

《面向对象程序设计实训》

课程设计报告

得分：

学期： 21-22（2）

专业： 计算机科学与技术

班级： 2020级1班

学号： 202083290400

姓名： 鲁哲豪

# 课程设计目的

课程的主要目标是：

(1) 应用开发环境进行应用程序和系统的开发设计；

(2) 掌握面向对象程序设计的基本方法和步骤；

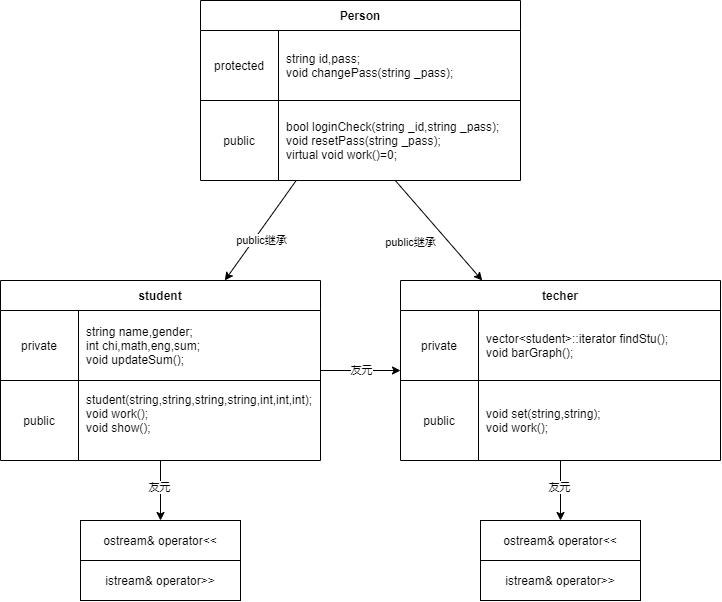
(3) 独立完成一个应用程序的开发，培养实践动手技能，提高分析问题和解决问题的能力。

# 二、开发环境

使用VSCode开发，编译器是GCC(MinGW)。源代码符合C++11标准，可移植性强。

# 三、系统设计

## 3.1模块（类之间的关系）



继承与派生：Person是student和teacher的基类。

多态关系： Person里面的work函数是纯虚函数，student和teacher给出了自己的实现。

## 3.2程序流程

### 3.2.1初始化

首次打开，会要求初始化，设定老师的账号密码。并设置开始时的学生数量与属性。

### 3.2.2登录

输入账号密码，正确则跳转到对应的欢迎界面。错误会要求重试。

登录界面只有一个，根据账号自动判断用户的身份。输入老师的账号密码会跳到老师的界面。

### 3.2.3学生

学生不能修改信息，只能修改密码、查看成绩

首先进入欢迎界面，让用户输入不同选择，进入不同选项

1.退出账号

2.修改密码

3.显示我的信息

退出账号：回到3.2.2的登陆界面

修改密码：修改自己的密码并保存

显示我的信息：显示账户信息，包括学号密码姓名性别，以及学科成绩、总分、排名。

### 3.2.4老师

首先进入欢迎界面，让用户输入不同选择，进入不同选项

1.退出账号

2.修改密码

3.增加学生信息

4.删除学生信息

5.修改学生信息

6.查找学生信息

7.显示分数分布

8.关闭系统

退出账号：回到3.2.2的登陆界面

修改密码：修改自己的密码并保存

增加学生信息：添加新学生，过程类似初始化时

以下3个操作要搜索学生。搜索有两种方式，一个按学生编号查找，一个按姓名查找，如果用户查找的内容不为空则继续，如果为空提示不存在要查找的学生

删除学生信息，可以按照学生编号或姓名进行删除学生

修改学生信息，可以按照学生编号或姓名，修改学生新的信息，并保存到文件中

1.学号

2.密码重置

3.性别

4.语文分数

5.数学分数

6.英语分数

按键选择，并输入新的数据。其中学号、性别同样会判断是否合法。

查找学生信息：显示学生信息，包括学号密码姓名性别，以及学科成绩、总分

显示分数分布：输出各科的平均成绩及及格率。画柱状图。会根据总分的最大值和最小值，自动计算每个柱对应的数据范围

关闭系统：保存并退出管理系统

# 四、系统程序源代码（每页不少于50行）。

#include<string>

#include<iostream>

#include<iomanip>

#include<fstream>

#include<vector>

#include<set>

#include<algorithm>

using namespace std;

const string g[]={ "男","女"};

void login();

set<string> ids;

void save();

void load();

class person{

protected:

string id; //学号/工号

string pass;//密码

void changePass(string \_pass){

pass=\_pass;

save();

}

public:

bool loginCheck(string \_id,string \_pass){

if(id==\_id&&pass==\_pass){

work();

return true;

}

return false;

}

void resetPass(string \_pass){

pass=\_pass;

}

virtual void work()=0;

};

class student :public person{

private:

string name,gender;

int chi,math,eng,sum;

void updateSum(){

sum=chi+math+eng;

}

public:

student(string \_id="",string \_pass="8888",string \_name="",string \_gender=g[0],

int \_chi=0,int \_math=0,int \_eng=0) :chi(\_chi),math(\_math),eng(\_eng){

updateSum();

if(ids.find(\_id)!=ids.end()){

throw "账号重复，请检查输入";

return;

}

ids.insert(id);

name=\_name;id=\_id;

pass=\_pass;

if(\_gender==g[0]||\_gender==g[1])

gender=\_gender;

else

throw "性别错误(只能输入男/女)";

}

void work(){

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

cout<<"\*|----------------|\*"<<endl;

cout<<"\*| 学生查分系统 |\*"<<endl;

cout<<"\*|----------------|\*"<<endl;

cout<<"\*| 1.退出账号 |\*"<<endl;

cout<<"\*| 2.修改密码 |\*"<<endl;

cout<<"\*| 3.显示我的信息 |\*"<<endl;

cout<<"\*|----------------|\*"<<endl;

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

while(1){

cout<<"请选择操作: ";

int choice;

cin>>choice;

string s;

switch(choice){

case 1:

login();

return;

case 2:

cout<<"新密码: ";

cin>>s;

changePass(s);

save();

break;

case 3:

show();

break;

default:

cout<<"没有此操作，请重试"<<endl;

}

}

}

void show(){

cout<<"学号: "<<id<<"\t密码: "<<pass<<"\t姓名: "<<name<<"\t性别: "<<gender

<<"\t语文: "<<chi<<"\t数学: "<<math<<"\t英语: "<<eng<<"\t总分: "<<sum<<endl;

}

friend class teacher;

friend ostream& operator<<(ostream& out,student& p);

friend istream& operator>>(istream& in,student& p);

};

ostream& operator<<(ostream& out,student& p){

out<<p.id<<'\t'<<p.pass<<'\t'<<p.name<<'\t'<<p.gender<<'\t'

<<p.chi<<'\t'<<p.math<<'\t'<<p.eng<<'\t'<<p.sum;

return out;

}

istream& operator>>(istream& in,student& p){

in>>p.id>>p.pass>>p.name>>p.gender>>p.chi>>p.math>>p.eng>>p.sum;

return in;

}

vector<student> students;

void addStudent(){

cout<<"学号 姓名 性别:"<<endl;

string a,b,c;

cin>>a>>b>>c;

cout<<"语文 数学 英语分数:"<<endl;

int e,f,g;

cin>>e>>f>>g;

students.push\_back(student(a,"8888",b,c,e,f,g));

}

class teacher :public person{

private:

vector<student>::iterator findStu(){

cout<<"要搜索学号还是姓名？（0:学号 其他:姓名）";

int op;

cin>>op;

cout<<"要搜索的内容: ";

string s;

cin>>s;

auto iter=students.begin();

for(;iter<=students.end();iter++){

string ss=op?iter->name:iter->id;

if(ss==s){

break;

}

}

return iter;

}

void barGraph(){

int n=students.size();

int\*a=new int[n],i=0;

for(student s:students)

a[i++]=s.sum;

sort(a,a+n);

float l=a[0],r=a[n-1],dl=(r-l)/10;

int per[10],x=0;

for(int i=0;i<10;i++){

cout<<setiosflags(ios::fixed)<<setprecision(1)<<l<<endl;

cout<<'\t';

int c=0;

l+=dl;

for(;a[x]<=l+0.001&&x<n;x++)

c++;

c=c\*50/n;

while(c--)cout<<'\*';

cout<<endl;

}

cout<<r<<endl;

delete[] a;

}

public:

void set(string \_id="",string \_pass="8888"){

id=\_id;

pass=\_pass;

ids.insert(id);

}

void work(){

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

cout<<"\*|----------------|\*"<<endl;

cout<<"\*| 学生管理系统 |\*"<<endl;

cout<<"\*|----------------|\*"<<endl;

cout<<"\*| 1.退出账号 |\*"<<endl;

cout<<"\*| 2.修改密码 |\*"<<endl;

cout<<"\*| 3.增加学生信息 |\*"<<endl;

cout<<"\*| 4.删除学生信息 |\*"<<endl;

cout<<"\*| 5.修改学生信息 |\*"<<endl;

cout<<"\*| 6.查找学生信息 |\*"<<endl;

cout<<"\*| 7.显示分数分布 |\*"<<endl;

cout<<"\*| 8.关闭系统 |\*"<<endl;

cout<<"\*|----------------|\*"<<endl;

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

while(1){

cout<<"请选择操作: ";

int choice;

cin>>choice;

string s;

vector<student>::iterator iter;

switch(choice){

case 1:

login();

return;

case 2:

cout<<"新密码: ";

cin>>s;

changePass(s);

break;

case 3:

addStudent();

save();

break;

case 4:

iter=findStu();

if(iter==students.end())

cout<<"没有此人"<<endl;

else

students.erase(iter);

save();

break;

case 5:

iter=findStu();

if(iter==students.end())

cout<<"没有此人"<<endl;

else{

iter->show();

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

cout<<"\*|----------------|\*"<<endl;

cout<<"\*| 修改哪项? |\*"<<endl;

cout<<"\*|----------------|\*"<<endl;

cout<<"\*| 1.学号 |\*"<<endl;

cout<<"\*| 2.密码重置 |\*"<<endl;

cout<<"\*| 3.性别 |\*"<<endl;

cout<<"\*| 4.语文分数 |\*"<<endl;

cout<<"\*| 5.数学分数 |\*"<<endl;

cout<<"\*| 6.英语分数 |\*"<<endl;

cout<<"\*|----------------|\*"<<endl;

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

student& stu=students[iter-students.begin()];

int choice;

cin>>choice;

string s;

int n;

switch(choice){

case 1:

cin>>s;

stu.id=s;

break;

case 2:

stu.pass="8888";

break;

case 3:

cin>>s;

while(s!=g[0]&&s!=g[1]){

cout<<"性别错误(只能输入男/女)"<<endl;

cin>>s;

}

stu.gender=s;

break;

case 4:

cin>>n;

stu.chi=n;

break;

case 5:

cin>>n;

stu.math=n;

break;

case 6:

cin>>n;

stu.eng=n;

break;

}

stu.updateSum();

save();

}

break;

case 6:

iter=findStu();

if(iter==students.end())

cout<<"没有此人"<<endl;

else

iter->show();

break;

case 7:

barGraph();

break;

case 8:

save();

cout<<"系统关闭" << endl;

exit(0);

default:

cout<<"没有此操作，请重试"<<endl;

}

}

}

friend ostream& operator<<(ostream& out,teacher& p);

friend istream& operator>>(istream& in,teacher& p);

};

ostream& operator<<(ostream& out,teacher& p){

out<<p.id<<'\t'<<p.pass;

return out;

}

istream& operator>>(istream& in,teacher& p){

in>>p.id>>p.pass;

return in;

}

teacher admin;

string filepath="data.txt";

void save(){

fstream file(filepath,ios::out);

if(!file.is\_open()){

cout<<"error"<<endl;

}

file<<admin<<endl;

for(auto s:students){

file<<s<<endl;

}

file.close();

cout<<"文件已保存"<<endl;

}

void load(){

fstream file(filepath,ios::in);

if(!file.is\_open()){

cout<<"error"<<endl;

}

file>>admin;

students.clear();

student s;

while(file>>s){

students.push\_back(s);

}

file.close();

cout<<"文件已读取"<<endl;

}

bool isFileExists(){

ifstream f(filepath);

return f.good();

}

void init(){

ids.clear();

cout<<"没有数据，需要初始化"<<endl;

string id,pass;

cout<<"管理员账号: ";

cin>>id;

cout<<"管理员密码: ";

cin>>pass;

admin.set(id,pass);

students.clear();

int num;

cout<<"人数"<< endl;

cin>>num;

while(num--){

try{

addStudent();

}

catch(const char\* msg){

cout<<msg<<endl;

num++;

}

}

save();

}

void login(){

string id,pass;

cout<<"账号: ";

cin>>id;

cout<<"密码(默认为8888): ";

cin>>pass;

if(admin.loginCheck(id,pass))

return;

for(student& a : students)

if(a.loginCheck(id,pass))

return;

cout<<"账号密码有误，请重试"<<endl;

login();

}

int main(){

if(isFileExists())

load();

else

init();

login();

}

# 五、系统测试

## 5.1初始化所用的测试数据

123

pass

3

123 John 男

6 9 10

11 John 男

6 9 10

12 张三 难

8 5 7

12 张三 男

8 5 7

13 李四四 女

9 4 6

## 5.2测试柱状图所用的data.txt

123 pass

11 8888 John 男 6 9 10 25

12 6666 张三 男 8 5 7 21

13 8888 李四四 女 9 4 6 20

14 8888 罗翔 男 9 8 6 23

21 8888 John 男 6 9 10 25

22 6666 张三 男 8 5 7 20

23 8888 李四四 女 9 4 6 19

24 8888 罗翔 男 9 8 6 23

31 8888 John 男 6 9 10 25

32 6666 张三 男 8 5 7 20

33 8888 李四四 女 9 4 6 19

34 8888 罗翔 男 9 8 6 23

# 六、课程设计总结（不少于500字）

这一部分很重要，要对前面做的工作进行总结，说明自己为什么完成了课程设计的目的。要将自己主要的工作总结出来，例如，自己设计了几个类，这些类之间是什么关系，如何实现了面向对象思想；又例如，完成了课程设计要求的基本功能之外，自己又添加了什么功能，这个功能对实际应用或者对面向对象程序设计的掌握有什么贡献。

本次面向对象程序设计实训课程设计，我熟练应用开发环境进行应用程序和系统的开发设计；掌握了面向对象程序设计的基本方法和步骤；独立完成了一个应用程序的开发，培养实践动手技能，提高分析问题和解决问题的能力。

《要求》中的功能，我均已实现。并独立完成了分析、开发、测试、提交的全流程。

可以对系统进行基本的增删改查功能。

通过文件输入输出，程序可以在下次运行时，获得上次的信息。

这说明了，我已经基本完成了课程设计的目的。除此之外，我还做到了：

对每个类均设计了左移、右移运算符的重载，在输入输出与文件操作时，操作方便，代码可复用。

设计了账号密码，提供了登陆界面。

对用户做出不同的选择分别做不同的处理，并设计了欢迎界面，方便用户选择。

在每个需要用户输入的地方，均设计了错误提示，避免用户错误导致的不稳定性。增加了程序鲁棒性。

使用STL中的vector、algorithm库，程序运行效率高，减少了出错的概率。

在检查输入正确性时，还用到了异常处理，这不是考试重点，所以很多同学掌握不扎实。本次课程设计增进了我对C++异常处理机制的了解。

本案例设计到了面向对象中的封装、继承、多态以及文件IO流，在设计中有person基类，以及分别派生类为老师与学生，基类中有纯虚函数，子类分别作了实现。

# 七、参考资料

[1] Aditya Bhargava．《算法图解》．北京：人民邮电出版社．2017．29-68．

[2] Runoob．《C++ 异常处理》．https://www.runoob.com/．2021年12月30日访问