

《面向对象程序设计》三级项目

**人员信息管理系统**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 分工 | 自评分 |
| 李培显 | 202311040030 | 23-计算机2班 | 增删查改等基本代码框架 | 34% |
| 赵国兴 | 202311040035 | 23-计算机2班 | 集合，队列的实现（日程表） | 33% |
| 滕赛豪 | 202311040029 | 23-计算机2班 | 栈的实现与应用，实验报告总结 | 33% |

指导教师：王倩

开发日期：2024.4.24

目录

[1.项目特色创新点 5](#_Toc164987792)

[1.1 文件读人员信息，确保数据不丢失 5](#_Toc164987793)

[1.2 日期查询与计算，精确到秒没烦恼 5](#_Toc164987794)

[1.3 自定义导出功能，条件区间全覆盖 6](#_Toc164987795)

[1.4 可撤销删除操作，杜绝误删误操作 6](#_Toc164987796)

[1.5 对类的输入输出，方便易读可维护 6](#_Toc164987797)

[1.6 日程表打卡功能，制定清晰的计划 6](#_Toc164987798)

[2.总体设计 6](#_Toc164987799)

[2.1模板类链表list 10](#_Toc164987800)

[2.2学生类和老师类 11](#_Toc164987801)

[2.3增删查改功能 11](#_Toc164987802)

[3.特殊功能 12](#_Toc164987803)

[3.1撤销删除功能 12](#_Toc164987804)

[3.2日程表打卡功能 13](#_Toc164987805)

[3.3日期计算功能 13](#_Toc164987806)

[3.4自定义导出人员信息功能 14](#_Toc164987807)

[结论 14](#_Toc164987808)

人员信息管理系统

李培显，赵国兴，滕赛豪

（燕山大学 信息科学与工程学院）

摘 要：

本项目旨在使用C++以面向对象方式开发一个人员信息管理系统，用于有效管理学生和教师的相关信息。系统具备查询、统计、添加和删除，自定义导出人员信息，日期查询与计算等功能。使用双向链表和栈与队列等数据结构，满足特定功能需求， Node类负责结点初始化和内存回收，List类负责链表整体管理，包括初始化和回收。通过List类派生出Stack用来恢复删除人员信息和Queue完成打卡签到功能，以及Set类，实现特定操作。运用topk等算法实现自定义导出人员信息等功能。

**关键词：**

人员信息管理系统；教师与学生类；C++面向对象；双向链表；栈与队列；数据结构；topk应用，日程表打卡

**前言：**

随着信息化时代的来临，数据管理成为各类组织不可或缺的一部分。特别是在教育领域中，学生和教师的信息管理至关重要，它关乎到学校运营的效率、教学质量的监控以及资源分配的合理性。因此，开发一套高效、便捷的人员信息管理系统显得尤为重要。

本项目的研究报告旨在详细阐述一个人员信息管理系统的设计与实现过程，该系统主要针对学校场景，用于管理学生和教师的学号，姓名，得分/工资等各类信息。通过该系统的应用，能够实现对人员信息的快速查询、准确统计、灵活添加以及安全删除等功能，从而大大提升学校日常管理的效率与准确性。

在相关领域的工作中，已经有许多人员信息管理系统的研究与开发实践。然而，不同的系统往往针对特定的需求和环境进行设计，其功能与性能也各有千秋。本项目在充分调研现有系统的基础上，力求设计出一个功能全面、性能优越、操作简便的人员信息管理系统，以满足学校对于人员信息管理的实际需求。

本报告将详细介绍该系统的整体架构、模块设计、功能实现等方面的内容。同时，我们还将对系统开发过程中遇到的问题进行分析，并给出相应的解决方案。通过本报告，读者可以全面了解该系统的设计与实现过程，以及我们在项目中所做的工作和取得的成果。

此外，本报告还将介绍项目组的分工情况。我们根据项目需求，给每个人派发了不同任务，每个人负责系统的不同模块或功能。李培显负责Student，Teacher，List类的定义与增删查改基础功能的实现，赵国兴负责栈和队列以及Set类的实现与实验报告编辑，滕赛豪负责栈的实现及与代码测试，通过明确的分工与协作，我们得以高效地推进项目的开发工作，并最终实现预期的研究结果。

综上所述，本项目研究报告旨在呈现一个功能全面、性能优越的人员信息管理系统的设计与实现过程，以期为相关领域的研究与实践提供参考与借鉴。

功能

**需求分析：**

功能需求：一个人员信息管理系统必须要有增删查改四项基本功能，除此之外，系统应能够记录和维护每位员工（包括学生和教师）的基本信息，如姓名、工号（学号）、职位（身份）、部门（班级）、工资等。这些信息应支持搜索、排序，导出和过滤，以方便管理人员进行员工数据的查找和分析。同时应该有一些额外的功能来达到更好的使用体验和效果。

操作需求：人员信息管理系统通常面向的是非专业的计算机用户，因此系统应具备良好的用户界面和易用性如简单易懂的菜单。同时，系统也应具备易维护性如规范代码书写风格，以便管理员或开发人员能够方便地进行系统的维护和管理。

安全性

安全需求：人员信息管理系统涉及大量的敏感数据，如教师，学生个人信息、薪资信息等。因此，系统应具备高度的安全性，应将类成员设为私有，防止非法访问与篡改。

1.项目特色创新点

文件读人员信息，确保数据不丢失

本程序会在每次程序结束后将学生与教师信息分别储存在对应文件中，在下次程序开始时自动读取，确保人员信息不会随程序结束而丢失。

日期查询与计算，精确到秒没烦恼

用户可在不联网的情况下一键查询当前时间，且精确到秒。除此之外，用户还可以计算两日期间隔天数，如今天离五一/周末/高考还有多少天，用户还可以计算n天后对应日期。

自定义导出功能，条件区间全覆盖

用户不仅可以一键导出所有教师/学生信息，还可以按照需求，自定义条件区间，导出符合条件（排名，分数，工资等）的人员信息.

可撤销删除操作，杜绝误删误操作

通过用栈将被删除的人员信息入栈，需要恢复信息时出栈，实现了将不小心删除的人员信息删除的功能，提高用户操作容错率

对类的输入输出，方便易读可维护

通过运算符重载，Node类与Teacher和Student的对象支持cout<<输出，提高了程序的可读性与可维护性

日程表打卡功能，制定清晰的计划

通过队列先进先出的特性，实现用户可以手动加入日程，打卡日程，并且一键查询当前剩余的所有日程，可以用来当做课程表。

2.总体设计

首先定义了Person这一虚基类，后用Teacher与Student类继承Person，用来存储不同身份数据，Node为类模板，其中以Student或Teacher作为data，Node本身则是List的节点，而List则用来管理整个链表，实现增删查改，自定义导出人员信息，文件信息读写等功能，用Stack类继承List来支持恢复最近被删除的人员信息，Queue和Set类也继承自List，以支持实现其功能。此外，有独立的Date类用来日期的实时查询与计算以及独立的日程表类ScheduleItem来实现日程表操作。

本项目中main函数确定了程序的结构，main函数首先实例化了两个链表对象和两个栈对象用于，在中间部分，main函数通过用户输入调用相应函数，do while结构保证了程序在用户不主动退出情况下循环运行。Main函数最后部分释放了向程序申请的空间。

主函数如下

int main()

{

system("title 信息管理系统");

// Queue<course>\* schedule = new Queue<course>;

List<Student>\* StuList = new List<Student>;

List<Teacher>\* TchList = new List<Teacher>;

Stack<Student>\* StuStack = new Stack<Student>;

Stack<Teacher>\* TchStack = new Stack<Teacher>;

Schedule\* pschedule\_stu = new Schedule;

Schedule\* pschedule\_tch = new Schedule;

int choice = 1;

do

{

menu\_begin();

choice = \_getch()-48;

cout<<choice<<endl;

if (choice==1)

{

//教师管理信息系统

int choice\_tch = 1;

do{

menu\_tch();

choice\_tch = \_getch()-48;

cout<<choice\_tch<<endl;

switch (choice\_tch)

{

case 0:

cout<<"成功退出教师信息管理系统\n任意键继续"<<endl;

break;

case 1:

TchList->add();

break;

case 2:

TchList->remove(TchStack);

break;

case 3:

TchList->find();

break;

case 4:

TchList->change();

break;

case 5:

TchList->getall();

break;

case 6:

TchList->topk();

break;

case 7:

TchStack->show();

break;

case 8:

TchList->add(TchStack->Pop());

break;

case 9:

schedule(pschedule\_tch, "学生");

break;

default:

cout<<"输入有误，请重试"<<endl;

break;

}

if(choice\_tch>=1&&choice\_tch<=9)cout<<"操作完毕!\n(任意键继续)"<<endl;

\_getch();

}while(choice\_tch);

}

else if(choice==2)

{

//学生管理信息系统

int choice\_stu = 1;

do{

menu\_stu();

choice\_stu = \_getch()-48;

cout<<choice\_stu<<endl;

switch (choice\_stu)

{

case 0:

cout<<"成功退出学生信息管理系统\n任意键继续"<<endl;

break;

case 1:

StuList->add();

break;

case 2:

StuList->remove(StuStack);

break;

case 3:

StuList->find();

break;

case 4:

StuList->change();

break;

case 5:

StuList->getall();

break;

case 6:

StuList->topk();

break;

case 7:

StuStack->show();

break;

case 8:

StuList->add(StuStack->Pop());

break;

case 9:

schedule(pschedule\_stu,"教师");

break;

default:

cout<<"输入有误，请重试"<<endl;

break;

}

if(choice\_stu>=1&&choice\_stu<=9)cout<<"操作完毕!\n(任意键继续)"<<endl;

\_getch();

}while(choice\_stu);

}

else if(choice==3)

{

date();

}

else

{

if(choice!=0)cout<<"输入有误，请重试"<<endl;

}

}while(choice);

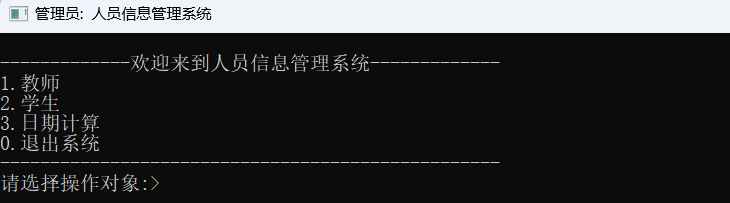
cout<<"成功退出系统\n"<<endl;

delete StuList;

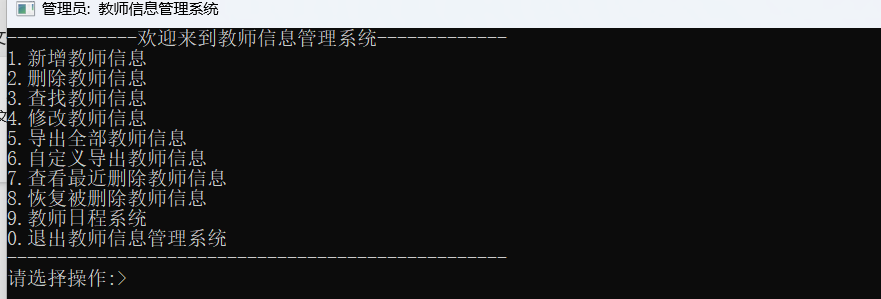
delete TchList;

return 0;

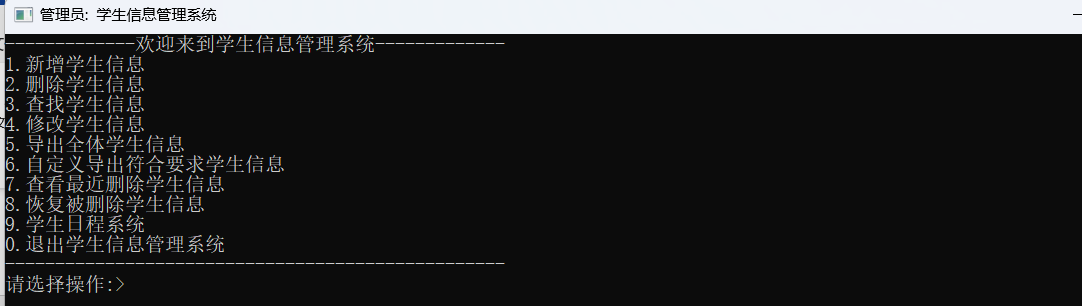
}

主界面：

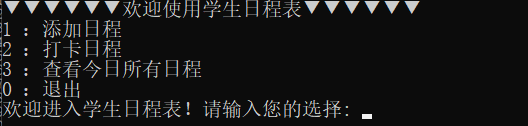
教师系统功能：



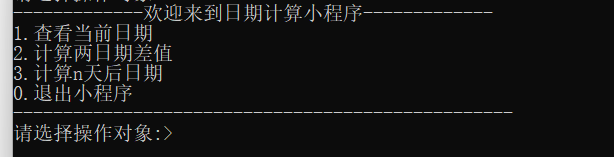
学生系统功能：



日期表系统功能：



日期计算系统功能：



2.1模板类链表list

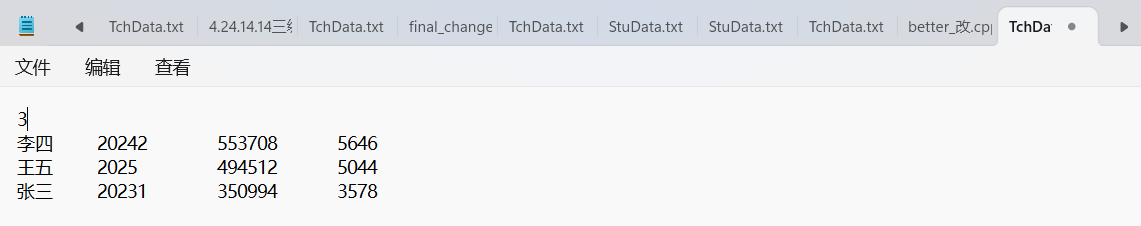
List构造函数会自动从相应文件（没有则自动创建）读取上次运行后存储的数据，自动根据从文件中导入数据

~List析构函数会将List中信息导入相应文件，确保人员信息不会随程序结束而丢失

~List析构函数:

该成员函数会将在本次程序运行结束后，自动将教师链表和学生链表中的数据保存至源文件所在目录下的TchData.txt，StuData.txt文件中并释放各Node空间，若没有相应文件则自动创建。数据在文件中的保存形式为：首行记录数据个数，之后每行记录一名人员信息。

文件保存内容：



List构造函数

该成员函数会从上次运行结束后保留信息的文件中读取数据并创建包含这些数据的链表。没有相应文件则创建空链表。首先，通过getclas（）得到要打开的文件的名称。之后，先读取文件第一个数字确定文件中数据数量，之后循环将该文件中数据通过void add(Node<T>\* new\_node)加入到相应链表中

2.2学生类和老师类

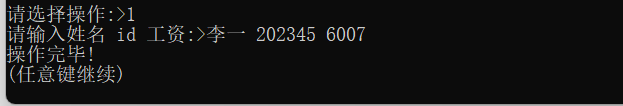
分别定义了学生类和老师类两者都继承自person类，来对应学生信息管理系统和教师信息管理系统。每个类都包含了增删查改，导出，查询等基本功能。

2.3增删查改功能

由于本项目List类用了模板，管理学生和教师信息的链表来自同一List模板，所以本部分将一起介绍其相同功能

**2.3.2新增人员信息**

该模块声明形式为void add()与void add(Node<T>\* new\_node)。本模块主要由List的add成员函数负责，该成员函数功能为从用户输入接受数据并将其按照相关数据（教师为工资，学生为分数）排名插入到链表的正确位置（由大到小）。通过重名函数重载，该成员函数有两种形式，分别为void add()与void add(Node<T>\* new\_node)。其中，void add()会在程序进入该函数时，收集用户输入的新增人员信息并建立一个Node来储存和管理这些信息，之后便利链表确保新增用户id不与已有人员id冲突。之后再次遍历链表找到第一个工资/分数比Node小大节点，将Node 插入在该节点前，最后再次便利List，重新给各Node的rangking（节点类的成员，表排名）赋值。而void add(Node<T>\* new\_node)函数与void add()只在接受数据方式有所不同，前者直接接收stack中Pop出的已有的节点，用于恢复被删除的数据。而之后的操作与void add()基本相同



**2.3.2删除人员信息**

该模块声明形式为void remove(Stack<T>\* stack)；

由于Stack对象作用域只在main中，该函数接受Stack的地址为参数，用来将删除的节点Push到Stack中，为之后恢复被删除人员信息做准备。本模块首先便利链表，找到与指定id相同的人员，之后将其从链表中移除并Push进stack中。如果在遍历完链表后还未找到指定id人员，则向用户反馈“删除失败，没有此id人员信息”，并不对原链表做任何修改。



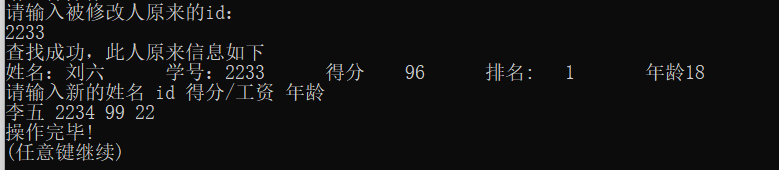
2.3.3**查询人员信息**

该模块声明为bool find()；该模块便利链表寻找与用户输入id一致的人员信息，若找到，则将其信息打印并提示操作成功，若失败，则提示用户查找失败，没有此id人员信息



2.3.4**修改人员信息**

该模块声明为void change()；模块首先与find一样，会便利链表找出被修改人员信息的Node，通过调用data类中的set函数完成对该Node信息的修改。最后，该模块还会再次便利列表确保修改后的id不会与其他又有人员id重复

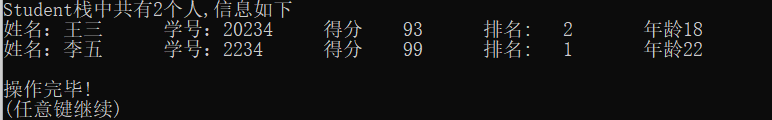


3.特殊功能

3.1撤销删除功能

**3.1.1查看被删除的人员信息**

此功能相当于回收站，stack中有所有被删除的人员的信息。本模块通过便利stack，导出stack中所有人员信息使用户得以查看所有被删除的人员信息

****

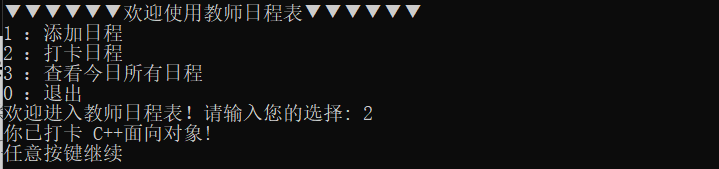
**3.1.2恢复被删除的人员信息**

本模块可恢复最近一次被删除的人员信息。首先通过stack的Pop函数得到最近一次被删除的节点，后用void add(Node<T>\* new\_node)将被删除的节点重新添加到链表中，从而实现恢复被删除的人员信息。



3.2日程表打卡功能

通过定义两个类，schedule类和scheduleItem类，并利用继承自list的queue队列来实现构造一个日程表打卡功能，包括加入日程，打卡日程，一键查看今日日程的功能，十分方便。

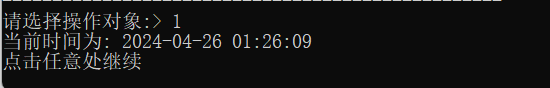


3.3日期计算功能

该小程序主要有如下三个功能

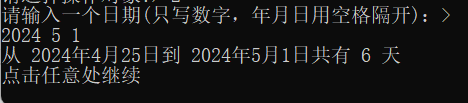
1. 查看当前日期

void date\_current()通过<ctime>获取当前时间戳转化为当地时间，从而为用户准确的当前时间



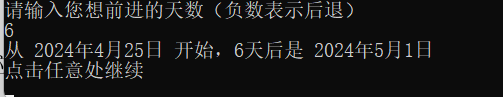
1. 计算两日期差值

通过Date类的操作符重载，计算今天到给定日期间隔天数，正数表给定日期大于今天，负数表给定日期小于今天。如从 2024年4月25日到 2024年5月1日共有 6 天



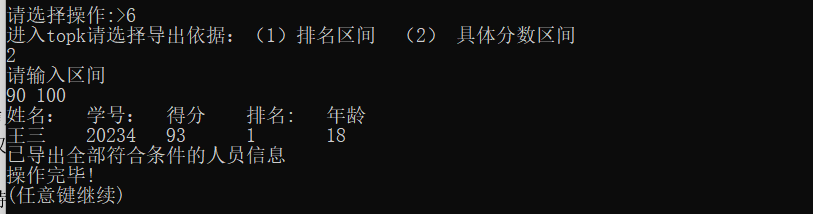
3.计算n天后日期

该模块会以今天为基准，计算n天后对应的日期，其中n可以为负数，表示倒数。如从 2024年4月25日 开始，6天后是 2024年5月1日



3.4自定义导出人员信息功能

用户既可以一键导出所有人员信息，也可以自定义导出条件，只导出符合条件的人员信息。如排名前五的学生和工资为3000-5000的教师。



结论

经过对人员信息管理系统的研究与开发，本项目成功实现了对学生和教师信息的有效管理。系统不仅具备基础的查询、统计、添加和删除功能，还创新性地引入了自定义导出人员信息，恢复删除的人员信息的功能，提高了信息处理的灵活性和便捷性。同时，系统还支持日期查询与计算，为人员信息的时空分析提供了有力支持。

在项目实施过程中，我们采用了C++中面向对象思想进行编程，使用了双向链表、栈与队列等数据结构，通过Node类、List类以及派生出的Queue、Stack和Set类，实现了对人员信息的精细化管理。这些数据结构的选择和应用，不仅满足了系统的功能需求，也提高了系统的运行效率。

此外，我们还运用了topk等算法，实现了自定义导出人员信息的功能。这一功能的实现，使得系统能够根据用户需求，快速筛选出关键信息，为决策提供有力支持。

在项目实施过程中，我们深刻体会到了理论与实践相结合的重要性。在课堂上看似简单的代码，自己动手不一定能写得出来，只有通过项目实践，才能真正提高编程水平。通过对C++面向对象应用，我们不仅解决了实际问题，也提升了自身的编程能力和问题解决能力。同时，我们也发现，在系统设计和实现过程中，需要充分考虑用户需求和系统性能，确保系统的稳定性和易用性。

下一步，我们将继续完善和优化系统，进一步拓展其功能和应用范围如增添班级数量等。我们将关注新的数据结构和算法的发展，以提高系统的运行效率和准确性。同时，我们也将多在实践中调研学生和老师需要的功能，了解他们的实际需求，为系统的持续改进提供有力支持。

总之，本项目的成功实施为我们积累了C++编程与面向对象思想应用的实践经验，也为后续学习奠定了坚实的基础。我们相信，我们的人员信息管理系统将发挥重要作用，为人们的生产生活带来更多便利。