郑 州 轻 工 业 大 学

**实 验 报 告**

**课程名称： 面向对象程序设计**

**姓 名： 原彬贺**

**学 号： 542001020223**

**专业班级： 计算机科学与技术20-02**

**任课教师： 王岩**

**2021 年 11 月 17 日**

**实验报告正文**

**实验六**

**综合设计**

**一、 实验目的**

1.清楚面向对象程序设计软件开发过程中的各个阶段；

2.学会进行实际问题的分析，抽象实际问题的功能模块；

3.能够抽象出解决实际问题的对象和类，并对其进行设计与实现；

4.能够采用 C++进行编程进行功能的实现；

5.能够对 C++程序进行调试和改进。

**二、 问题的本质和抽象描述**

**高校教师管理**

先定义一个Person基类，其含有工号，姓名，性别，工资四个保护的数据成员，以供其子类继承。再定义其两个个公有函数，一个是输入函数，将其基本信息输入。另一个保存函数，将该信息储存的result文件中。再通过友元函数重载输出运算符<<,使其可以直接将该该类直接进行输出。再分别定义teacher、staff、tempor\_worker、teacher\_staff等四个子类。设置一个menu函数给用户提示。在主函数中，通过选择是否继续，来连续输入信息。

**三、 测试**

**1.方案**

定义一个Person基类，其含有工号，姓名，性别，工资四个保护的数据成员，和两个个公有函数，一个是输入函数，一个保存函数，再定义一个友元重载输出运算符函数。再分别定义teacher、staff、tempor\_worker、teacher\_staff等四个子类。设置一个menu函数。在主函数中，通过指针来对其进行访问和存储，先通过选择职业，再定义一个相关职业的指针，通过指针访问其数据，进行输出和保存。定义一个while函数，当用户输入不为零时，继续进行工资操作。

**2.结果**

1. 开始界面图，如图1

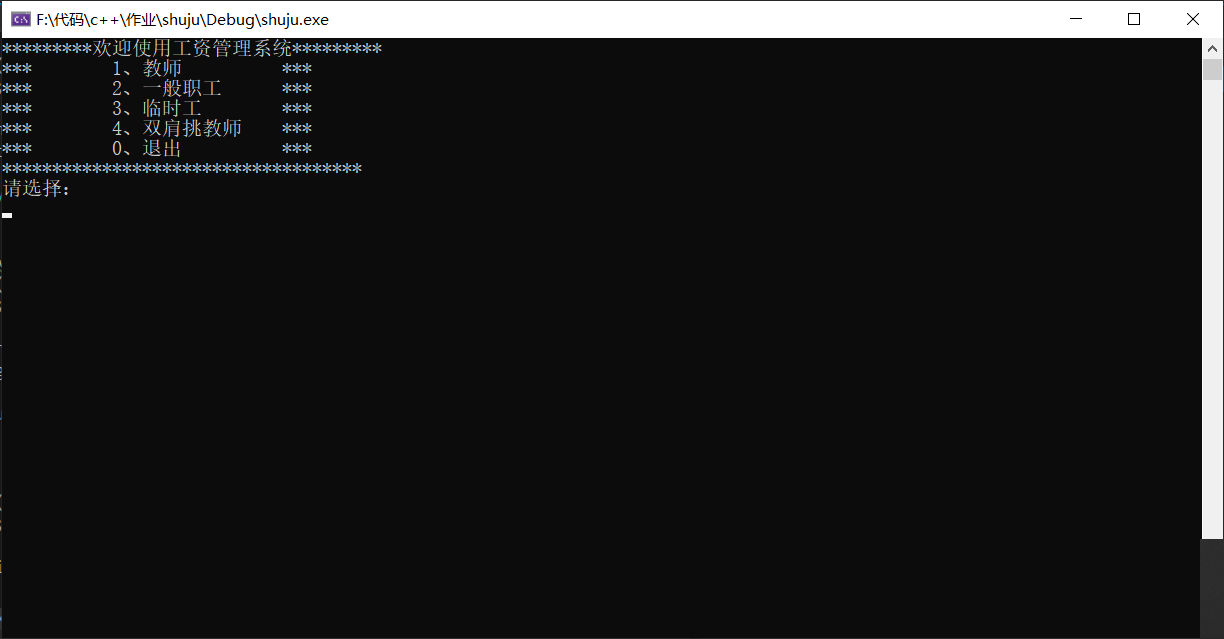
****

图1 开始界面

1. 进行第一次工资查询，通过输入1或0进行决定是否继续。如图2，图3

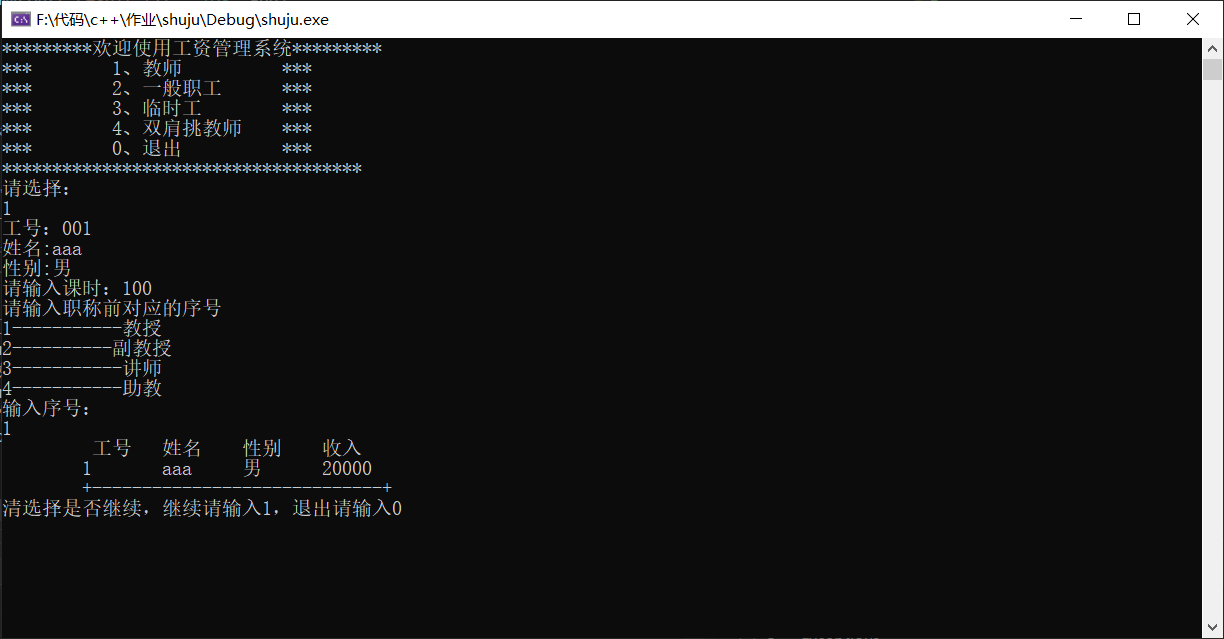


图2, 第一次输入结果

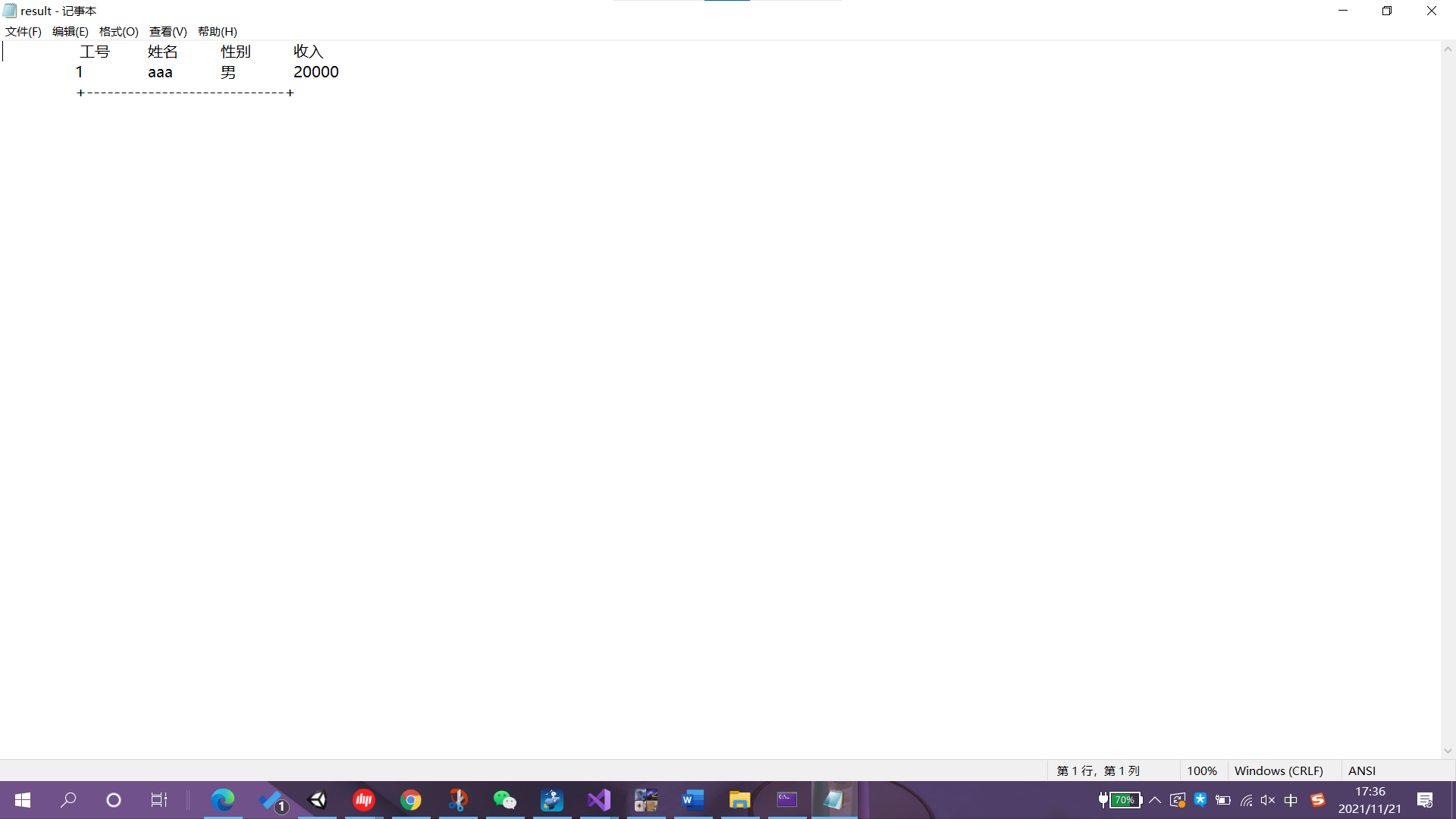


图3 result文件内容

1. 进行下一次输入，如图4，图5

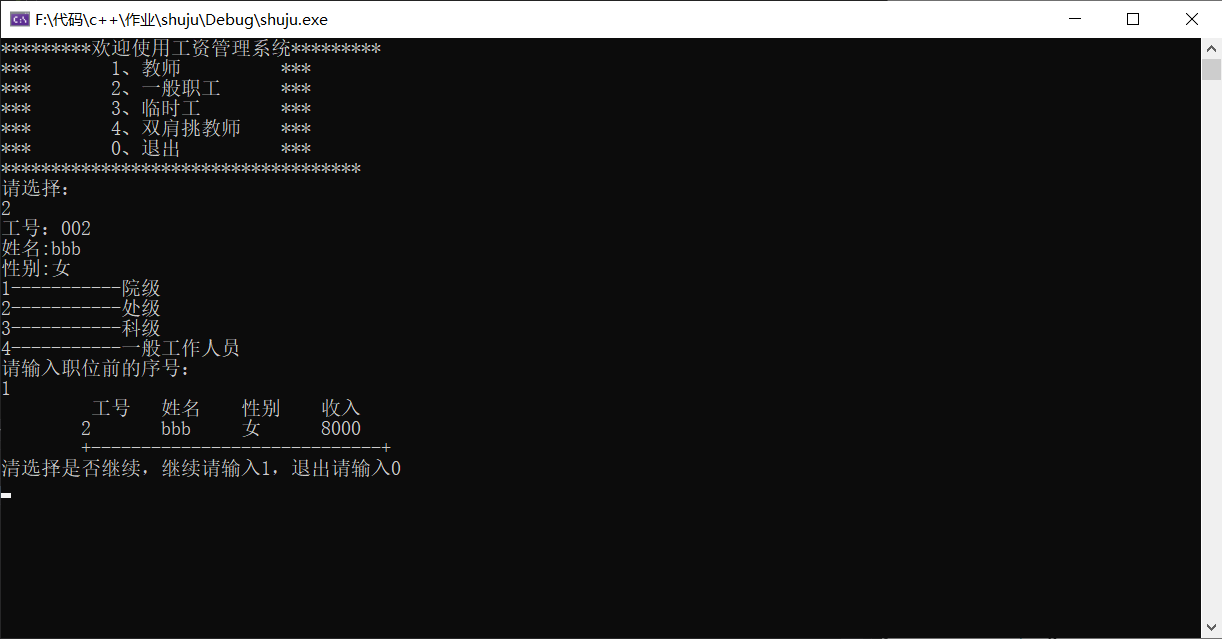


图4 下一次输入

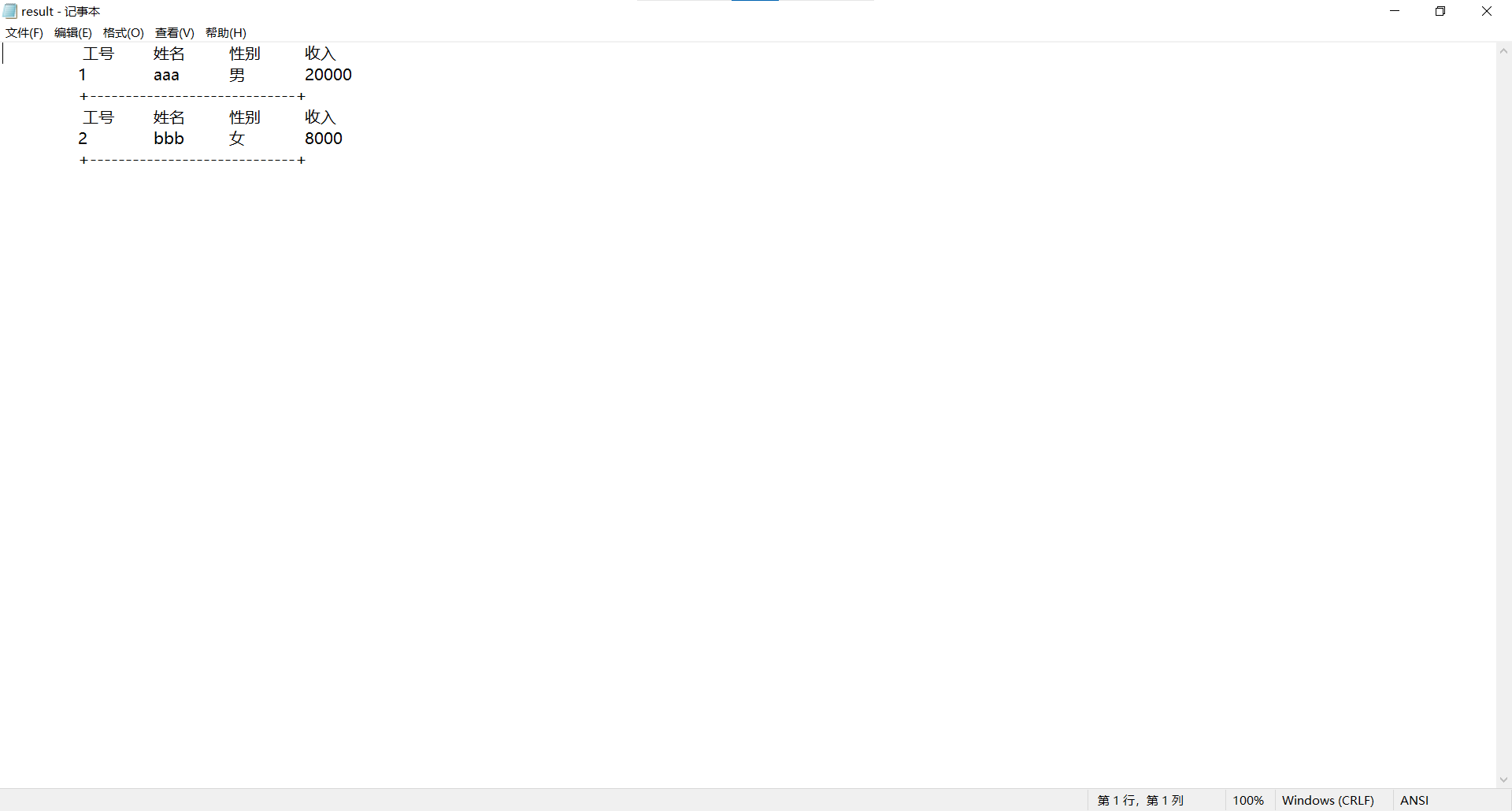


图5 进行第二次数据保存

**四、 总结与讨论**

在实验时，对于连续将数据保存到一个文件中不熟悉，通过网上查询，知道了使用ios：：app来将其连续储存到一个文件中。在实验时，学会了使用指针进行数据的存储与输出，更加安全。实验后，学会了一个人处理一些在操作时出现的问题，例如代码报错，无法正常运行，通过进行网络查询，成功的解决了这些出现的问题，学到了很多。

**附：源代码**

#include<iostream>

# #include<string>

#include<fstream>

using namespace std;

class Person

{

public:

friend ostream& operator<<(ostream&, Person&);

void input();

void save();

protected:

int number;

string name;

string sex;

double salary;

};

class Teacher :public virtual Person

{

public:

void input();

void Pay();

protected:

int teaching\_hours;

double title;

};

class Temporary\_workers :public virtual Person

{

public:

Temporary\_workers() :hourlyPay(30) {}

void input();

void Pay();

private:

double hourlyPay;

int workingHours;

};

class Staff :public virtual Person

{

public:

Staff() {}

void input();

void Pay();

protected:

double position;

};

class Staff\_Teacher :public Staff, public Teacher

{

public:

Staff\_Teacher() {}

void input();

void Pay();

};

void Person::save()

{

ofstream mycout("result.txt", ios::app);

mycout << "\t 工号\t姓名\t性别\t收入"<<endl;

mycout << "\t" << number << "\t" << name << "\t" << sex << "\t" << salary << endl;

mycout << "\t+-----------------------------+" << endl;

}

ostream& operator<<(ostream& os, Person& temp)

{

os << "\t 工号\t姓名\t性别\t收入" <<endl;

os << "\t" << temp.number << "\t" << temp.name << "\t" << temp.sex << "\t" << temp.salary << endl;

os << "\t+-----------------------------+" << endl;

return os;

}

void Person::input()

{

cout << "工号：";

cin >> number;

cout << "姓名:";

cin >> name;

cout << "性别:";

cin >> sex;

}

void Teacher::input()

{

Person::input();

cout << "请输入课时：";cin>> teaching\_hours;

cout << "请输入职称前对应的序号" << endl;

cout << "1-----------教授" << endl;

cout << "2----------副教授" << endl;

cout << "3-----------讲师" << endl;

cout << "4-----------助教" << endl;

cout << "输入序号：" << endl;

cin >> title;

}

void Teacher::Pay()

{

if (title == 1)

salary = teaching\_hours \* 150 + 5000;

else if (title == 2)

salary = teaching\_hours \* 120 + 5000;

else if (title == 3)

salary = teaching\_hours \* 100 + 5000;

else

salary = teaching\_hours \* 80 + 5000;

}

void Temporary\_workers::input()

{

cout << "当月工作时数"; cin >> workingHours;

}

void Temporary\_workers::Pay()

{

salary = workingHours \* hourlyPay;

}

void Staff::input()

{

Person::input();

cout << "1-----------院级" << endl;

cout << "2-----------处级" << endl;

cout << "3-----------科级" << endl;

cout << "4-----------一般工作人员" << endl;

cout << "请输入职位前的序号：" << endl; cin >> position;

}

void Staff::Pay()

{

if (position == 1)

salary = 8000;

else if (position == 2)

salary = 6900;

else if (position == 3)

salary = 6600;

else

salary = 5000;

}

void Staff\_Teacher::input()

{

Teacher::input();

cout << "1-----------院级" << endl;

cout << "2-----------处级" << endl;

cout << "3-----------科级" << endl;

cout << "4-----------一般工作人员" << endl;

cout << "请输入职位前的序号：" << endl; cin >> position;

}

void Staff\_Teacher::Pay()

{

Staff::Pay();

if (title == 1)

salary = teaching\_hours \* 1.5 + 5000 + salary;

else if (title == 2)

salary = teaching\_hours \* 1.2 + 4000 + salary;

else if (title == 3)

salary = teaching\_hours \* 1.2 + 3000 + salary;

else

salary = teaching\_hours \* 1.2 + 2000 + salary;

}

void menu()

{

system("cls");

cout << "\*\*\*\*\*\*\*\*\*欢迎使用工资管理系统\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\* 1、教师 \*\*\*" << endl;

cout << "\*\*\* 2、一般职工 \*\*\*" << endl;

cout << "\*\*\* 3、临时工 \*\*\*" << endl;

cout << "\*\*\* 4、双肩挑教师 \*\*\*" << endl;

cout << "\*\*\* 0、退出 \*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout << "请选择："<<endl ;

}

int main()

{

int a = 1;

while (a)

{

int i;

menu();

cin >> i;

switch (i)

{

case 1:

Teacher \* teacherptr;

teacherptr = new Teacher;

teacherptr->input();

teacherptr->Pay();

teacherptr->save();

cout << \*teacherptr;

delete teacherptr;

break;

case 2:

Staff \* staffptr;

staffptr = new Staff;

staffptr->input();

staffptr->Pay();

cout << \*staffptr;

staffptr->save();

delete staffptr;

break;

case 3:

Temporary\_workers \* temporary\_workerptr;

temporary\_workerptr = new Temporary\_workers;

temporary\_workerptr->input();

temporary\_workerptr->Pay();

temporary\_workerptr->save();

cout << \*temporary\_workerptr;

delete temporary\_workerptr;

break;

case 4:

Staff\_Teacher \* staff\_teacherptr;

staff\_teacherptr = new Staff\_Teacher;

staff\_teacherptr->input();

staff\_teacherptr->Pay();

staff\_teacherptr->save();

cout << \*staff\_teacherptr;

delete staff\_teacherptr;

break;

case 0:

cout << "已关闭系统" << endl;

return 0;

}

cout << "清选择是否继续，继续请输入1，退出请输入0" << endl;

cin >> a;

}

return 0;

}