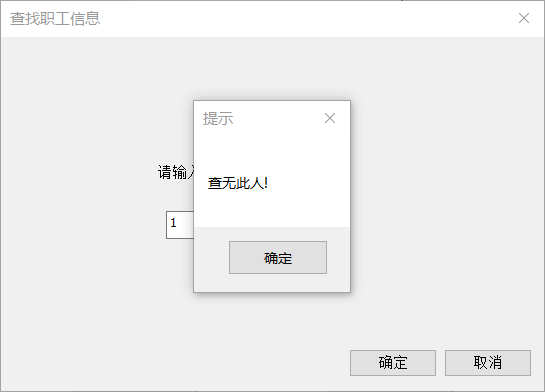


图31 查找失败提示

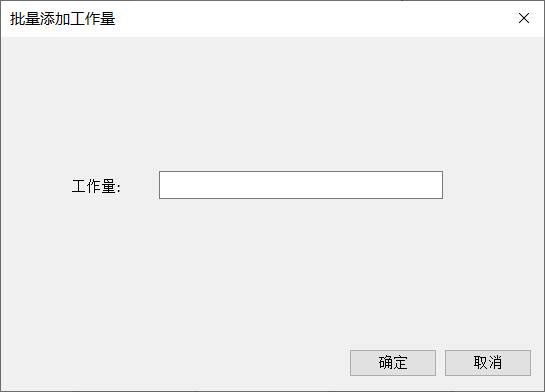


图32 添加工作量



图33 批量添加工作量

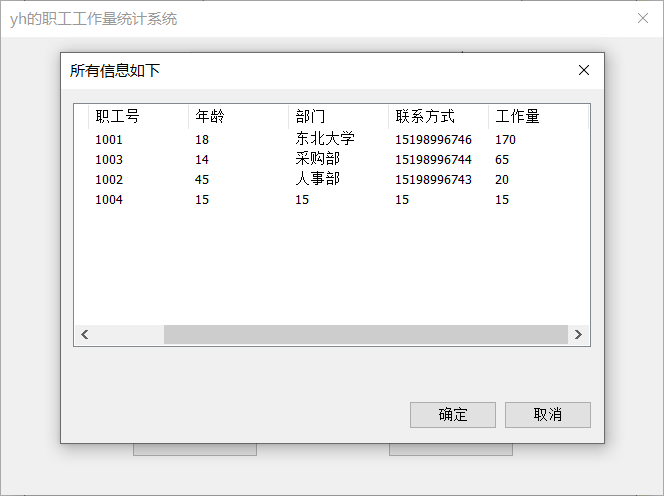


图34 统计结果显示

【课程设计所用知识点总结】

表2. 本项目所涉及知识点总结

|  |  |  |
| --- | --- | --- |
| 序号 | 知识点 | 说明 |
| 1 | 函数 | 本系统定义了很多个函数，包括带返回值、传入基本数据类型、指针、引用类型 |
| 2 | 链表 | 本项目定义了类链表，用于存储数据 |
| 3 | 文件 | 本项目使用txt文件做数据持久化存储 |
| 4 | 类与对象 | 定义了16个类，分别实现了不同功能 |
| 5 | 安全头文件包含 | 类自动包含 |
| 6 | 多文件 | 运用了多文件的处理方式，使项目层次清晰 |
| 7 | 算法 | 运用了选择排序的思想对工作量进行排序 |

【项目总结】

设计过程：我是先把整个思路设计好，然后系统的把代码敲完，逐步调试，完善漏洞，最后去自学了MFC把代码拆解进MFC里面完成可视化编程。

得意之处：我的所有数据存储是用链表完成，优化了内存利用，并且利用MFC使界面可视化。把重复出现的部分全部封进函数，精简了代码量。

存在的问题：我的测试并不是很完美，必然存在我没发现的漏洞，但是我也只能尽量去的完善。

心得与体会：写代码比较顺利，一气呵成，但是debug的时间却远远大于写代码时间，说明自己的基础不够扎实，逻辑思维不够严谨，还需继续努力。通过本次项目编写，我对类、链表、指针的知识点得到了进一步巩固，锻炼了全局思维。另外，在此项目编写过程中不仅我的代码能力得到提升，而且自学能力也得到了提升，我遇到不清楚的地方就打开网课看一看，知识点也就更清楚。而且我自学了MFC实现了可视化界面编程，这是我的得意之处，希望以后能有更多这样的项目机会锻炼自己。

【参考文献】

[1]谭浩强. C++语言程序设计（第3版)[M]. 北京：清华大学出版社.2015.

[2]C++Reference. [http://www.cplusplus.com[EB/OL].](http://www.cplusplus.com[EB/OL%5d.)

[3]CSDN