|  |
| --- |
| 清华大学 电子工程系 计算机程序设计基础课程 |
| 学生成绩管理系统 |
| 小学期程序设计 实例二 |

|  |
| --- |
| 池雨泽  2013-7-7 |

# 学生成绩管理系统

索引

[学生成绩管理系统 1](#_Toc360964658)

[系统需求分析 1](#_Toc360964659)

[总体设计 2](#_Toc360964660)

[详细设计 3](#_Toc360964661)

[系统调试 5](#_Toc360964662)

[结果分析 6](#_Toc360964663)

[总结 7](#_Toc360964664)

[附录1：源程序清单 7](#_Toc360964665)

[附录2：评分表 42](#_Toc360964666)

## 系统需求分析

一个巨大的学校有数以万计的教工、学生和相应的资料需要管理。一个好的学生成绩管理系统可以协助管理员管理巨大的数据库，允许管理员、教师跟学生这三种用户登录进行相应的操作。

管理员具有管理数据库的一切权限。管理员负责录入可以使用本系统的用户名单，赋予他们使用权限，分发初始密码；可以重置用户密码，也可以修改自己的登录密码；可以以任意用户用身份登录进行管理与操作。为了保护用户隐私，不允许批量导出用户信息。对于管理员账户，提供功能如下：

1. 添加用户：可以赋予用户登录系统的权限；
2. 修改用户信息：可以修改已经加入系统的用户姓名、编号，重置密码等；
3. 删除已有用户：可以删除已经加入系统的用户资料；
4. 修改管理员登录密码：修改自己的登录密码[[1]](#footnote-1)。

教师可以录入学生的成绩，修改自己录入的成绩，删除自己录入的成绩，修改自己的登录密码等。为了保护用户隐私，不允许批量导出学生成绩信息。对于教师账户，提供功能如下：

1. 录入学生成绩：录入自己执教的科目的成绩；
2. 修改学生成绩：修改自己录入的成绩；
3. 删除学生成绩：删除自己录入的成绩；
4. 修改登录密码：修改自己的登录密码。

学生可以查询自己的成绩，修改自己的登录密码等。为了保护用户隐私，不允许查询他人的成绩。对于教师账户，提供功能如下：

1. 查询成绩：列表显示现有成绩信息，包括科目、分数、负责的教师；
2. 修改登录密码：修改自己的登录密码。

广大老师和同学可以通过这一系统安全方便地管理成绩。程序数据结构非常紧凑，可以存储海量数据，满足大学等专业院校的数据规模需求。总之，利用这一学生成绩管理系统，可以满足各种用户的需求，实现对学生成绩的管理工作。

## 总体设计

系统提供三种账户：管理员、教师和学生。

管理员具有最高权限并且是唯一的，初始密码默认为password，需要及时修改，否则整个数据库将暴露在风险之中。管理员负责赋予用户权限，分发初始密码；一旦用户忘记密码，还可以帮用户重置密码；管理员可以以任意用户用身份登录进行管理与操作，对数据库的细节进行维护。教师可以录入、修改、删除自己负责的科目成绩；学生可以查询自己的成绩。

学生成绩管理系统中功能模块图：

图 1 学生管理系统中模块之间的关系

## 详细设计

用户分为4类，其继承关系如下：

Score

Admin

增加属性：Teacher、Student

图 2 学生成绩管理系统中类的关系

学生成绩管理系统各功能模块的实现：

1. 管理员登录
2. 教师登录
3. 学生登录

## 系统调试

调试过程中最大的Bug还是出在数据IO上。由于每次读取的内容长度并不能保证固定，因此必须先读取数据长度，然后根据读取结果确定读取的内容长度。测试中发现，数据文件如果不用”r+b”模式打开读取，会在数据库体积较大（大概几百字节）导致异常。这是之前没有注意到的。

为了测试系统对海量数据的处理能力，我随机生成了15000个左右的教师和50000个左右的学生供系统读取，顺带测试数据库大小。测试结果显示，系统可以流畅地读取数万条用户记录，生成的数据库文件大小为2.5M左右。这与每个对象的数据成员大小息息相关，即每条用户信息平均占用大概32字节的空间。按此，每个成绩信息占用大概64字节，即使按照每人附带10科的成绩，总的大小也不会超过~10M的数量级。这一规模的数据大致相当于我校目前的人数规模，可以保证满足实际应用需求。

其他的问题主要有登录信息的初始化与保存。为了保证密码的安全性，输入密码的UI我采用了监视键盘输入的方式，不会将明文显示在屏幕上。但同时带来的问题就是多字节字符的读取问题：getch()无法读取多字节字符，这就意味着汉字无法输入。当然，对于密码这种东西可以将输入进行限制，剔除无效字符；但用户名就无法进行实时监视输入，导致的问题就是输入用户名的时候不能退出，必须输错然后报错退出……

## 结果分析

在源代码的组织上，仍将源代码与工程文件分开保存，这样既便于跨平台编译，又利于保证安全防止误操作。类的定义单独保存为类名为文件名的.h和.cpp中，且头文件使用编译头保护，既避免了重复编译又减少了编译的工作量。

在数据的组织上，为每个类设置IO函数，保证了数据结构的稳定读写。

在代码的细节上，根据Google C++风格指引的建议，没有使用C++风格的流IO。对于异常处理，我也根据Google的建议没有采用C++风格的try-throw-catch结构，而是对于可以预见的非致命错误进行提示后继续运行，对于不可预见的致命性错误使用printf输出后暂停程序。调试中并没有因为异常处理带来额外的麻烦，大部分的错误提示也都没有出现。

编译环境的选择。使用的是Code::Blocks+TDM64 GCC。64位编译器，但加了-m32参数，编译结果为32位程序，在Windows 8 x64上编译并测试运行通过。调用了非标准库函数，使用VS特别是老的VS编译不保证通过。

## 总结

这个系统应该说比人事管理系统更加复杂，因为有账户系统、大量的数据处理。经过这些天的Debug，在消灭代码中种种不足的同时更增加了C++编程的经验。感谢老师为大家提供了这样一个提高自己的机会！

## 附录1：源程序清单

Code::Blocks工程文件:

|  |
| --- |
| Filename 学生成绩管理系统/project.codeblocks/personnelmanaging.cbp  1 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>  2 <CodeBlocks\_project\_file>  3 <FileVersion major="1" minor="6" />  4 <Project>  5 <Option title="学生成绩管理系统" />  6 <Option pch\_mode="2" />  7 <Option compiler="gcc" />  8 <Build>  9 <Target title="Debug">  10 <Option output="bin/Debug/学生成绩管理系统" prefix\_auto="1" extension\_auto="1" />  11 <Option object\_output="obj/Debug/" />  12 <Option type="1" />  13 <Option compiler="tdm64\_gcc\_compiler" />  14 <Compiler>  15 <Add option="-Weffc++" />  16 <Add option="-pedantic" />  17 <Add option="-Wfatal-errors" />  18 <Add option="-Wextra" />  19 <Add option="-g" />  20 <Add option="-m32" />  21 </Compiler>  22 <Linker>  23 <Add option="-m32" />  24 </Linker>  25 </Target>  26 <Target title="Release">  27 <Option output="bin/Release/学生成绩管理系统" prefix\_auto="1" extension\_auto="1" />  28 <Option object\_output="obj/Release/" />  29 <Option type="1" />  30 <Option compiler="tdm64\_gcc\_compiler" />  31 <Compiler>  32 <Add option="-Os" />  33 <Add option="-O3" />  34 <Add option="-m32" />  35 </Compiler>  36 <Linker>  37 <Add option="-s" />  38 <Add option="-m32" />  39 </Linker>  40 </Target>  41 </Build>  42 <Compiler>  43 <Add option="-Wall" />  44 </Compiler>  45 <Unit filename="../codes/classes/admin.cpp" />  46 <Unit filename="../codes/classes/admin.h" />  47 <Unit filename="../codes/classes/score.cpp" />  48 <Unit filename="../codes/classes/score.h" />  49 <Unit filename="../codes/classes/student.cpp" />  50 <Unit filename="../codes/classes/student.h" />  51 <Unit filename="../codes/classes/teacher.cpp" />  52 <Unit filename="../codes/classes/teacher.h" />  53 <Unit filename="../codes/classes/user.cpp" />  54 <Unit filename="../codes/classes/user.h" />  55 <Unit filename="../codes/main.cpp" />  56 <Extensions>  57 <code\_completion />  58 <envvars />  59 <debugger />  60 <DoxyBlocks>  61 <comment\_style block="0" line="0" />  62 <doxyfile\_project />  63 <doxyfile\_build />  64 <doxyfile\_warnings />  65 <doxyfile\_output />  66 <doxyfile\_dot />  67 <general />  68 </DoxyBlocks>  69 </Extensions>  70 </Project>  71 </CodeBlocks\_project\_file> |

main.cpp:

|  |
| --- |
| Filename学生成绩管理系统/codes/main.cpp  1 #include<cerrno>  2 #include<cstdlib>  3 #include<cstdio>  4 #include<ctime>  5  6 #include<iostream>  7  8 #include<conio.h>  9 #include<windows.h>  10  11 #include"classes/admin.h"  12 #include"classes/score.h"  13 #include"classes/student.h"  14 #include"classes/teacher.h"  15 #include"classes/user.h"  16  17 using namespace std;  18  19 bool go\_on=true;  20 int main()  21 {  22 system("title 学生成绩管理系统");  23 system("cls");  24 Sleep(200);  25 for(int i=0;i<12;++i)  26 {  27 for(int j=0;j<i;++j)  28 {  29 printf(" ");  30 }  31 printf("欢迎使用学生成绩管理系统\r");  32 Sleep(100);  33 }  34 system("cls");  35  36 //读取数据结构  37 FILE\* read\_data=fopen("default.admin","r+b");  38 for(;read\_data==NULL;)  39 {  40 if(errno==ENOENT)  41 {  42 printf("没有找到数据库文件\"default.admin\"，将创建空文件。");  43 fclose(fopen("default.admin","wb"));  44 read\_data=fopen("default.admin","r+b");  45 Wait();  46 Wait();  47 putchar('\r');  48 }  49 else  50 {  51 perror("Unable to open database default.admin");  52 }  53 }  54 Admin\* admin;  55 long int init\_pos=ftell(read\_data);  56 fseek(read\_data,0,SEEK\_END); //测试文件尾位置  57 if(ftell(read\_data)==init\_pos) //文件空  58 {  59 admin=new Admin(Hash("password")); //TODO:new:admin  60 fseek(read\_data,0,SEEK\_SET); //重置文件指针  61 }  62 else  63 {  64 size\_t size=ftell(read\_data)-init\_pos;  65 char\* buffer=new char[size]; //NOTE:new:buffer  66 fseek(read\_data,0,SEEK\_SET); //重置文件指针  67 size\_t read\_size=fread(buffer,1,size,read\_data);  68 if(read\_size!=size)  69 {  70 printf("Failed to read database.\n%d bytes should be read, but only %d bytes were read.",size,read\_size);  71 Pause();  72 return -1;  73 }  74 admin=new Admin(buffer,size); //TODO:new:admin  75 delete[] buffer; //NOTE:delete:buffer  76 fseek(read\_data,0,SEEK\_SET); //重置文件指针  77 }  78  79 for(;go\_on;)  80 {  81 system("cls");  82 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎使用学生成绩管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  83 char login\_name[20]={'\0'}; //19位是uint64\_t类型的上限  84 char password\_char[25]={'\0'}; //24字节密码经过Hash变为64位整数储存  85 printf("请输入您的姓名或编号:");  86 fgets(login\_name,20,stdin);  87 for(int i=0;i<20;++i)  88 {  89 login\_name[19-i]=(login\_name[19-i]=='\n'||login\_name[19-i]=='\r')?'\0':login\_name[19-i];  90 }  91 if(login\_name[0]=='\0')  92 {  93 printf("用户名不能为空；按Q键退出，或按其他键重新输入。\n");  94 Wait();  95 char quit=getch();  96 if(quit=='Q'||quit=='q')  97 {  98 break;  99 }  100 continue;  101 }  102 printf("请输入您的密码:");  103 for(int i=0;i<24;++i)  104 {  105 char get=getch();  106 switch(get)  107 {  108 case'\r':  109 password\_char[i]='\0';  110 i=24;  111 putch('\n');  112 Wait(); //输入密码结束之后等待，防止暴力破解  113 break;  114 case 8:  115 if(i>0)  116 {  117 putch(8);  118 putch(0);  119 putch(8);  120 --i;  121 }  122 --i;  123 break;  124 default:  125 if(get>=0x20&&get<=0x7E)  126 {  127 putch('\*');  128 password\_char[i]=get;  129 }  130 else  131 {  132 --i;  133 }  134 }  135 }  136 bool number=true; //判断输入的是id还是用户名  137 if(strlen(login\_name)<MAX\_NAME\_LENGTH)  138 {  139 for(int i=0;i<20;++i)  140 {  141 if((login\_name[i]>'9'||login\_name[i]<'0')&&login\_name[i]!='\0')  142 {  143 number=false;  144 }  145 break;  146 }  147 }  148 Teacher\* teacher=NULL;  149 Student\* student=NULL;  150 if(number)  151 {  152 uint64\_t id;  153 sscanf(login\_name,"%llud",&id);  154 if(id==admin->id\_) //管理员登录  155 {  156 if(Hash(password\_char)!=admin->hash\_)  157 {  158 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  159 Wait();  160 char quit=getch();  161 if(quit=='Q'||quit=='q')  162 {  163 break;  164 }  165 continue;  166 }  167 admin->Manage();  168 continue;  169 }  170 else //其他用户登录  171 {  172 if(admin->admin\_id\_set\_.count(id)!=0) //如果已存在  173 {  174 teacher=admin->FindTeacher(id);  175 if(teacher==NULL)  176 {  177 student=admin->FindStudent(id);  178 if(student!=NULL)  179 {  180 if(Hash(password\_char)!=student->hash\_)  181 {  182 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  183 Wait();  184 char quit=getch();  185 if(quit=='Q'||quit=='q')  186 {  187 break;  188 }  189 continue;  190 }  191 student->Login(admin);  192 continue;  193 }  194 }  195 else  196 {  197 if(Hash(password\_char)!=teacher->hash\_)  198 {  199 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  200 Wait();  201 char quit=getch();  202 if(quit=='Q'||quit=='q')  203 {  204 break;  205 }  206 continue;  207 }  208 teacher->Login(admin);  209 continue;  210 }  211 }  212 }  213 }  214 if(strcmp(login\_name,"admin")==0||  215 strcmp(login\_name,"Admin")==0||  216 strcmp(login\_name,"ADMIN")==0||  217 strcmp(login\_name,"administrator")==0||  218 strcmp(login\_name,"Administrator")==0||  219 strcmp(login\_name,"ADMINISTRATOR")==0)  220 {  221 if(Hash(password\_char)!=admin->hash\_)  222 {  223 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  224 Wait();  225 char quit=getch();  226 if(quit=='Q'||quit=='q')  227 {  228 break;  229 }  230 continue;  231 }  232 admin->Manage();  233 continue;  234 }  235 teacher=admin->FindTeacher(login\_name);  236 if(teacher==NULL)  237 {  238 student=admin->FindStudent(login\_name);  239 if(student!=NULL)  240 {  241 if(Hash(password\_char)!=student->hash\_)  242 {  243 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  244 Wait();  245 char quit=getch();  246 if(quit=='Q'||quit=='q')  247 {  248 break;  249 }  250 continue;  251 }  252 student->Login(admin);  253 continue;  254 }  255 }  256 else  257 {  258 if(Hash(password\_char)!=teacher->hash\_)  259 {  260 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  261 Wait();  262 char quit=getch();  263 if(quit=='Q'||quit=='q')  264 {  265 break;  266 }  267 continue;  268 }  269 teacher->Login(admin);  270 continue;  271 }  272 printf("用户名或密码不正确；按Q键退出，或按其他键重新输入。\n");  273 Wait();  274 char quit=getch();  275 if(quit=='Q'||quit=='q')  276 {  277 break;  278 }  279 continue;  280 }  281 size\_t size=admin->WriteTo(NULL,0);  282 char\* buffer=new char[size]; //TODO:new:buffer  283 admin->WriteTo(buffer,size);  284 fwrite(buffer,1,size,read\_data);  285 fclose(read\_data);  286 printf("正在保存...\r");  287 Wait();  288 delete[] buffer; //NOTE:delete:buffer  289 printf("感谢您的使用，再见!");  290  291  292  293  294  295  296  297  298  299  300  301  302 return 0;  303 } |

user.h:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/user.h  1 #ifndef USER\_H\_  2 #define USER\_H\_  3  4 #define MAX\_NAME\_LENGTH 16  5  6 #include<cstdint>  7 #include<cstdio>  8 #include<cstdlib>  9 #include<cstring>  10 #include<ctime>  11  12 #include<set>  13  14 #include<conio.h>  15 #include<windows.h>  16  17 inline void Pause()  18 {  19 fflush(stdin);  20 fflush(stdout);  21 getch();  22 }  23  24 inline void Wait()  25 {  26 fflush(stdin);  27 fflush(stdout);  28 time\_t start,current;  29 time(&start);  30 for(time(&current);current-start<1.0;time(&current));  31 }  32  33 class User  34 {  35 public:  36 User(const char\* name,uint64\_t id,uint64\_t password);  37 virtual ~User();  38  39 bool Identify(const char\* password);  40 virtual size\_t WriteTo(char\* buffer,size\_t size)=0;  41 virtual size\_t ReadFrom(const char\* buffer,size\_t size)=0;  42  43 friend class Teacher;  44 friend class Student;  45 friend class Admin;  46 friend int main();  47  48 protected:  49 char name\_[MAX\_NAME\_LENGTH]; //不允许超长；空为出错  50 uint64\_t id\_; //0x0保留  51 uint64\_t hash\_; //散列后的密码  52 };  53  54 inline uint64\_t Hash(const char\* password\_char)  55 {  56 char password[25]={'\0'};  57 strncpy(password,password\_char,24);  58 return ((\*((uint64\_t\*)(password ))+8)>>1|((\*((uint64\_t\*)(password ))+8)<<7))+  59 ((\*((uint64\_t\*)(password+ 8))+5)>>2|((\*((uint64\_t\*)(password+ 8))+5)<<6))+  60 ((\*((uint64\_t\*)(password+16))+1)>>5|((\*((uint64\_t\*)(password+16))+1)<<3));  61 }  62  63 #endif // USER\_H\_ |

user.cpp:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/user.cpp  1 #include<cstring>  2  3 #include<set>  4  5 #include"user.h"  6  7 using namespace std;  8  9  10 User::User(const char\* name,uint64\_t id,uint64\_t password):id\_(id),hash\_(password)  11 {  12 if(name==NULL)  13 {  14 name\_[0]='\0';  15 strcpy(name\_+1,"NAME\_IS\_NULL");  16 return;  17 }  18 if(strlen(name)>=MAX\_NAME\_LENGTH)  19 {  20 name\_[0]='\0';  21 strcpy(name\_+1,"NAME\_TOO\_LONG");  22 id\_=0;  23 hash\_=0;  24 }  25 else  26 {  27 if(id==0)  28 {  29 name\_[0]='\0';  30 strcpy(name\_+1,"ID\_IS\_ZERO");  31 id\_=0;  32 hash\_=0;  33 }  34 else  35 {  36 strcpy(name\_,name);  37  38 }  39 }  40 }  41  42 User::~User()  43 {  44 }  45  46 bool User::Identify(const char\* password)  47 {  48 return hash\_==Hash(password);  49 } |

teacher.h:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/teacher.h  1 #ifndef TEACHER\_H\_  2 #define TEACHER\_H\_  3 #include"user.h"  4 class Teacher:virtual public User  5 {  6 public:  7 Teacher(const char\* name,uint64\_t id,uint64\_t password):  8 User(name,id,password){}  9 Teacher(const char\* buffer,size\_t size);  10 ~Teacher(){}  11 virtual size\_t WriteTo(char\* buffer,size\_t size);  12 virtual size\_t ReadFrom(const char\* buffer,size\_t size);  13  14 int Login(Admin\* admin);  15  16 friend class Student;  17 friend class Admin;  18 };  19 #endif // TEACHER\_H\_ |

teacher.cpp:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/teacher.cpp  1 #include<cstring>  2 #include<cstdio>  3  4 #include<windows.h>  5  6 #include"admin.h"  7 #include"teacher.h"  8  9 using namespace std;  10  11 Teacher::Teacher(const char\* buffer,size\_t size):User(NULL,0,0)  12 {  13 ReadFrom(buffer,size);  14 }  15  16 size\_t Teacher::WriteTo(char\* buffer,size\_t size)  17 {  18 if(buffer==NULL)  19 {  20 return MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2;  21 }  22 if(MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2>size)  23 {  24 return 0;  25 }  26  27 strcpy(buffer,name\_);  28 buffer+=MAX\_NAME\_LENGTH;  29  30 for(size\_t i=0;i<sizeof(uint64\_t);++i)  31 {  32 buffer[i]=((char\*)(&id\_))[i];  33 }  34 buffer+=sizeof(uint64\_t);  35  36 for(size\_t i=0;i<sizeof(uint64\_t);++i)  37 {  38 buffer[i]=((char\*)(&hash\_))[i];  39 }  40 buffer+=sizeof(uint64\_t);  41  42 return MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2;  43 }  44  45 size\_t Teacher::ReadFrom(const char\* buffer,size\_t size)  46 {  47 if(size<MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2)  48 {  49 return 0;  50 }  51  52 strncpy(name\_,buffer,MAX\_NAME\_LENGTH);  53 buffer+=MAX\_NAME\_LENGTH;  54  55 for(size\_t i=0;i<sizeof(uint64\_t);++i)  56 {  57 ((char\*)(&id\_))[i]=buffer[i];  58 }  59 buffer+=sizeof(uint64\_t);  60  61 for(size\_t i=0;i<sizeof(uint64\_t);++i)  62 {  63 ((char\*)(&hash\_))[i]=buffer[i];  64 }  65 buffer+=sizeof(uint64\_t);  66  67 return MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2;  68 }  69  70 char\* GetString(char\* str,int n,FILE\* file=stdin);  71 inline uint64\_t GetId(const char\* str="请输入编号",FILE\* file=stdin);  72  73 int Teacher::Login(Admin\* admin)  74 {  75 bool init=true;  76 for(;;)  77 {  78 system("cls");  79 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:教师\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  80 if(init)  81 {  82 printf("欢迎使用，%s老师！",name\_);  83 Wait();  84 init=false; //只显示一次欢迎信息;  85 }  86 putchar('\r');  87 printf("请选择欲使用的功能: \n 1.录入学生成绩\n 2.修改学生成绩\n 3.删除学生成绩\n 4.修改登录密码\n B.退出登录\n Q.退出程序\n");  88 char choice=getch();  89  90 switch(choice)  91 {  92 case'1': //录入学生成绩  93 {  94 system("cls");  95 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:教师\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  96 printf("请要录入的科目:");  97 char input\_subject[MAX\_SUBJECT\_NAME\_LENGTH]={'\0'};  98 if(GetString(input\_subject,MAX\_SUBJECT\_NAME\_LENGTH)==NULL)  99 {  100 printf("输入错误。");  101 Wait();  102 break;  103 }  104  105 for(;;)  106 {  107 system("cls");  108 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:教师\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  109  110 size\_t star\_num=80-10\*2-strlen(input\_subject);  111 for(size\_t i=0;i<star\_num/2;++i)  112 {  113 putchar('=');  114 }  115 printf("现在您录入的是%s的成绩",input\_subject);  116 for(size\_t i=star\_num/2;i<star\_num;++i)  117 {  118 putchar('=');  119 }  120  121 uint64\_t input\_id=GetId("请输入学生的学号:");  122 if(input\_id==0)  123 {  124 printf("输入错误。\n");  125 printf("退出录入吗？按Q/Y/Esc退出。");  126 char quit=getch();  127 if(quit=='q'||quit=='Q'||quit=='Y'||quit=='y'||quit==27)  128 {  129 break;  130 }  131 continue;  132 }  133  134 if(admin->FindStudent(input\_id)==NULL)  135 {  136 printf("学号未找到。\n");  137 printf("退出录入吗？按Q/Y/Esc退出。");  138 char quit=getch();  139 if(quit=='q'||quit=='Q'||quit=='Y'||quit=='y'||quit==27)  140 {  141 break;  142 }  143 continue;  144 }  145 else  146 {  147 Student\* student=admin->FindStudent(input\_id);  148 size\_t found=0;  149 for(vector<Score>::iterator iter=student->score\_vector\_.begin();iter!=student->score\_vector\_.end();++iter)  150 {  151 if(strcmp(input\_subject,iter->subject\_)==0)  152 {  153 found++;  154 }  155 }  156 if(found>0)  157 {  158 printf("该学生已有本课程的成绩存在。");  159 Wait();  160 break;  161 }  162 printf("请输入分数:");  163 double input\_mark=.0;  164 fflush(stdin);  165 scanf("%lf",&input\_mark);  166 if(ferror(stdin))  167 {  168 printf("输入错误。");  169 printf("退出录入吗？按Q/Y/Esc退出。");  170 char quit=getch();  171 if(quit=='q'||quit=='Q'||quit=='Y'||quit=='y'||quit==27)  172 {  173 break;  174 }  175 continue;  176 }  177 fflush(stdin);  178 student->score\_vector\_.push\_back(Score(input\_subject,input\_mark,id\_));  179 printf("保存成功。\n");  180 printf("退出录入吗？按Q/Y/Esc退出。");  181 char quit=getch();  182 if(quit=='q'||quit=='Q'||quit=='Y'||quit=='y'||quit==27)  183 {  184 break;  185 }  186 continue;  187 }  188 }  189  190  191 Wait();  192 break;  193 }  194  195 case'2': //修改学生成绩  196 {  197 uint64\_t input\_id=GetId("请输入学生的学号:");  198 if(input\_id==0)  199 {  200 printf("输入错误。");  201 Wait();  202 break;  203 }  204 else  205 {  206 if(admin->FindStudent(input\_id)==NULL)  207 {  208 printf("学号未找到。\n");  209 Wait();  210 break;  211 }  212 else  213 {  214 printf("请输入欲修改的科目:");  215 char input\_subject[MAX\_SUBJECT\_NAME\_LENGTH]={'\0'};  216 if(GetString(input\_subject,MAX\_SUBJECT\_NAME\_LENGTH)==NULL)  217 {  218 printf("输入错误。");  219 Wait();  220 break;  221 }  222 else  223 {  224 Student\* student=admin->FindStudent(input\_id);  225 size\_t found=0;  226 for(vector<Score>::iterator iter=student->score\_vector\_.begin();iter!=student->score\_vector\_.end();++iter)  227 {  228 if(strcmp(input\_subject,iter->subject\_)==0)  229 {  230 ++found;  231 printf("请输入修改后的分数:");  232 double input\_mark;  233 if(scanf("%lf",&input\_mark)==0)  234 {  235 printf("输入错误。");  236 Wait();  237 }  238 else  239 {  240 iter->mark\_=input\_mark;  241 printf("保存成功。");  242 Wait();  243 }  244  245 }  246 }  247 if(found==0)  248 {  249 printf("未找到该科目。");  250 Wait();  251 break;  252 }  253 break;  254  255 }  256 break;  257  258 }  259 break;  260  261 }  262 break;  263 }  264  265 case'3': //删除学生成绩  266 {  267 uint64\_t input\_id=GetId("请输入学生编号:");  268 if(admin->FindStudent(input\_id)==NULL)  269 {  270 printf("学号未找到。");  271 Wait();  272 break;  273 }  274 else  275 {  276 Student\* student=admin->FindStudent(input\_id);  277 vector<Score> found;  278 for(vector<Score>::iterator iter=student->score\_vector\_.begin();iter!=student->score\_vector\_.end();++iter)  279 {  280 if(iter->teacher\_==id\_)  281 {  282 found.push\_back(\*iter);  283 }  284 }  285 if(found.empty())  286 {  287 printf("该学生的记录中没有您录入的成绩。");  288 Wait();  289 break;  290 }  291 else  292 {  293 for(size\_t i=0;i<found.size();++i)  294 {  295 printf("%4u.\t科目:%s\t分数:%g\n",i+1,found[i].subject\_,found[i].mark\_);  296 }  297 size\_t sub=0;  298 printf("请选择要删去的科目:");  299 if(scanf("%u",&sub)==0||sub==0||sub>found.size())  300 {  301 printf("输入错误。");  302 Wait();  303 break;  304 }  305 else  306 {  307 for(vector<Score>::iterator iter=student->score\_vector\_.begin();iter!=student->score\_vector\_.end();++iter)  308 {  309 if(strcmp(iter->subject\_,found[sub-1].subject\_)==0)  310 {  311 student->score\_vector\_.erase(iter);  312 printf("成功删除科目 %s 。",found[sub-1].subject\_);  313 Wait();  314 Wait();  315 if(iter==student->score\_vector\_.end())  316 {  317 break;  318 }  319 }  320 }  321 break;  322 }  323 break;  324  325 }  326 break;  327  328 }  329 break;  330  331 }  332  333 case'4': //修改登录密码  334 {  335 printf("请输入原来的密码:");  336 char password\_char[25]={'\0'};  337 for(int i=0;i<24;++i)  338 {  339 char get=getch();  340 switch(get)  341 {  342 case'\r':  343 password\_char[i]='\0';  344 i=24;  345 putch('\n');  346 Wait();  347 break;  348 case 8:  349 if(i>0)  350 {  351 putch(8);  352 putch(0);  353 putch(8);  354 --i;  355 }  356 --i;  357 break;  358 default:  359 if(get>=0x20&&get<=0x7E)  360 {  361 putch('\*');  362 password\_char[i]=get;  363 }  364 else  365 {  366 --i;  367 }  368 }  369 }  370 if(Hash(password\_char)!=hash\_)  371 {  372 printf("密码错误。");  373 Wait();  374 Wait();  375 break;  376 }  377  378 printf("请输入新的密码:");  379 for(int i=0;i<24;++i)  380 {  381 char get=getch();  382 switch(get)  383 {  384 case'\r':  385 password\_char[i]='\0';  386 i=24;  387 putch('\n');  388 break;  389 case 8:  390 if(i>0)  391 {  392 putch(8);  393 putch(0);  394 putch(8);  395 --i;  396 }  397 --i;  398 break;  399 default:  400 if(get>=0x20&&get<=0x7E)  401 {  402 putch('\*');  403 password\_char[i]=get;  404 }  405 else  406 {  407 --i;  408 }  409 }  410 }  411 printf("请再次输入密码以确认:");  412 char password\_check[25]={'\0'}; //24字节密码经过Hash变为64位整数储存  413 for(int i=0;i<24;++i)  414 {  415 char get=getch();  416 switch(get)  417 {  418 case'\r':  419 password\_check[i]='\0';  420 i=24;  421 putch('\n');  422 break;  423 case 8:  424 if(i>0)  425 {  426 putch(8);  427 putch(0);  428 putch(8);  429 --i;  430 }  431 --i;  432 break;  433 default:  434 if(get>=0x20&&get<=0x7E)  435 {  436 putch('\*');  437 password\_check[i]=get;  438 }  439 else  440 {  441 --i;  442 }  443 }  444 }  445 if(strcmp(password\_char,password\_check)==0)  446 {  447 hash\_=Hash(password\_char);  448 printf("密码修改成功。");  449 Wait();  450 Wait();  451 break;  452 }  453 else  454 {  455 printf("两次输入的密码不符。");  456 Wait();  457 Wait();  458 break;  459 }  460 break;  461 }  462  463 case'B':case'b':case 27:  464 return 0;  465  466 case'Q':case'q':  467 extern bool go\_on;  468 go\_on=false;  469 return 0;  470 default:  471 printf("输入错误。");  472 Wait();  473 }  474 }  475 return 0;  476 }  477  478 inline char\* RemoveLastEndline(char\* str,int n)  479 {  480 for(int i=0;i<n;++i)  481 {  482 if(str[n-1-i]=='\n')  483 {  484 str[n-1-i]='\0';  485 break;  486 }  487 }  488 return str[0]=='\0'?NULL:str;  489 }  490  491 inline char\* GetString(char\* str,int n,FILE\* file)  492 {  493 return fgets(str,n,file)==NULL?NULL:RemoveLastEndline(str,n);  494 }  495  496 inline uint64\_t GetId(const char\* str,FILE\* file)  497 {  498 printf(str);  499 uint64\_t input\_id=0;  500 fscanf(file,"%llu",&input\_id);  501 fflush(file);  502 return input\_id;  503 } |

student.h:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/student.h  1 #ifndef STUDENT\_H\_  2 #define STUDENT\_H\_  3  4 #include<cstdlib>  5 #include<vector>  6  7 #include"score.h"  8 #include"user.h"  9  10 class Admin;  11  12 class Student:virtual public User  13 {  14 public:  15 Student(const char\* name,uint64\_t id,uint64\_t password,std::vector<Score> score\_vector):  16 User(name,id,password),score\_vector\_(score\_vector){}  17 Student(const char\* buffer,size\_t size);  18 ~Student(){}  19 virtual size\_t WriteTo(char\* buffer,size\_t size);  20 virtual size\_t ReadFrom(const char\* buffer,size\_t size);  21  22 int Login(Admin\* admin);  23  24 friend class Teacher;  25 friend class Admin;  26  27 protected:  28 std::vector<Score> score\_vector\_;  29 };  30 #endif // STUDENT\_H\_ |

student.cpp:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/student.cpp  1 #include<cstring>  2  3 #include"admin.h"  4 #include"student.h"  5  6 using namespace std;  7  8 Student::Student(const char\* buffer,size\_t size):User(NULL,0,0),score\_vector\_()  9 {  10 ReadFrom(buffer,size);  11 }  12  13 size\_t Student::WriteTo(char\* buffer,size\_t size)  14 {  15 if(buffer==NULL)  16 {  17 return MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2+sizeof(size\_t)+sizeof(Score)\*score\_vector\_.size();  18 }  19  20 if(MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2+sizeof(size\_t)+sizeof(Score)\*score\_vector\_.size()>size)  21 {  22 return 0;  23 }  24  25 strcpy(buffer,name\_);  26 buffer+=MAX\_NAME\_LENGTH;  27  28 for(size\_t i=0;i<sizeof(uint64\_t);++i)  29 {  30 buffer[i]=((char\*)(&id\_))[i];  31 }  32 buffer+=sizeof(uint64\_t);  33  34 for(size\_t i=0;i<sizeof(uint64\_t);++i)  35 {  36 buffer[i]=((char\*)(&hash\_))[i];  37 }  38 buffer+=sizeof(uint64\_t);  39  40 size\_t score\_vector\_size=score\_vector\_.size();  41 for(size\_t i=0;i<sizeof(size\_t);++i)  42 {  43 buffer[i]=((char\*)(&score\_vector\_size))[i];  44 }  45 buffer+=sizeof(size\_t);  46  47 for(vector<Score>::iterator iter=score\_vector\_.begin();iter!=score\_vector\_.end();++iter)  48 {  49 for(size\_t i=0;i<sizeof(Score);++i)  50 {  51 buffer[i]=((char\*)(&(\*iter)))[i];  52 }  53 buffer+=sizeof(Score);  54 }  55 return MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2+sizeof(size\_t)+sizeof(Score)\*score\_vector\_size;  56 }  57  58 size\_t Student::ReadFrom(const char\* buffer,size\_t size)  59 {  60 if(size<MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2+sizeof(size\_t))  61 {  62 return 0;  63 }  64  65 strncpy(name\_,buffer,MAX\_NAME\_LENGTH);  66 buffer+=MAX\_NAME\_LENGTH;  67  68 for(size\_t i=0;i<sizeof(uint64\_t);++i)  69 {  70 ((char\*)(&id\_))[i]=buffer[i];  71 }  72 buffer+=sizeof(uint64\_t);  73  74 for(size\_t i=0;i<sizeof(uint64\_t);++i)  75 {  76 ((char\*)(&hash\_))[i]=buffer[i];  77 }  78 buffer+=sizeof(uint64\_t);  79  80 size\_t score\_vector\_size;  81 for(size\_t i=0;i<sizeof(size\_t);++i)  82 {  83 ((char\*)(&score\_vector\_size))[i]=buffer[i];  84 }  85 buffer+=sizeof(size\_t);  86  87 score\_vector\_.clear();  88 for(size\_t i=0;i<score\_vector\_size;++i)  89 {  90 score\_vector\_.push\_back(Score(buffer,MAX\_SUBJECT\_NAME\_LENGTH+sizeof(double)+sizeof(uint64\_t)));  91 buffer+=sizeof(Score);  92 }  93 return MAX\_NAME\_LENGTH+sizeof(uint64\_t)\*2+sizeof(size\_t)+sizeof(Score)\*score\_vector\_size;  94 }  95  96 int Student::Login(Admin\* admin)  97 {  98 bool init=true;  99 for(;;)  100 {  101 system("cls");  102 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  103 if(init)  104 {  105 printf("欢迎使用，%s同学！",name\_);  106 Wait();  107 init=false; //只显示一次欢迎信息;  108 }  109 putchar('\r');  110 printf("请选择欲使用的功能: \n 1.查询成绩\n 2.修改登录密码\n B.退出登录\n Q.退出程序\n");  111 char choice=getch();  112  113 switch(choice)  114 {  115 case'1': //查询成绩  116 {  117 system("cls");  118 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  119 for(vector<Score>::iterator iter=score\_vector\_.begin();iter!=score\_vector\_.end();++iter)  120 {  121 printf(" 科目：%s\t分数：%g\t负责教师：%s\n",iter->subject\_,iter->mark\_,(admin->FindTeacher(iter->teacher\_)==NULL)?"\0x8\0x8\0x8\0x8\0x8\0x8\0x8\0x8":admin->FindTeacher(iter->teacher\_)->name\_);  122 }  123  124  125 Pause();  126 break;  127 }  128  129 case'2': //修改登录密码  130 {  131 printf("请输入原来的密码:");  132 char password\_char[25]={'\0'};  133 for(int i=0;i<24;++i)  134 {  135 char get=getch();  136 switch(get)  137 {  138 case'\r':  139 password\_char[i]='\0';  140 i=24;  141 putch('\n');  142 Wait();  143 break;  144 case 8:  145 if(i>0)  146 {  147 putch(8);  148 putch(0);  149 putch(8);  150 --i;  151 }  152 --i;  153 break;  154 default:  155 if(get>=0x20&&get<=0x7E)  156 {  157 putch('\*');  158 password\_char[i]=get;  159 }  160 else  161 {  162 --i;  163 }  164 }  165 }  166 if(Hash(password\_char)!=hash\_)  167 {  168 printf("密码错误。");  169 Wait();  170 Wait();  171 break;  172 }  173  174 printf("请输入新的密码:");  175 for(int i=0;i<24;++i)  176 {  177 char get=getch();  178 switch(get)  179 {  180 case'\r':  181 password\_char[i]='\0';  182 i=24;  183 putch('\n');  184 break;  185 case 8:  186 if(i>0)  187 {  188 putch(8);  189 putch(0);  190 putch(8);  191 --i;  192 }  193 --i;  194 break;  195 default:  196 if(get>=0x20&&get<=0x7E)  197 {  198 putch('\*');  199 password\_char[i]=get;  200 }  201 else  202 {  203 --i;  204 }  205 }  206 }  207 printf("请再次输入密码以确认:");  208 char password\_check[25]={'\0'}; //24字节密码经过Hash变为64位整数储存  209 for(int i=0;i<24;++i)  210 {  211 char get=getch();  212 switch(get)  213 {  214 case'\r':  215 password\_check[i]='\0';  216 i=24;  217 putch('\n');  218 break;  219 case 8:  220 if(i>0)  221 {  222 putch(8);  223 putch(0);  224 putch(8);  225 --i;  226 }  227 --i;  228 break;  229 default:  230 if(get>=0x20&&get<=0x7E)  231 {  232 putch('\*');  233 password\_check[i]=get;  234 }  235 else  236 {  237 --i;  238 }  239 }  240 }  241 if(strcmp(password\_char,password\_check)==0)  242 {  243 hash\_=Hash(password\_char);  244 printf("密码修改成功。");  245 Wait();  246 Wait();  247 break;  248 }  249 else  250 {  251 printf("两次输入的密码不符。");  252 Wait();  253 Wait();  254 break;  255 }  256 break;  257 }  258  259 case'B':case'b':case 27:  260 return 0;  261  262 case'Q':case'q':  263 extern bool go\_on;  264 go\_on=false;  265 return 0;  266 default:  267 printf("输入错误。");  268 Wait();  269 }  270 }  271 return 0;  272 } |

admin.h:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/admin.h  1 #ifndef ADMIN\_H\_  2 #define ADMIN\_H\_  3 #include"score.h"  4 #include"student.h"  5 #include"teacher.h"  6 #include"user.h"  7 class Admin:virtual public Student,virtual public Teacher  8 {  9 public:  10 Admin(uint64\_t password):  11 User(NULL,0,password),Student(NULL,0,password,std::vector<Score>()),Teacher(NULL,0,password),teacher\_vector\_(),student\_vector\_(),admin\_id\_set\_(){}  12 Admin(const char\* buffer,size\_t size);  13 ~Admin(){};  14 virtual size\_t WriteTo(char\* buffer,size\_t size);  15 virtual size\_t ReadFrom(const char\* buffer,size\_t size);  16  17 int AddTeacher(const Teacher& add);  18 int AddStudent(const Student& add);  19 int UpdateTeacher(const Teacher& add);  20 int UpdateStudent(const Student& add);  21 int DeleteTeacher(const Teacher& del){return DeleteTeacher(del.id\_);}  22 int DeleteStudent(const Student& del){return DeleteStudent(del.id\_);}  23 int DeleteTeacher(uint64\_t);  24 int DeleteStudent(uint64\_t);  25  26 int Manage();  27 Teacher\* FindTeacher(char name[MAX\_NAME\_LENGTH]);  28 Teacher\* FindTeacher(uint64\_t id);  29 Student\* FindStudent(char name[MAX\_NAME\_LENGTH]);  30 Student\* FindStudent(uint64\_t id);  31  32 std::vector<Teacher> teacher\_vector\_;  33 std::vector<Student> student\_vector\_;  34 std::set<uint64\_t> admin\_id\_set\_;  35 };  36 #endif // ADMIN\_H\_ |

admin.cpp:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/admin.cpp  1 #include<cerrno>  2 #include<cstdio>  3 #include<cstdlib>  4 #include<cstring>  5 #include<ctime>  6  7 #include"admin.h"  8 #include"score.h"  9 #include"student.h"  10 #include"teacher.h"  11 #include"user.h"  12  13 using namespace std;  14  15 inline uint64\_t GenerateRandomPassword(char pwd[21])  16 {  17 for(int i=0;i<20;++i)  18 {  19 pwd[i]=char(double(rand())/RAND\_MAX\*(0x7F-0x21))+0x21;  20 }  21 pwd[20]='\0';  22 return Hash(pwd);  23 }  24  25 Admin::Admin(const char\* buffer,size\_t size):  26 User(NULL,0,0),Student(NULL,0,0,vector<Score>()),Teacher(NULL,0,0),teacher\_vector\_(),student\_vector\_(),admin\_id\_set\_()  27 {  28 if(ReadFrom(buffer,size)==0)  29 {  30 errno=EIO;  31 perror("Initialization error");  32 Pause();  33 }  34 }  35  36 size\_t Admin::WriteTo(char\* buffer,size\_t size)  37 {  38 char\* buffer\_init=buffer;  39 if(buffer==NULL)  40 {  41 buffer+=sizeof(uint64\_t);  42 buffer+=sizeof(vector<Teacher>::size\_type);  43  44 for(vector<Teacher>::iterator iter=teacher\_vector\_.begin();iter!=teacher\_vector\_.end();++iter)  45 {  46  47 buffer+=sizeof(size\_t);  48 buffer+=iter->WriteTo(NULL,0);  49 }  50  51 buffer+=sizeof(vector<Student>::size\_type);  52  53 for(vector<Student>::iterator iter=student\_vector\_.begin();iter!=student\_vector\_.end();++iter)  54 {  55 buffer+=sizeof(size\_t);  56 buffer+=iter->WriteTo(NULL,0);  57 }  58 }  59 else  60 {  61 if(sizeof(uint64\_t)+sizeof(vector<Teacher>::size\_type)+teacher\_vector\_.size()\*(sizeof(size\_t))+student\_vector\_.size()\*(sizeof(size\_t))>size)  62 {  63 return 0;  64 }  65 for(size\_t i=0;i<sizeof(uint64\_t);++i)  66 {  67 buffer[i]=((char\*)(&hash\_))[i];  68 }  69 buffer+=sizeof(uint64\_t);  70  71 vector<Teacher>::size\_type teacher\_vector\_size=teacher\_vector\_.size();  72 for(size\_t i=0;i<sizeof(teacher\_vector\_size);++i)  73 {  74 buffer[i]=((char\*)(&teacher\_vector\_size))[i];  75 }  76 buffer+=sizeof(teacher\_vector\_size);  77  78 for(vector<Teacher>::iterator iter=teacher\_vector\_.begin();iter!=teacher\_vector\_.end();++iter)  79 {  80 size\_t write\_size=iter->WriteTo(NULL,0);  81 for(size\_t i=0;i<sizeof(write\_size);++i)  82 {  83 buffer[i]=((char\*)(&write\_size))[i];  84 }  85 buffer+=sizeof(write\_size);  86  87 buffer+=iter->WriteTo(buffer,iter->WriteTo(NULL,0));  88 }  89  90 vector<Student>::size\_type student\_vector\_size=student\_vector\_.size();  91 for(size\_t i=0;i<sizeof(student\_vector\_size);++i)  92 {  93 buffer[i]=((char\*)(&student\_vector\_size))[i];  94 }  95 buffer+=sizeof(student\_vector\_size);  96  97 for(vector<Student>::iterator iter=student\_vector\_.begin();iter!=student\_vector\_.end();++iter)  98 {  99 size\_t write\_size=iter->WriteTo(NULL,0);  100 for(size\_t i=0;i<sizeof(write\_size);++i)  101 {  102 buffer[i]=((char\*)(&write\_size))[i];  103 }  104 buffer+=sizeof(write\_size);  105  106 buffer+=iter->WriteTo(buffer,iter->WriteTo(NULL,0));  107 }  108 }  109 return buffer-buffer\_init;  110 }  111  112 size\_t Admin::ReadFrom(const char\* buffer,size\_t size)  113 {  114 const char\* buffer\_init=buffer;  115 if(size<sizeof(uint64\_t)+sizeof(vector<Teacher>::size\_type)+sizeof(vector<Student>::size\_type))  116 {  117 return 0;  118 }  119  120 for(size\_t i=0;i<sizeof(uint64\_t);++i)  121 {  122 ((char\*)(&hash\_))[i]=buffer[i];  123 }  124 buffer+=sizeof(uint64\_t);  125  126 vector<Teacher>::size\_type teacher\_vector\_size;  127 for(size\_t i=0;i<sizeof(teacher\_vector\_size);++i)  128 {  129 ((char\*)(&teacher\_vector\_size))[i]=buffer[i];  130 }  131 buffer+=sizeof(teacher\_vector\_size);  132  133 teacher\_vector\_.clear();  134 for(size\_t iter=0;iter<teacher\_vector\_size;++iter)  135 {  136 size\_t read\_size=0;  137 for(size\_t i=0;i<sizeof(read\_size);++i)  138 {  139 ((char\*)(&read\_size))[i]=buffer[i];  140 }  141 buffer+=sizeof(read\_size);  142  143 AddTeacher(Teacher(buffer,read\_size));  144 buffer+=read\_size;  145 }  146  147 vector<Student>::size\_type student\_vector\_size;  148 for(size\_t i=0;i<sizeof(student\_vector\_size);++i)  149 {  150 ((char\*)(&student\_vector\_size))[i]=buffer[i];  151 }  152 buffer+=sizeof(student\_vector\_size);  153  154 student\_vector\_.clear();  155 for(size\_t iter=0;iter<student\_vector\_size;++iter)  156 {  157 size\_t read\_size;  158 for(size\_t i=0;i<sizeof(read\_size);++i)  159 {  160 ((char\*)(&read\_size))[i]=buffer[i];  161 }  162 buffer+=sizeof(read\_size);  163  164 AddStudent(Student(buffer,read\_size));  165 buffer+=read\_size;  166 }  167  168 return buffer-buffer\_init;  169 }  170  171 int Admin::AddTeacher(const Teacher& add)  172 {  173 if(admin\_id\_set\_.count(add.id\_)!=0)  174 {  175 return 0;  176 }  177 teacher\_vector\_.push\_back(add);  178 admin\_id\_set\_.insert(add.id\_);  179 return 1;  180 }  181  182 int Admin::AddStudent(const Student& add)  183 {  184 if(admin\_id\_set\_.count(add.id\_)!=0)  185 {  186 return 0;  187 }  188 student\_vector\_.push\_back(add);  189 admin\_id\_set\_.insert(add.id\_);  190 return 1;  191 }  192  193 int Admin::UpdateTeacher(const Teacher& update)  194 {  195 if(admin\_id\_set\_.count(update.id\_)==0)  196 {  197 return 0;  198 }  199 for(vector<Teacher>::iterator iter=teacher\_vector\_.begin();iter!=teacher\_vector\_.end();++iter)  200 {  201 if(iter->id\_==update.id\_)  202 {  203 \*iter=update;  204 return 1;  205 }  206 }  207 return 0;  208 }  209  210 int Admin::UpdateStudent(const Student& update)  211 {  212 if(admin\_id\_set\_.count(update.id\_)==0)  213 {  214 return 0;  215 }  216 for(vector<Student>::iterator iter=student\_vector\_.begin();iter!=student\_vector\_.end();++iter)  217 {  218 if(iter->id\_==update.id\_)  219 {  220 \*iter=update;  221 return 1;  222 }  223 }  224 return 0;  225 }  226  227 int Admin::DeleteTeacher(uint64\_t del)  228 {  229 if(admin\_id\_set\_.count(del)==0)  230 {  231 return 0;  232 }  233 for(vector<Teacher>::iterator iter=teacher\_vector\_.begin();iter!=teacher\_vector\_.end();++iter)  234 {  235 if(iter->id\_==del)  236 {  237 teacher\_vector\_.erase(iter);  238 return 1;  239 }  240 }  241 return 0;  242 }  243  244 int Admin::DeleteStudent(uint64\_t del)  245 {  246 if(admin\_id\_set\_.count(del)==0)  247 {  248 return 0;  249 }  250 for(vector<Student>::iterator iter=student\_vector\_.begin();iter!=student\_vector\_.end();++iter)  251 {  252 if(iter->id\_==del)  253 {  254 student\_vector\_.erase(iter);  255 return 1;  256 }  257 }  258 return 0;  259 }  260  261 int Admin::Manage()  262 {  263 bool init=true;  264 for(;;)  265 {  266 system("cls");  267 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:管理员\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  268 if(init)  269 {  270 printf("欢迎您的使用！");  271 Wait();  272 init=false; //只显示一次欢迎信息;  273 }  274 putchar('\r');  275 printf("请选择欲使用的功能: \n 1.添加用户\n 2.修改用户信息\n 3.删除已有用户\n 4.修改管理员登录密码\n B.退出登录\n Q.退出程序\n");  276 char choice=getch();  277  278 switch(choice)  279 {  280 case'1': //添加用户  281  282 for(bool stop=false;!stop;)  283 {  284 system("cls");  285 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:管理员\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  286 printf("请选择输入的设备:\n 1.从格式化文件读取\n 2.从键盘输入\n B.返回上一层菜单\n Q.退出程序\n");  287 choice=getch();  288 FILE\* input\_file=NULL,\*output\_file=NULL;  289 char input\_line[MAX\_NAME\_LENGTH+20+20]; //多预留20个字符空间  290 char t\_or\_s='\0';  291  292 char input\_name[MAX\_NAME\_LENGTH];  293 uint64\_t input\_id;  294 char random\_password[21];  295 int counter=0;  296 time\_t time\_ptr;  297  298 switch(choice)  299 {  300 case'1':  301 printf("T.老师\tS.学生");  302 t\_or\_s=getch();  303 putchar('\r');  304 if(t\_or\_s!='T'&&t\_or\_s!='t'&&t\_or\_s!='S'&&t\_or\_s!='s')  305 {  306 printf("输入错误。 \n");  307 Wait();  308 break;  309 }  310 printf("请输入文件的完整路径:");  311 char path[MAX\_PATH+1];  312 gets(path);  313 input\_file=fopen(path,"r");  314 if(input\_file==NULL)  315 {  316 if(errno==ENOENT)  317 {  318 printf("文件不存在!\n");  319 Wait();  320 break;  321 }  322 else  323 {  324 perror("Failed to open the file");  325 Wait();  326 Wait();  327 break;  328 }  329 }  330 else  331 {  332 output\_file=fopen("initial\_password.of.txt","a+");  333 if(output\_file==NULL)  334 {  335 perror("Failed to open output file \"initial\_password.of.txt\"");  336 Wait();  337 break;  338 }  339 srand(time(&time\_ptr));  340 if(t\_or\_s=='t'||t\_or\_s=='T')  341 {  342 counter=0;  343 for(;fgets(input\_line,MAX\_NAME\_LENGTH+20+20,input\_file);)  344 {  345 sscanf(input\_line,"%s%llu",input\_name,&input\_id);  346 //初始密码随机产生  347 if(AddTeacher(Teacher(input\_name,input\_id,GenerateRandomPassword(random\_password))))  348 {  349 //保存至initial\_password.of.txt  350 fprintf(output\_file,"%s\t%llu\t%s\n",input\_name,input\_id,random\_password);  351 ++counter;  352 }  353 }  354 if(counter>0)  355 {  356  357 printf("成功读取%d条用户信息；已将初始密码保存至\"initial\_password.of.txt\"\n",counter);  358 }  359 else  360 {  361 printf("没有读取任何用户信息。");  362 }  363 Wait();  364 Wait();  365 }  366 if(t\_or\_s=='s'||t\_or\_s=='S')  367 {  368 counter=0;  369 for(;fgets(input\_line,MAX\_NAME\_LENGTH+20+20,input\_file);)  370 {  371 sscanf(input\_line,"%s%llu",input\_name,&input\_id);  372 //初始密码随机产生  373 if(AddStudent(Student(input\_name,input\_id,GenerateRandomPassword(random\_password),vector<Score>())))  374 {  375 //保存至initial\_password.of.txt  376 fprintf(output\_file,"%s\t%llu\t%s\n",input\_name,input\_id,random\_password);  377 ++counter;  378 }  379 }  380 if(counter>0)  381 {  382  383 printf("成功读取%d条用户信息；已将初始密码保存至\"initial\_password.of.txt\"\n",counter);  384 }  385 else  386 {  387 printf("没有读取任何用户信息。");  388 }  389 Wait();  390 Wait();  391 }  392 if(counter>0)  393 {  394 fprintf(output\_file,"Generated @ %s\n",ctime(&time\_ptr));  395 }  396 fclose(input\_file);  397 fclose(output\_file);  398  399 }  400 break;  401 case'2':  402 printf("T.老师\tS.学生");  403 t\_or\_s=getch();  404 putchar('\r');  405 if(t\_or\_s!='T'&&t\_or\_s!='t'&&t\_or\_s!='S'&&t\_or\_s!='s')  406 {  407 printf("输入错误。 \n");  408 Wait();  409 break;  410 }  411  412 printf("请输入姓名: ");  413 putchar(8);  414 putchar(8);  415 putchar(8);  416 if(fgets(input\_name,MAX\_NAME\_LENGTH,stdin)==NULL)  417 {  418 perror("Failed to read the name");  419 Pause();  420 break;  421 }  422  423 for(int i=0;i<MAX\_NAME\_LENGTH;++i)  424 {  425 if(input\_name[MAX\_NAME\_LENGTH-1-i]=='\n')  426 {  427 input\_name[MAX\_NAME\_LENGTH-1-i]='\0';  428 break;  429 }  430 }  431  432 printf("请输入编号:");  433 if(scanf("%llu",&input\_id)==0)  434 {  435 printf("输入错误。");  436 Wait();  437 break;  438 }  439 if(ferror(stdin))  440 {  441 perror("Failed to read the id");  442 Pause();  443 break;  444 }  445  446 if(admin\_id\_set\_.count(input\_id)!=0)  447 {  448 printf("编号重复。");  449 Wait();  450 Wait();  451 break;  452 }  453  454 printf("请输入密码，直接回车则使用随机初始化密码:");  455 fflush(stdin);  456 if(fgets(random\_password,21,stdin)==NULL)  457 {  458 perror("Unexpected input error");  459 Pause();  460 break;  461 }  462 if(random\_password[0]=='\n')  463 {  464 GenerateRandomPassword(random\_password);  465 }  466 else  467 {  468 for(int i=0;i<21;++i)  469 {  470 if(random\_password[21-1-i]=='\n')  471 {  472 random\_password[21-1-i]='\0';  473 break;  474 }  475 }  476  477 }  478  479 output\_file=fopen("initial\_password.of.txt","a+");  480 time(&time\_ptr);  481 switch(t\_or\_s)  482 {  483 case't':case'T':  484 if(AddTeacher(Teacher(input\_name,input\_id,GenerateRandomPassword(random\_password))))  485 {  486 printf("已保存。\n");  487 //保存至initial\_password.of.txt  488 fprintf(output\_file,"%s\t%llu\t%s\n",input\_name,input\_id,random\_password);  489 fprintf(output\_file,"Generated @ %s\n",ctime(&time\_ptr));  490 Wait();  491 }  492 break;  493 case's':case'S':  494 if(AddStudent(Student(input\_name,input\_id,GenerateRandomPassword(random\_password),vector<Score>())))  495 {  496 printf("已保存。\n");  497 //保存至initial\_password.of.txt  498 fprintf(output\_file,"%s\t%llu\t%s\n",input\_name,input\_id,random\_password);  499 fprintf(output\_file,"Generated @ %s\n",ctime(&time\_ptr));  500 Wait();  501 }  502 break;  503 default:  504 perror("Unexpected result");  505 }  506 fclose(output\_file);  507  508 break;  509  510 case'B':case'b':case 27:  511 stop=true;  512 break;  513 case'Q':case'q':  514 extern bool go\_on;  515 go\_on=false;  516 return 0;  517 }  518 }  519 break;  520 case'2': //修改用户信息  521 for(bool stop=false;!stop;)  522 {  523 system("cls");  524 printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*您的身份是:管理员\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  525 printf("请选择要修改的项目:\n 1.修改用户姓名\n 2.修改用户编号\n 3.重置用户密码\n 4.以用户身份登录\n B.返回上一层菜单\n Q.退出程序\n");  526  527 char choice=getch();  528 uint64\_t input\_id=0,new\_id=0;  529 char input\_name[MAX\_NAME\_LENGTH];  530 char random\_password[21];  531 FILE\* output\_file=NULL;  532 time\_t time\_ptr;  533  534  535 switch(choice)  536 {  537 case'1': //修改用户姓名  538 printf("请输入用户编号:");  539 if(scanf("%llu",&input\_id)==0)  540 {  541 printf("输入错误。");  542 Wait();  543 break;  544 }  545 if(ferror(stdin))  546 {  547 perror("Failed to read the id");  548 Pause();  549 break;  550 }  551  552 if(admin\_id\_set\_.count(input\_id)==0)  553 {  554 printf("编号不存在。");  555 Wait();  556 Wait();  557 break;  558 }  559 else  560 {  561 printf(" 请注意，用户名重复将可能导致无法使用姓名登录系统，届时只能使用编号登录。\n请输入新的用户名:");  562 fflush(stdin);  563 if(fgets(input\_name,MAX\_NAME\_LENGTH,stdin)==NULL)  564 {  565 perror("Failed to read the name");  566 Pause();  567 break;  568 }  569  570 for(int i=0;i<MAX\_NAME\_LENGTH;++i) //fgets会将换行读进去  571 {  572 if(input\_name[MAX\_NAME\_LENGTH-1-i]=='\n')  573 {  574 input\_name[MAX\_NAME\_LENGTH-1-i]='\0';  575 break;  576 }  577 }  578  579 if(FindTeacher(input\_id)!=NULL)  580 {  581 strcpy(FindTeacher(input\_id)->name\_,input\_name);  582 printf("修改成功。");  583 Wait();  584 break;  585 }  586 if(FindStudent(input\_id)!=NULL)  587 {  588 strcpy(FindStudent(input\_id)->name\_,input\_name);  589 printf("修改成功。");  590 Wait();  591 break;  592 }  593 break;  594 }  595 break;  596 case'2': //修改用户编号  597 printf("请输入用户原有的编号:");  598 if(scanf("%llu",&input\_id)==0)  599 {  600 printf("输入错误。");  601 Wait();  602 break;  603 }  604 if(ferror(stdin))  605 {  606 perror("Failed to read the id");  607 Pause();  608 break;  609 }  610  611 if(admin\_id\_set\_.count(input\_id)==0)  612 {  613 printf("编号不存在。");  614 Wait();  615 Wait();  616 break;  617 }  618 else  619 {  620 printf(" 请注意，编号不可以重复。\n请输入新的用户编号:");  621 fflush(stdin);  622 if(scanf("%llu",&new\_id)==0)  623 {  624 printf("输入错误。");  625 Wait();  626 break;  627 }  628 if(ferror(stdin))  629 {  630 perror("Failed to read the id");  631 Pause();  632 break;  633 }  634  635 if(admin\_id\_set\_.count(new\_id)!=0)  636 {  637 printf("编号重复。");  638 Wait();  639 Wait();  640 break;  641 }  642 else  643 {  644  645 if(FindTeacher(input\_id)!=NULL)  646 {  647 Teacher\* teacher\_ptr=FindTeacher(input\_id);  648 admin\_id\_set\_.erase(teacher\_ptr->id\_);  649 teacher\_ptr->id\_=new\_id;  650 admin\_id\_set\_.insert(teacher\_ptr->id\_);  651 printf("修改成功。");  652 Wait();  653 break;  654 }  655 if(FindStudent(input\_id)!=NULL)  656 {  657 Student\* student\_ptr=FindStudent(input\_id);  658 admin\_id\_set\_.erase(student\_ptr->id\_);  659 student\_ptr->id\_=new\_id;  660 admin\_id\_set\_.insert(student\_ptr->id\_);  661 printf("修改成功。");  662 Wait();  663 break;  664 }  665 break;  666 }  667 break;  668 }  669 break;  670  671 case'3': //重置用户密码  672 printf("请输入用户编号:");  673 if(scanf("%llu",&input\_id)==0)  674 {  675 printf("输入错误。");  676 Wait();  677 break;  678 }  679 if(ferror(stdin))  680 {  681 perror("Failed to read the id");  682 Pause();  683 break;  684 }  685  686 if(admin\_id\_set\_.count(input\_id)==0)  687 {  688 printf("编号不存在。");  689 Wait();  690 Wait();  691 break;  692 }  693 else  694 {  695 printf("请输入密码，直接回车则使用随机初始化密码:");  696 fflush(stdin);  697 if(fgets(random\_password,21,stdin)==NULL)  698 {  699 perror("Unexpected input error");  700 Pause();  701 break;  702 }  703 if(random\_password[0]=='\n')  704 {  705 GenerateRandomPassword(random\_password);  706 }  707 else  708 {  709 for(int i=0;i<21;++i)  710 {  711 if(random\_password[21-1-i]=='\n')  712 {  713 random\_password[21-1-i]='\0';  714 break;  715 }  716 }  717  718 }  719  720 output\_file=fopen("initial\_password.of.txt","a+");  721 time(&time\_ptr);  722  723 printf("已保存。\n");  724 //保存至initial\_password.of.txt  725 if(FindTeacher(input\_id)!=NULL)  726 {  727 fprintf(output\_file,"%s\t%llu\t%s\n",FindTeacher(input\_id)->name\_,input\_id,random\_password);  728 fprintf(output\_file,"Generated @ %s\n",ctime(&time\_ptr));  729 Wait();  730 }  731 if(FindStudent(input\_id)!=NULL)  732 {  733 fprintf(output\_file,"%s\t%llu\t%s\n",FindStudent(input\_id)->name\_,input\_id,random\_password);  734 fprintf(output\_file,"Generated @ %s\n",ctime(&time\_ptr));  735 Wait();  736 }  737  738 fclose(output\_file);  739 }  740 break;  741  742 case'4': //以用户身份登录  743 printf("请输入用户编号:");  744 if(scanf("%llu",&input\_id)==0)  745 {  746 printf("输入错误。");  747 Wait();  748 break;  749 }  750 if(ferror(stdin))  751 {  752 perror("Failed to read the id");  753 Pause();  754 break;  755 }  756  757 if(admin\_id\_set\_.count(input\_id)==0)  758 {  759 printf("编号不存在。");  760 Wait();  761 Wait();  762 break;  763 }  764 else  765 {  766 if(FindTeacher(input\_id)!=NULL)  767 {  768 FindTeacher(input\_id)->Login(this);  769 break;  770 }  771 if(FindStudent(input\_id)!=NULL)  772 {  773 FindStudent(input\_id)->Login(this);  774 break;  775 }  776 break;  777 }  778 break;  779 case'B':case'b':case 27:  780 stop=true;  781 break;  782 case'Q':case'q':  783 extern bool go\_on;  784 go\_on=false;  785 return 0;  786 default:  787 printf("输入错误。");  788 Wait();  789 break;  790 }  791 }  792 break;  793 case'3': //删除已有用户  794 for(;;)  795 {  796 uint64\_t input\_id=0;  797  798 printf("请输入用户编号:");  799 if(scanf("%llu",&input\_id)==0)  800 {  801 printf("输入错误。");  802 Wait();  803 break;  804 }  805 if(ferror(stdin))  806 {  807 perror("Failed to read the id");  808 Pause();  809 break;  810 }  811 fflush(stdin);  812 if(admin\_id\_set\_.count(input\_id)==0)  813 {  814 printf("编号不存在。");  815 Wait();  816 Wait();  817 break;  818 }  819 else  820 {  821 if(DeleteTeacher(input\_id)||DeleteStudent(input\_id))  822 {  823 printf("删除成功。");  824 Wait();  825 break;  826 }  827 else  828 {  829 perror("Unexpected result");  830 Pause();  831 break;  832 }  833 }  834 }  835 break;  836  837 case'4': //修改管理员登录密码  838 {  839 printf("请输入原来的密码:");  840 char password\_char[25]={'\0'}; //24字节密码经过Hash变为64位整数储存  841 for(int i=0;i<24;++i)  842 {  843 char get=getch();  844 switch(get)  845 {  846 case'\r':  847 password\_char[i]='\0';  848 i=24;  849 putch('\n');  850 Wait();  851 break;  852 case 8:  853 if(i>0)  854 {  855 putch(8);  856 putch(0);  857 putch(8);  858 --i;  859 }  860 --i;  861 break;  862 default:  863 if(get>=0x20&&get<=0x7E)  864 {  865 putch('\*');  866 password\_char[i]=get;  867 }  868 else  869 {  870 --i;  871 }  872 }  873 }  874 if(Hash(password\_char)!=hash\_)  875 {  876 printf("密码错误。");  877 Wait();  878 break;  879 }  880  881 printf("请输入新的密码:");  882 for(int i=0;i<24;++i)  883 {  884 char get=getch();  885 switch(get)  886 {  887 case'\r':  888 password\_char[i]='\0';  889 i=24;  890 putch('\n');  891 break;  892 case 8:  893 if(i>0)  894 {  895 putch(8);  896 putch(0);  897 putch(8);  898 --i;  899 }  900 --i;  901 break;  902 default:  903 if(get>=0x20&&get<=0x7E)  904 {  905 putch('\*');  906 password\_char[i]=get;  907 }  908 else  909 {  910 --i;  911 }  912 }  913 }  914 printf("请再次输入密码以确认:");  915 char password\_check[25]={'\0'}; //24字节密码经过Hash变为64位整数储存  916 for(int i=0;i<24;++i)  917 {  918 char get=getch();  919 switch(get)  920 {  921 case'\r':  922 password\_check[i]='\0';  923 i=24;  924 putch('\n');  925 break;  926 case 8:  927 if(i>0)  928 {  929 putch(8);  930 putch(0);  931 putch(8);  932 --i;  933 }  934 --i;  935 break;  936 default:  937 if(get>=0x20&&get<=0x7E)  938 {  939 putch('\*');  940 password\_check[i]=get;  941 }  942 else  943 {  944 --i;  945 }  946 }  947 }  948 if(strcmp(password\_char,password\_check)==0)  949 {  950 hash\_=Hash(password\_char);  951 printf("密码修改成功。");  952 Wait();  953 break;  954 }  955 else  956 {  957 printf("两次输入的密码不符。");  958 Wait();  959 Wait();  960 break;  961 }  962 break;  963 }  964  965 case'B':case'b':case 27:  966 return 0;  967  968 case'Q':case'q':  969 extern bool go\_on;  970 go\_on=false;  971 return 0;  972 }  973 }  974  975 perror("Unexpected result");  976 Pause();  977  978 return 0;  979 }  980  981 Teacher\* Admin::FindTeacher(char name[MAX\_NAME\_LENGTH])  982 {  983 for(vector<Teacher>::iterator iter=teacher\_vector\_.begin();iter!=teacher\_vector\_.end();++iter)  984 {  985 if(strcmp(name,iter->name\_)==0)  986 {  987 return &(\*iter);  988 }  989 }  990 return NULL;  991 }  992  993 Teacher\* Admin::FindTeacher(uint64\_t id)  994 {  995 for(vector<Teacher>::iterator iter=teacher\_vector\_.begin();iter!=teacher\_vector\_.end();++iter)  996 {  997 if(id==iter->id\_)  998 {  999 return &(\*iter);  1000 }  1001 }  1002 return NULL;  1003 }  1004 Student\* Admin::FindStudent(char name[MAX\_NAME\_LENGTH])  1005 {  1006 for(vector<Student>::iterator iter=student\_vector\_.begin();iter!=student\_vector\_.end();++iter)  1007 {  1008 if(strcmp(name,iter->name\_)==0)  1009 {  1010 return &(\*iter);  1011 }  1012 }  1013 return NULL;  1014 }  1015 Student\* Admin::FindStudent(uint64\_t id)  1016 {  1017 for(vector<Student>::iterator iter=student\_vector\_.begin();iter!=student\_vector\_.end();++iter)  1018 {  1019 if(id==iter->id\_)  1020 {  1021 return &(\*iter);  1022 }  1023 }  1024 return NULL;  1025 } |

score.h:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/score.h  1 #ifndef SCORE\_H\_  2 #define SCORE\_H\_  3  4 #define MAX\_SUBJECT\_NAME\_LENGTH 48  5  6 #include<cstdint>  7 #include<cstdlib>  8  9 class Score  10 {  11 public:  12 Score(const char\* subject,double mark,uint64\_t teacher);  13 Score(const char\* buffer,size\_t size);  14 Score(const Score& score);  15 ~Score();  16  17 size\_t WriteTo(char\* buffer,size\_t size);  18 size\_t ReadFrom(const char\* buffer,size\_t size);  19  20 friend class Teacher;  21 friend class Student;  22 friend class Admin;  23  24 protected:  25 char subject\_[MAX\_SUBJECT\_NAME\_LENGTH];  26 double mark\_;  27 uint64\_t teacher\_;  28 };  29 #endif // SCORE\_H\_ |

score.cpp:

|  |
| --- |
| Filename 学生成绩管理系统/codes/classes/score.cpp  1 #include<cstring>  2  3 #include"score.h"  4 #include"user.h"  5  6 using namespace std;  7  8 Score::Score(const char\* subject,double mark,uint64\_t teacher):mark\_(mark),teacher\_(teacher)  9 {  10 if(strlen(subject)>=MAX\_SUBJECT\_NAME\_LENGTH)  11 {  12 subject\_[0]='\0';  13 strcpy(subject\_+1,"SUBJECT\_NAME\_TOO\_LONG");  14 mark\_=0;  15 teacher\_=0;  16 }  17 else  18 {  19 strcpy(subject\_,subject);  20 mark\_=mark;  21 teacher\_=teacher;  22 }  23 }  24  25 Score::Score(const char\* buffer,size\_t size):mark\_(),teacher\_()  26 {  27 ReadFrom(buffer,size);  28 }  29  30 Score::Score(const Score& score):mark\_(score.mark\_),teacher\_(score.teacher\_)  31 {  32 for(size\_t i=0;i<MAX\_SUBJECT\_NAME\_LENGTH;++i)  33 {  34 subject\_[i]=score.subject\_[i];  35 }  36 }  37  38 Score::~Score(){}  39  40 size\_t Score::WriteTo(char\* buffer,size\_t size)  41 {  42 if(size!=MAX\_SUBJECT\_NAME\_LENGTH+sizeof(mark\_)+sizeof(teacher\_))  43 {  44 subject\_[0]='\0';  45 strcpy(subject\_+1,"BINARY\_DATA\_ERROR");  46 mark\_=0;  47 teacher\_=0;  48 return 0;  49 }  50 strncpy(buffer,subject\_,MAX\_SUBJECT\_NAME\_LENGTH);  51 buffer+=MAX\_SUBJECT\_NAME\_LENGTH;  52 for(size\_t i=0;i<sizeof(mark\_);++i)  53 {  54 buffer[i]=((char\*)&mark\_)[i];  55 }  56 buffer+=sizeof(mark\_);  57 for(size\_t i=0;i<sizeof(teacher\_);++i)  58 {  59 buffer[i]=((char\*)&teacher\_)[i];  60 }  61 buffer+=sizeof(teacher\_);  62 return size;  63 }  64  65 size\_t Score::ReadFrom(const char\* buffer,size\_t size)  66 {  67 if(size!=MAX\_SUBJECT\_NAME\_LENGTH+sizeof(mark\_)+sizeof(teacher\_))  68 {  69 subject\_[0]='\0';  70 strcpy(subject\_+1,"BINARY\_DATA\_ERROR");  71 mark\_=0;  72 teacher\_=0;  73 return 0;  74 }  75  76 strncpy(subject\_,buffer,MAX\_SUBJECT\_NAME\_LENGTH);  77 buffer+=MAX\_SUBJECT\_NAME\_LENGTH;  78  79 for(size\_t i=0;i<sizeof(mark\_);++i)  80 {  81 ((char\*)&mark\_)[i]=buffer[i];  82 }  83 buffer+=sizeof(mark\_);  84  85 for(size\_t i=0;i<sizeof(teacher\_);++i)  86 {  87 ((char\*)&teacher\_)[i]=buffer[i];  88 }  89 buffer+=sizeof(teacher\_);  90  91 return size;  92 } |

## 附录2：评分表

课程名称： 面向对象程序设计

|  |  |  |
| --- | --- | --- |
| **项 目** | **评 价** | |
| 设计方案的合理性与创新性 | **3×2** |  |
| 设计与调试结果 | **4×2** |  |
| 设计说明书的质量 | **1×2** |  |
| 程序基本要求涵盖情况 | **4×2** |  |
| 程序代码编写素养情况 | **2×2** |  |
| 课程设计周表现情况 | **1×2** |  |
| 综合成绩 | **15×2** |  |

教师签名：\_\_\_\_\_\_\_\_\_\_\_

日 期：\_\_\_\_\_\_\_\_\_\_\_

1. *注意！*修改密码对于管理员账户来说*非常重要*。 [↑](#footnote-ref-1)