**高校宿舍管理系统项目论文**

**实验课程：程序设计课程项目**

**实验项目：高校宿舍管理系统**

**指导老师：黄荔**

**开课时间：2016~2017年度第二学期**

**院 系：计算机学院**

**专 业：计算机**

**班 级：2015级计算机2班**

**学 号：20152100027**

**姓 名：詹萍**

**高校宿舍管理系统**

**中文摘要**

这是一个高校宿舍管理系统。越来越多高等院校拥有学生宿舍，宿舍管理面临着学生的信息登记、管理，宿舍分配、管理，查询学生信息，来访者登记等问题。因此，一个辅助管理员管理宿舍的高校宿舍管理系统就诞生了。

**关键词**

登录管理、学生基本信息管理、住宿管理、出入管理、信息查询

**英文摘要**

This is a dormitory management system in Colleges and Universities. More and more colleges and universities have their own student dormitories. That’s why dormitory management is faced with many problems such as manage and register students’ information, manage and allot dormitories, query information and register visitors and so on. Therefore, a dormitory management system in Colleges and Universities which can help administrators achieve those functions is needed and final is born.

**英文关键词**

①Login Management ②Student Basic Information Management ③Accommodation Management ④Entry And Exit Management ⑤Information Inquiry

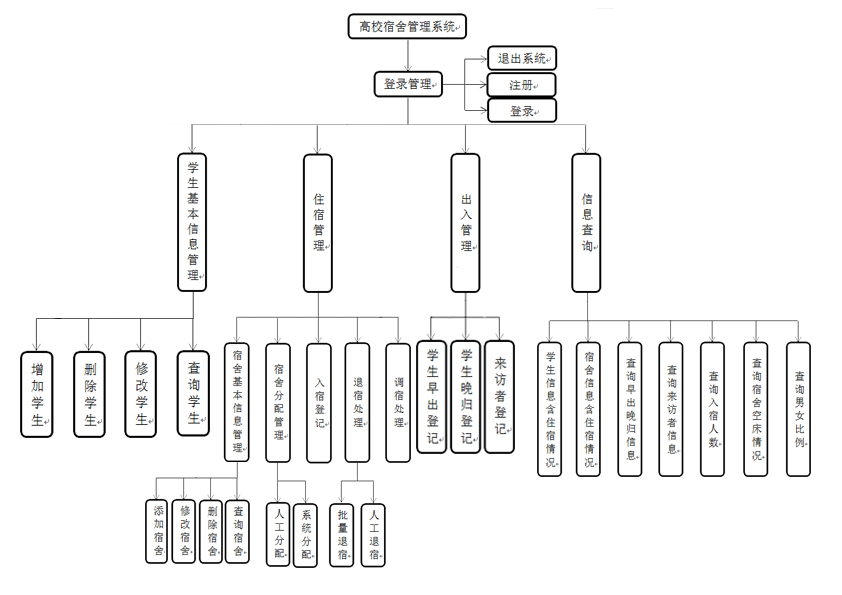
**第一章 前言**

随着社会经济的发展，越来越多的人能够接受高等教育，走进高等院校。现状是大多数学生在学校内住宿，为了使管理变得方便高效，我们需要一个管理系统来辅助管理员管理高等院校的学生宿舍。

这个高校宿舍管理系统主要用于辅助管理员管理高校宿舍，其功能大致分为以下几部分：①管理员登录模块 ②学生基本信息管理 ③住宿管理 ④出入管理 ⑤信息查询。通过以上主要功能来实现高校学生宿舍管理。

**第二章 系统功能描述**

**1.总流程框架**

****

**2.功能描述**

1. **主框架及功能描述：**

进入高校宿舍管理系统后首先进入登录管理。登录管理分为三个部分：退出系统、注册和登录。

选择退出系统，则直接结束程序。选择注册，则注册成功后返回登录管理的界面，可以选择退出系统、注册或者登陆。选择登录，只有当用户名和密码输入准确无误才能进入其他功能的选择。

只有登陆成功才能进入高校宿舍管理系统。进入系统后又分为四个部分：学生基本信息管理、住宿管理、出入管理和信息查询。

出入管理

高校宿舍管理系统

登录管理

学生基本信息管理

住宿管理

信息查询

退出系统

注册

登录

1. **各功能框架细分：**
2. 学生基本信息管理：

进入学生基本信息管理模块后分为三个部分：增加学生基本信息，删除学生基本信息、修改学生基本信息和查询学生基本信息。

学生基本信息管理

增加学生

删除学生

修改学生

查询学生

1. 住宿管理:

进入住宿管理后分为五个模块：宿舍基本信息管理、宿舍分配管理、入宿登记、退宿处理和调宿处理。

宿舍基本信息管理又分为四个模块：添加宿舍，修改宿舍信息，删除宿舍和查询宿舍信息。

宿舍分配管理分为人工分配和系统批量分配。

退宿处理分为批量退宿和人工退宿。

退宿处理

宿舍基本信息管理

宿舍分配管理

调宿处理

入宿登记

住宿管理

添加宿舍

查询宿舍

修改宿舍

删除宿舍

人工分配

系统分配

批量退宿

人工退宿

1. 出入管理:

出入管理分为三个模块：学生早出登记、学生晚归登记和来访者登记。

出入管理

学生早出登记

学生晚归登记

来访者登记

1. 信息查询：

信息查询分为七个模块：查询学生信息含住宿情况，查询宿舍信息含住宿情况、查询早出晚归信息、查询来访者信息、查询入宿人数、查询宿舍空床情况和查询男女比例。

信息查询

学生信息含住宿情况

宿舍信息含住宿情况

查询早出晚归信息

查询来访者信息

查询男女比例

查询宿舍空床情况

查询入宿人数

**3.开发平台及工具介绍**

**（1）工具介绍**

本项目选择[Code::Blocks](http://baike.baidu.com/view/1562377.htm)作为开发工具。[Code::Blocks](http://baike.baidu.com/view/1562377.htm)是一个开源、免费、跨平台（支持[Windows](http://baike.baidu.com/view/4821.htm)、[GNU/Linux](http://baike.baidu.com/view/2560767.htm)、[Mac OS X](http://baike.baidu.com/view/24778.htm)以及其他[类UNIX](http://baike.baidu.com/view/3289073.htm)）、支持插件扩展的C/C++[集成开发环境](http://baike.baidu.com/view/14867.htm)。Code::Blocks的源码使用GPL3.0发布，是免费[自由软件](http://baike.baidu.com/view/20965.htm)。

**（2）开发平台**

Windows或者其他平台，从软件源中安装即可

**第三章 系统设计过程**

1. **系统功能**
2. **登录管理**

登录管理分为登录、注册和退出系统三部分。允许管理员登陆，允许注册账号以及退出系统。只有登录成功，也即用户名和密码输入均正确才能进入系统，实现其它操作。

1. **学生基本信息管理**

进入学生基本信息管理模块后分为三个部分：增加学生基本信息，删除学生基本信息、修改学生基本信息和查询学生基本信息。

学生信息的增删查改与其住宿的信息也有关。如：入宿后相当于已经确定性别，因此性别不能修改；并且入宿后不能删除学生信息。

1. **住宿管理**

住宿管理分为宿舍基本信息管理、宿舍分配管理、入宿登记、退宿处理和调宿处理五个部分。

宿舍基本信息管理又分为添加宿舍，修改宿舍信息，删除宿舍和查询宿舍信息四个部分

宿舍分配管理分为人工分配和系统批量分配。可以由管理员选择分配宿舍的方式。当有一批学生时最好选择批量分配，批量分配是先按照学生性别划分学生，再按照学生的学号进行分配，当学生数量较少时可以选择人工分配，比较灵活。

退宿处理分为批量退宿和人工退宿。可以由管理员选择退宿登记的方式。

1. **出入管理**

出入管理分为学生早出登记、学生晚归登记和来访者登记三部分。

1. **信息查询**

信息查询分为查询学生信息含住宿情况，查询宿舍信息含住宿情况、查询早出晚归信息、查询来访者信息、查询入宿人数、查询宿舍空床情况和查询男女比例七个部分。

1. **设计思想**
   1. 学生宿舍管理系统主要是由学生和宿舍这两大模块组成的，因此我首先选择建立学生类和宿舍类分别来描述学生和宿舍这两个具体对象。
   2. 根据系统的功能需要，又建立了一个描述管理员的类。其中包括信息用户名和密码等数据成员。
   3. 出入管理模块需要记录学生信息，由此我建立了登记学生早出晚归信息的类以及登记来访者信息的类。
   4. 最后是将所有的模块联系起来实现系统的功能，相当于需要一个类来对所有的信息进行连接操作，由此我建立了一个功能类，实现所有模块相连的操作。
2. **数据详细描述**
   1. **基础类：Student类和Dormitory类**

对于一个高校宿舍管理系统而言，从最基础上看，所面对的不外乎是学生和宿舍这两个类。这两个类分别包含了学生和宿舍最基本的数据成员和函数成员，是实现系统各个功能的基础。

**（1）Student学生类：**

**① 数据成员：**

姓名name、性别sex、学号number、电话号码phone、宿舍号roomNumber床位号bedNumber、入宿标记mark和删除标记note。其中入宿标记：0表示离宿状态、1表示入宿状态。删除标记： 0表示已删、1表示未删。

**② 函数成员：**

**Ⅰ.构造函数：**

Student()：在构造函数中对宿舍号和床位号进行初始化，均为0。

**Ⅱ.获取信息的函数：**

char \*getname()：获得姓名，返回学生姓名。

char getsex()：获得性别，返回学生性别。

int getnumber()：获得学号，返回学生学号。

char \*getphone()：获得电话号码，返回学生电话号码。

int getroomNumber()：获得宿舍号，返回学生宿舍号。

int getbedNumber()：获得床位号，返回学生床位号。

bool getmark()：获得入宿状态，返回学生入宿状态。

bool getnote()：获得删除标记，返回学生床位号。

**Ⅲ.设置信息的函数：**

void addstudent(char \*na,char s,int num,char \*ph)：添加新学生信息。

void setname(char \*na)：设置姓名，用于修改学生姓名。

void setphone(char \*ph)：设置电话号码，用于修改学生电话号码。

void setsex(char s)：设置性别，用于修改学生性别，入宿后不允许修改。

void setroom\_bed(int room,int bed)：设置宿舍号和床位号，用于分配宿舍和床位。

**Ⅳ.其他功能的函数：**

void delstudent()：删除学生，将删除标记记为0，表示删除学生。

void show()：展示学生信息，打印该学生的信息。

void live()：入宿，将入宿标记记为1，表示该学生已经入宿。

void leave()：退宿，将入宿标记记为0，表示该学生已经退宿。

**（2）Dormitory宿舍类：**

**① 数据成员：**

宿舍号dormitoryNumber，床位号bedNum[4]、床位使用情况bed[4]、宿舍使用性别dormitorySex，每个床位的使用者学号beduser[4]、删除标记delNote。其中床位使用情况：0表示空床、1表示已使用。宿舍使用性别：女宿舍为f、男宿舍为m。删除标记：0表示删除、1表示未删。

**② 函数成员：**

**Ⅰ.构造函数：**

Dormitory()：在构造函数中将床位使用情况初始化为0，即空床。将每个床位的使用者学号初始化为0，表示暂无使用者。床位号定为1、2、3、4。

**Ⅱ.获取信息的函数：**

int getdormitoryNumber()：获得宿舍号，返回宿舍号。

bool getdelNote()：获得删除标记，返回删除标记。

char getdormitorySex()：获得宿舍的使用性别，返回宿舍使用性别。

int use\_dormitory()：获得宿舍的使用情况，返回宿舍已被使用的床位的数量。

bool bed\_use\_case(int bedNum)：获得具体某个床位的使用情况，将床位号做为形参，获得该床位的使用情况。已被使用返回1，未被使用返回0。

**Ⅲ.设置信息的函数：**

void setdormitoryNumber(int dorNum)：设置宿舍号，用于修改宿舍信息。

void setdormitorySex(char sex)：设置宿舍使用性别，用于修改宿舍信息。

void addDormitory(int num,char sex)：增加宿舍信息。

**Ⅳ.其他功能的函数：**

void showbed()：展示全部床位的使用情况，打印出每张床是否被使用，若被使用，则打印出使用者的学号。

void deletedormitory()：删除宿舍，将删除标记记为0，表示删除宿舍。

void not\_use\_bed(int bedNum)：空出床位，将床位号作为形参，空出此张床。将床位使用情况标记为0，表示空床。用于退宿时空出床位、调宿时空出床位。

void use\_bed(int bedNum,int num)：使用床位，将床位号和学生学号作为形参，将床位使用情况标记为1，表示该床位已被使用。

* 1. **其他功能类：Administrator类、Function类、Morning\_Evening\_record和Visitor类**

**（1）Administrator管理员类：**

* 1. **数据成员：**

用户名userName和密码password。

**② 函数成员：**

**Ⅰ. 获取信息的函数：**

char \*getUserName()：获取用户名，用于验证登录信息是否正确以及用户名是否已被注册。

char \*getPassword()：获取密码，用于验证登录信息是否正确。

**Ⅱ. 设置信息的函数：**

void addAdministrator(char \*na,char \*pass)：增加管理员，注册时将用户名和密码做为形参，填入对象数据成员中。

**（2）Morning\_Evening\_record早出晚归登记类：**

* 1. **数据成员：**

学生姓名studentName，学号studentNumber，早出次数morningTimes，晚归此数eveningTimes。

* 1. **函数成员：**

**Ⅰ. 构造函数：**

Morning\_Evening\_record()：构造函数，初始化数据成员morningTimes和eveningTimes为0。

**Ⅱ. 获取信息的函数：**

int getNum()：获取学号，在Function类中查找学生信息时使用。

**Ⅲ. 设置信息的函数：**

void addStud(char \*na,int nu)：添加早出晚归的学生，将姓名、学号作为形参，添加信息。

**Ⅳ.其他功能的函数：**

void morningRecord()：学生早出次数增加，相当于登记早出信息。

void eveningRecord()：学生晚归次数增加，相当于登记晚归信息。

void showStudentInformation()：打印出学生早出晚归信息。

**（3）Visitor来访者登记类：**

* 1. **数据成员：**

来访者姓名visitorName，来访者性别visitorSex，来访时间visitTime，离开时间leaveTime，来访理由reason和访问宿舍号visitDorNum。

**② 函数成员：**

**Ⅰ. 设置信息的函数：**

void setVisitInformation(char \*na,char se,char \*v\_t,char \*l\_t,char \*re,int dorNum)：添加来访者信息，将来访者姓名，性别，来访时间，离开时间，来访理由，访问宿舍号作为形参，添加来访者。

**Ⅱ.其他功能的函数：**

void showVisitorInformation()：打印来访者的信息。

**（4）Function功能类：**

1. **数据成员：**

**<1>登录管理模块：**

管理员类的对象数组administrator[Maxa]，其中Maxa=20，记录管理员数量整型adNum。最多有20个管理员。

**<2>学生管理模块：**

学生类的对象数组student[Maxs]，其中Maxs=500，记录学生数量的整型stuNum，学生类的指针\*ps。最多有500个学生。

**<3>宿舍管理模块：**

宿舍类的对象数组dormitory[Maxd]，其中Maxd=200，记录宿舍数量的整型dorNum，宿舍类的指针\*pd。最多有200个宿舍。

**<4>出入管理模块：**

来访者类的对象数组visitor[Maxv]，其中Maxv=100，记录来访者数量的整型visNum。最多有100个来访者记录。

早出晚归类的对象数组stud[Max]，其中Max=500，记录早出晚归学生数量的整型studNum。早出晚归类的指针\*pr。最多有500个早出晚归记录。

1. **函数成员：**

**Ⅰ.构造函数：**Function()

**<1>登录管理模块：**

打开文件administrator.dat，判断是否正常打开文件，将管理员数据从文件中读入。并用adNum统计管理员数目。

**<2>学生管理模块：**

打开文件student.dat，判断是否正常打开文件，将学生数据从文件中读入。并用stuNum统计学生数目。

**<3>宿舍管理模块：**

打开文件dormitory.dat，判断是否正常打开文件，将宿舍数据从文件中读入。并用dorNum统计宿舍数目。

**<4>早出晚归管理模块：**

打开文件morning\_evening\_record.dat，判断是否正常打开文件，将早出晚归学生数据从文件中读入。并用studNum统计早出晚归学生数目。

**<5>来访者模块：**

打开文件visitor.dat，判断是否正常打开文件，将来访者数据从文件中读入。并用visNum统计来访者数目。

**Ⅱ.析构函数：**~Function()

**<1>登录管理模块：**

打开文件administrator.dat，判断是否正常打开文件，根据adNum统计的数目，将管理员数据写入文件。

**<2>学生管理模块：**

打开文件student.dat，判断是否正常打开文件，根据stuNum统计的数目，将学生数据写入文件。

**<3>宿舍管理模块：**

打开文件dormitory.dat，判断是否正常打开文件，根据dorNum统计的数目，将宿舍数据写入文件。

**<4>早出晚归管理模块：**

打开文件morning\_evening\_record.dat，判断是否正常打开文件，根据studNum统计的数目，将早出晚归学生数据写入文件。

**<5>来访者模块：**

打开文件visitor.dat，判断是否正常打开文件，根据visNum统计的数目，将来访者数据写入文件。

**Ⅲ.其他功能的函数：**

**<1>登录管理模块：**

int queryUserName(char \*na)：查询用户名。将用户名和密码做为形参查找管理员的信息。若完全查找不到信息，则返回0。若找到用户名而密码不符返回2。若用户名和密码均正确返回1。这个函数用于判断登录时用户名与密码的正确与否。

int queryUser(char \*na,char \*pass)：查询用户。将用户名做为形参查找用户名是否存在。若存在，则返回1，若不存在，则返回0。这个函数用于注册时判断该用户名是否已被注册。

void Register()：注册账号。输入用户名和密码。先查找用户名是否已存在。若不存在，则管理员数目adNum自增1，并调用添加管理员的函数将信息填入，表示注册成功。返回选择页面。

int sign\_in()：登录。输入用户名和密码。查找用户名和密码是否对应正确。若均正确，提示登陆成功，进入其他功能选择的模块。

int login()：登录管理。

登录管理分为 3个模块：用switch语句进行选择

1）登录。

2）注册。

3）退出系统。结束程序。

调用以上相应的函数来实现登录和注册的功能。

**<2>学生基本信息管理模块：**

void addstudent(char \*na,char s,int num,char \*ph)：增加学生。将学生姓名，性别，学号和电话号码作为形参。先查找该学生是否已经存在，若不存在。调用学生类的添加学生函数将学生的信息填入。

void deletestudent(int num)：删除学生。将学生学号作为形参，按学号查找学生，若存在此学生，调用学生类的删除函数，将学生的删除标记记为0。其中0表示已删除、1表示未删除。

void modify(Student \*p)：修改学生信息。先查找到学生，将学生的地址作为形参调用此修改信息的函数。

修改学生信息分为3个模块：用switch语句进行选择

1）修改姓名。调用学生类设置姓名的函数来修改学生姓名。

2）修改电话。调用学生类设置电话号码的函数来修改学生的电话号码。

3） 修改性别。先判断学生是否已经入宿。只有学生尚未入宿才能修改性别。调用学生类设置性别的函数来修改学生的性别。

Student \*queryNum(int num)：查找学生。将学生的学号作为形参，查找学生的信息，若查找不到信息，则返回NULL，否则返回该学生对象的地址。

void studentdatabase()：学生基本信息管理。

学生基本信息管理分为4个模块：用switch语句进行选择

1）添加学生。

2）查询学生信息。

3）修改学生信息。

4）删除学生。

输入相应信息并调用以上相应的函数来实现功能。

**<3>住宿管理模块：**

void addDormitory(int num,char sex)：添加宿舍。将宿舍号和宿舍使用性别做为形参调用此函数。函数中先查找此宿舍号是否已存在，若不存在，则宿舍数量dorNum自增1并调用宿舍类的添加宿舍的函数实现宿舍的添加。

void modifyDormitory(int roomNum)：修改宿舍信息。将要修改信息的宿舍号作为形参。先查找宿舍号判断宿舍是否存在。若存在，则判断宿舍是否为空，只有空宿舍的信息才能修改。

修改宿舍信息分为2个模块：用switch语句进行选择

1）修改宿舍号。先判断该宿舍号是否已存在。

2）修改宿舍使用性别。

调用相应的设置宿舍信息的函数。

void deleteDormitory(int room)：删除宿舍。将要删除的宿舍的宿舍号做为形参。先查找宿舍号判断宿舍是否存在。若存在，则判断宿舍是否为空，只有空宿舍才能删除。当宿舍存在并且为空时，调用删除宿舍的函数，将宿舍的删除标记记为0，其中1表示未删除、0表示删除。

Dormitory \*queryroom(int num)：查询宿舍信息。将要查找的宿舍的宿舍号作为形参。若查找不到信息，则返回NULL，否则返回该宿舍对象的地址。

void allotDormitory()：分配宿舍。

宿舍分配分为2个模块：用switch语句进行选择

1）人工分配。先查找到要分配的学生，再判断学生是否已经分配了床位。若未分配，打印出所有有空床位的宿舍的宿舍号和宿舍使用性别。再选择宿舍和床位。在性别一致的情况下在学生信息填入宿舍中，将宿舍信息填入学生中。

2）系统批量分配。先判断宿舍数量是否足够学生居住。将所有学生分男女性别划分成两部分，再按按学号排序。然后按性别和学号把学生信息依次填入每个宿舍中，再将宿舍信息依次填入学生中。

void liveDormitory()：入宿登记。先按学号找到该学生，然后判断学生是否已经登记入宿，若未登记，则调用学生类的入宿函数将入宿标记记为1。其中1表示入宿、0表示离宿。

void leaveDormitory()：退宿处理。

退宿处理分为2个模块：用switch语句进行选择

1）人工退宿。先按学号找到该学生，然后判断学生是否已经登记入宿，若已登记，则调用学生类的离宿函数将入宿标记记为0。其中1表示入宿、0表示离宿。将学生的信息删除并将宿舍中学生的信息删去。

2）批量退宿。用循环将所有的学生的入宿标记记为0。并将所有的学生删除，即直接将学生数量stuNum记为-1。再用循环将所有的学生的信息从宿舍中删去。

void adjustDormitory()：调宿处理。

调宿处理分为2个模块：用switch语句进行选择

1）两学生交换宿舍。先按学号查找到两名学生，再判断他们的性别是否一致。若一致，交换两个学生的住宿信息，同时交换两个宿舍的学生信息。

2）调整一名学生的宿舍。先按学号查找到该学生，再打印出所有有空床位的宿舍信息。进行选择，选择后将学生信息填入宿舍中，将宿舍信息填入学生中，并将原宿舍中的学生信息删除。

void dormitorydatabase()：宿舍基本信息管理。

宿舍基本信息管理分为4个模块：用switch语句进行选择

1）添加宿舍。

2）查询宿舍信息。

3）修改宿舍信息。

4）删除宿舍。

输入相应信息并调用以上相应的函数来实现功能。

void dormitoryManage()：住宿管理。

住宿管理分为5个模块：用switch语句进行选择

1）宿舍基本信息管理。

2）宿舍分配管理。

3）入宿登记。

4）退宿处理。

5）调宿处理。

调用以上相应的函数来实现功能。

**<4>出入管理模块：**

Morning\_Evening\_record \*query(int num)：查找早出晚归学生。通过学号查找学生是否已经有过早出晚归的信息。

void morning\_go\_out()：早出登记。添加早出晚归学生信息包括姓名学号，将早出晚归标记记为 0。其中0表示早出、1表示晚归。早出晚归学生人数studNum自增1。

void evening\_go\_back()：晚归登记。添加早出晚归学生信息包括姓名学号，将早出晚归标记记为1.其中0表示早出、1表示晚归。早出晚归学生人数studNum自增1。

void visit\_record()：来访者登记。登记来访者姓名，来访者性别，来访时间，离开时间，来访理由和访问宿舍号。判断该宿舍是否为空，若不为空，则来访者人数visNum自增1并调用来访者类的添加来访者信息的函数。

void get\_in\_out\_manage()：出入管理。

出入管理分为3个模块：用switch语句进行选择

1）早出登记。

2）晚归登记。

3）来访者登记。

分别调用相应的功能函数。

**<5>信息查询模块：**

void query\_information()：信息查询。

信息查询分为以下7种查询方式：用switch语句进行选择

1）查询学生信息含住宿信息。

2）查询宿舍信息含住宿信息。

3）查询早出晚归的信息。

4）查询来访者信息。

5）查询入宿人数。

6）查询宿舍空床情况

7）查询住宿男女比例

分别调用相应的查询函数并进行计算的操作。

* 1. **主函数main**

1. **初始界面设计 —— 以美观为主**
2. **先进入登录模块**

只有登录模块返回值为0的时候结束系统，返回值为1时登录成功，进入 功能选择模块。

1. **高校宿舍管理系统的功能选择**

功能选择分为4个模块：用switch语句进行选择

1）学生基本信息管理。

2）住宿管理。

3）出入管理。

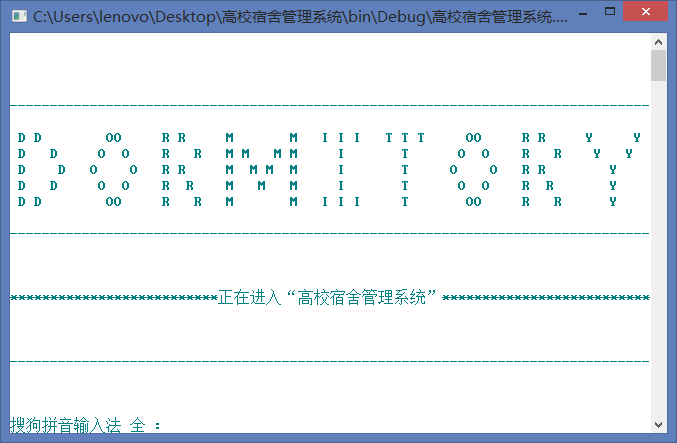
4）信息查询。

分别调用相应的函数实现上述功能。

**第四章 系统测试**

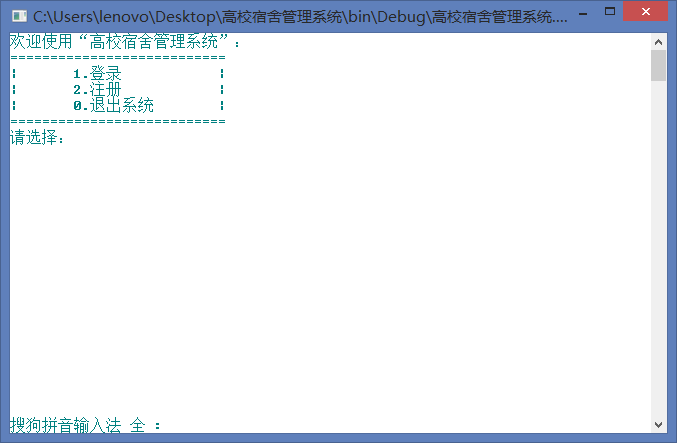
1. **系统的优点与不足**
2. **系统的优点**
3. 登录时，密码的输入使用“\*”来显示，更接近现实使用的需要，使信息不易暴露，比较友好普适。
4. 在程序中大量引入两个函数：
   1. Sleep(1000)：这个函数的插入出会使页面在该处停留括号内参数填写的时间，单位是毫秒。头文件是：#include <windows.h>。
   2. system(“cls”)：这个函数的作业是将页面清除。将它与上面的函数结合是页面整体美观简洁，实现跳转的功能。
5. 信息查询的方式多达七种，方便管理员了解学生宿舍的各方面信息，比较全面。
6. 登录管理增加了管理员注册的功能。
7. 每个功能的菜单均附有回退的功能，使管理员可以方便地在各个功能板块之间跳转。
8. 程序的健壮性比较强。
9. 通过函数system("color F3")修改控制台的颜色，第一个数字为背景，第二个是前景，使得窗口更加美观。
10. **系统的不足之处**
11. 宿舍没有划分为多栋，使得男女生的宿舍可以穿插在一起，这与实际不合。
12. 学生的学号是整型类型，数据的长度不够，因此我将学号限制为4位，无法实现较长位数的学生学号。
13. 早出晚归的登记中没有登记日期的成员，无法查询到具体的登记日期。
14. **遇到的问题及解决方法**
15. **项目的健壮性问题：**
    1. 问题：缺少判别输入出错的功能。如：添加学生时应该添加判断学生学号是否重复的功能、菜单选项应添加判断选项是否正确的功能等。
    2. 解决方法：在相应的地方加入相应的判别条件。添加学生时加入调用查找学生的函数来判别是否重复、菜单选项用if条件语句来划分选项是否正确等。
16. **分配宿舍的问题：**
    1. 问题：分配宿舍的系统批量分配。系统批量分配的划分宿舍和分配学生的算法实现比较难以实现。按学号来分配宿舍的算法比较难。
    2. 解决方法：引入两个整型数组来分别存放男生和女生的学号，然后对这两个数组分别排序。通过排序后的学号来逐个将学生找到并填入宿舍的信息。再将相应的学生信息填入相应的宿舍中。
17. **早出晚归类对象空间浪费的问题：**
    1. 问题：早出晚归类的对象数组，每次登记都新建立学生对象会有重复的现象，导致浪费空间。
    2. 解决方法：在早出晚归类总引入早出次数和晚归此数的数据成员，并添加查找学生的函数。当有登记信息时，先查找学生是否已经登记过：若没有，新建对象，再增加次数，若有登记的信息，直接增加次数。
18. **管理员类的函数功能复杂：**
    1. 问题：管理员类的函数功能直接与文件的读写挂钩。如：登录、注册等功能需要查找判别是否重复的问题，直接从文件读写会使程序代码更复杂。
    2. 解决方法：将登录等函数放到Function类中，在Function类中建立管理员的对象数组，通过构造函数从文件中读入管理员的信息。这样每次检测是否重复只需要在数组中查找，无需从文件读写。再通过构造函数将数组包括新添加的管理员信息写入文件中。
19. **测试结果**
20. **初始界面**

以美观为主。采用将宿舍的英文DORMITORY排布在界面中的设计，使项目功能一目了然。



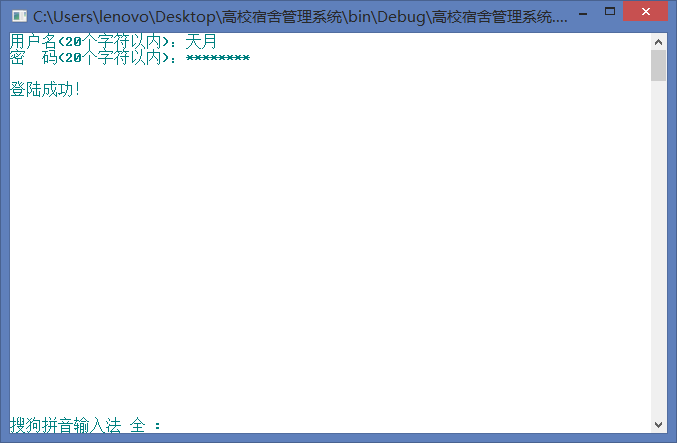
1. **登录管理**

1）登录管理菜单：



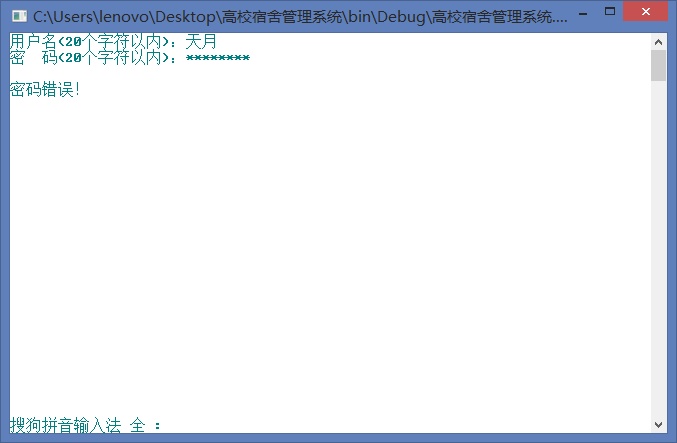
2）选择登陆：

* 1. 登陆成功：用户名和密码输入正确

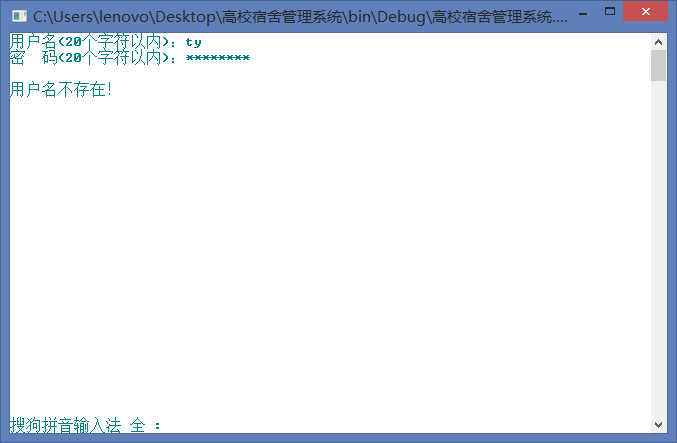


* 1. 登录失败情况

密码输入错误的情况。

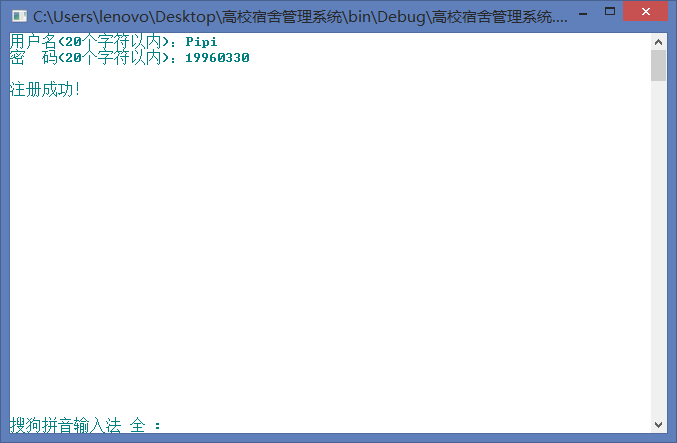


用户名不存在的情况。

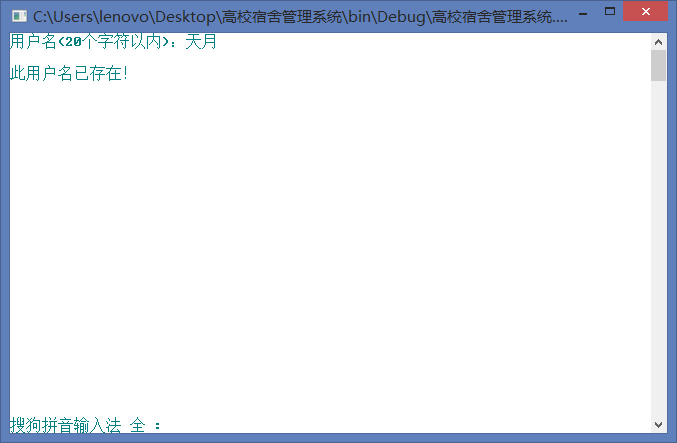


3）选择注册：

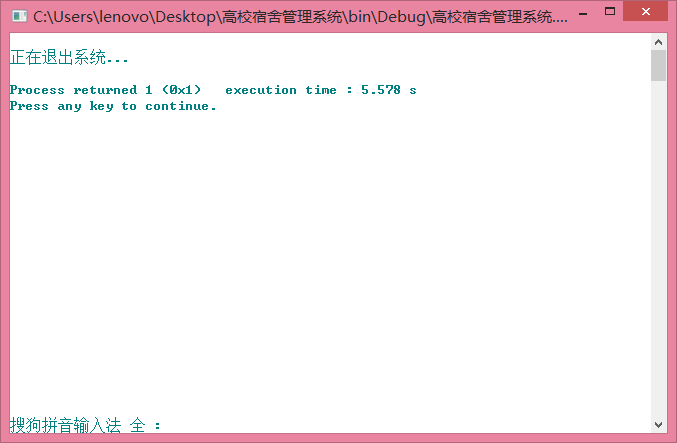
* 1. 注册成功：



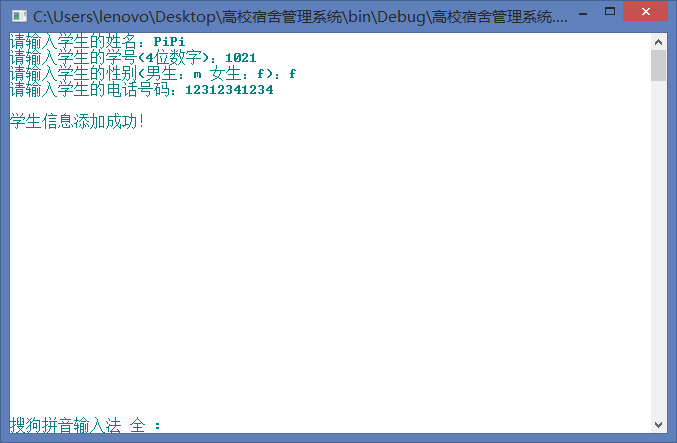
* 1. 注册失败情况：用户名已存在



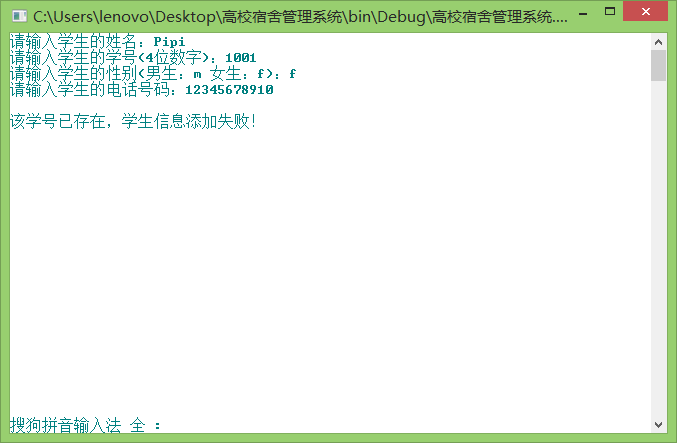
4）选择退出系统：



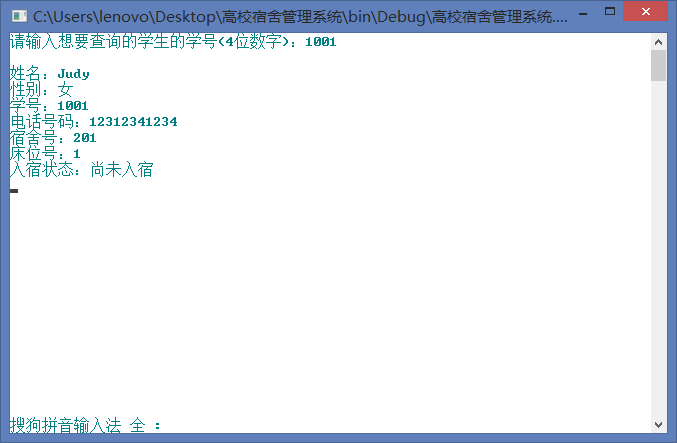
1. **学生基本信息管理**
2. 添加学生：
   1. 添加学生：



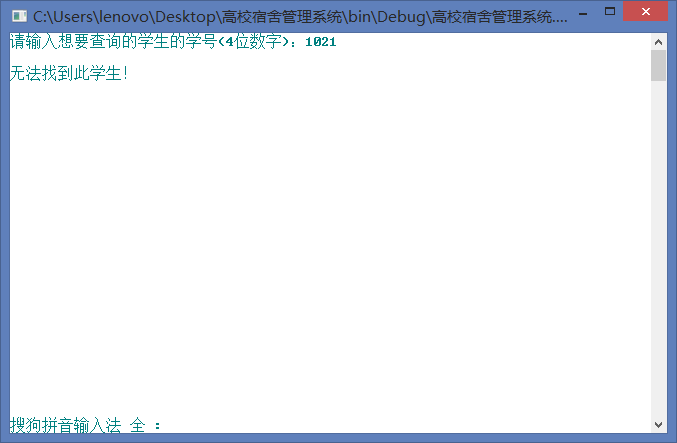
* 1. 添加学生失败情况：该学号已存在



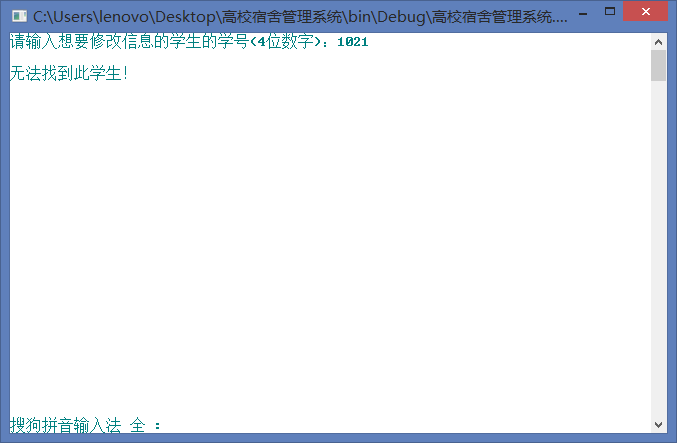
1. 查询学生信息：
   1. 查询学生信息：



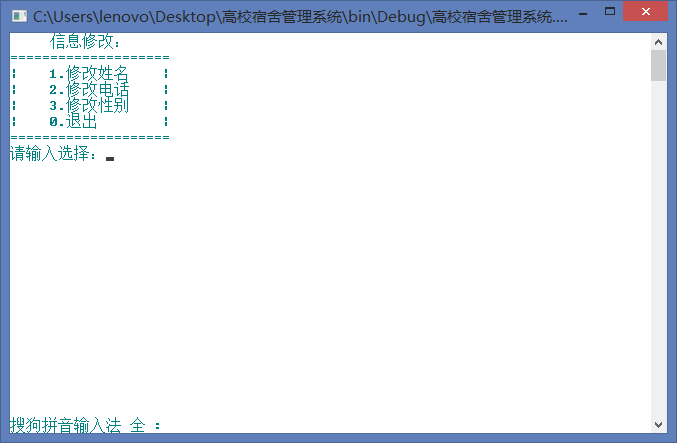
* 1. 查询学生信息失败情况：学号不存在



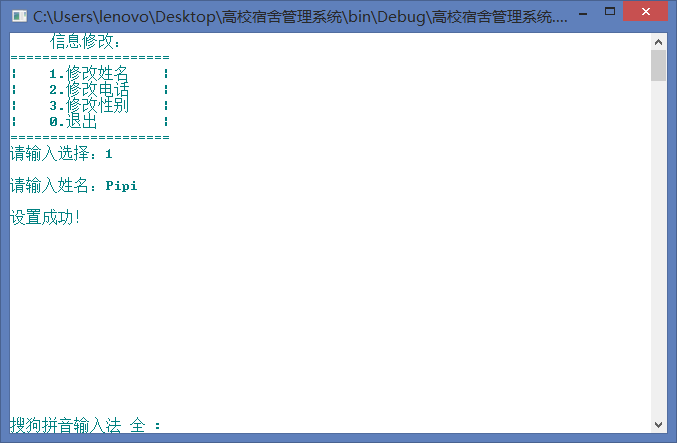
1. 修改学生信息：
   1. 选择学生失败情况：学号输入错误



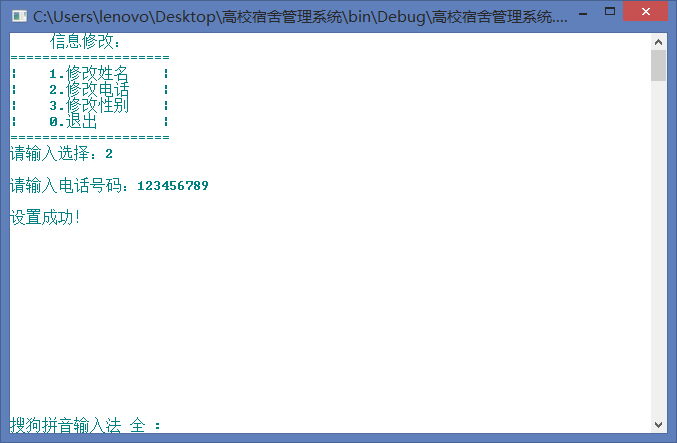
* 1. 修改学生信息菜单：



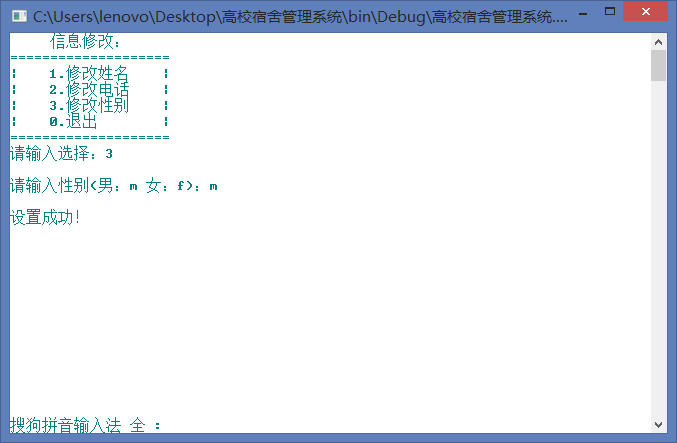
* 1. 修改学生姓名：

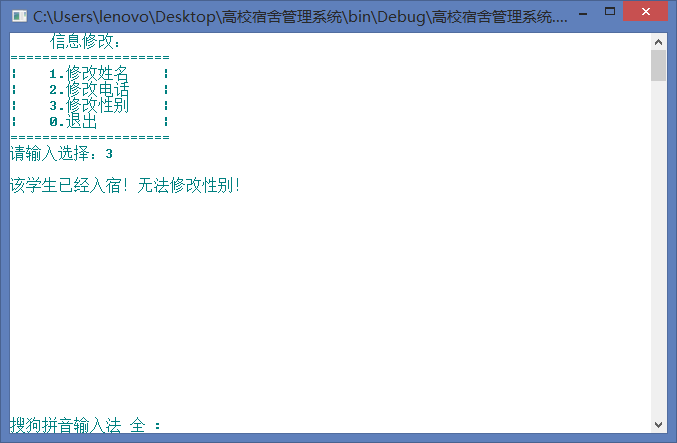


* 1. 修改学生电话：

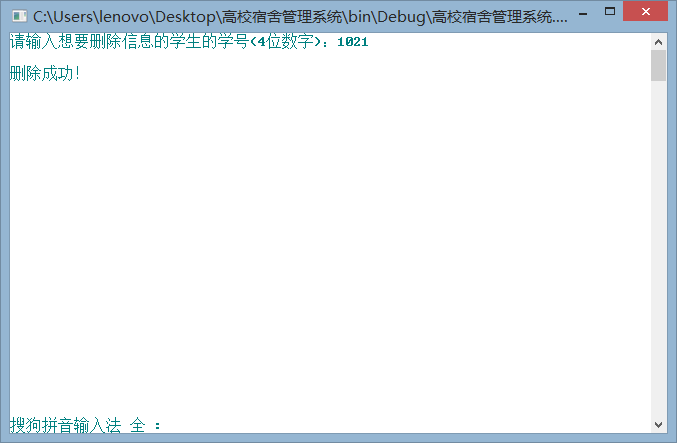


* 1. 修改学生性别：入宿后无法修改学生性别

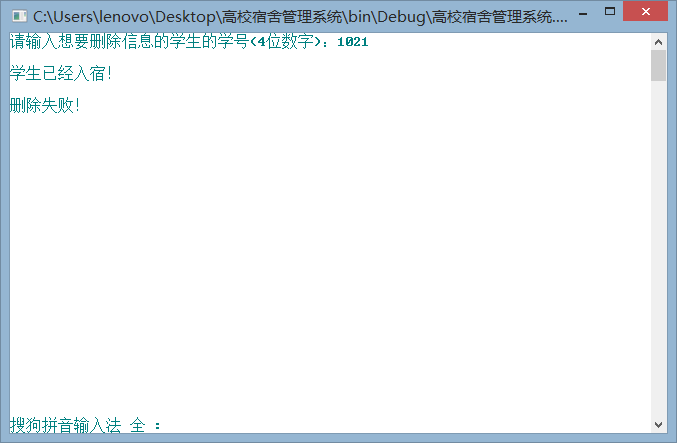




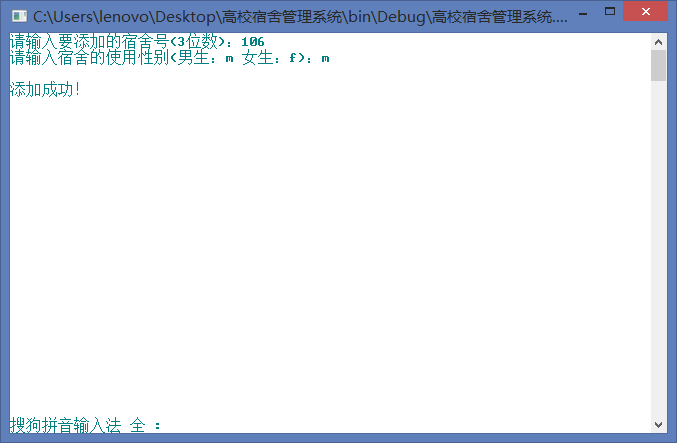
1. 删除学生：
   1. 删除学生：



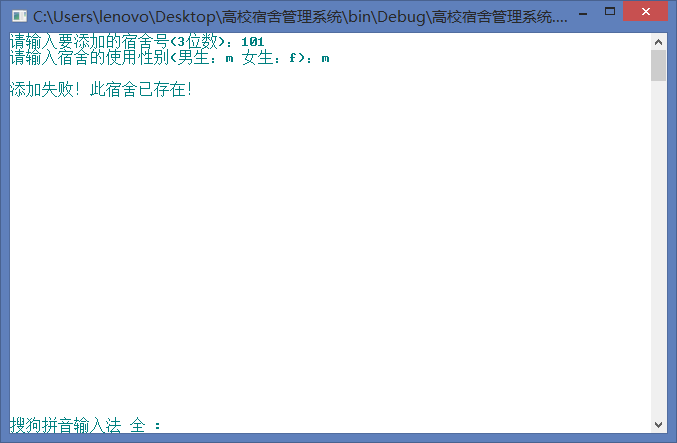
* 1. 入宿后无法删除学生：



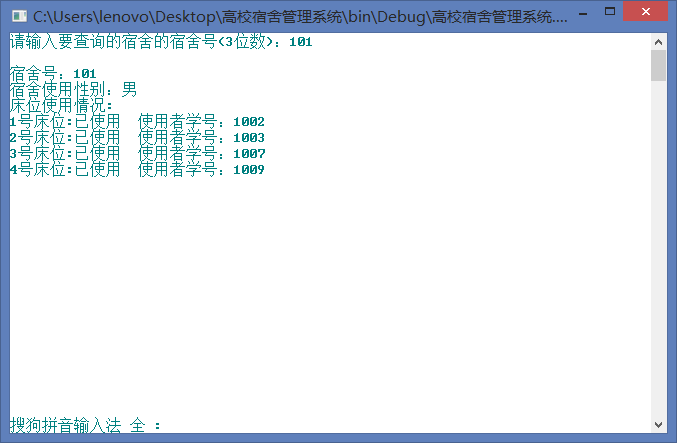
1. **住宿管理**
2. 宿舍基本信息管理：
3. 添加宿舍：
   1. 添加宿舍：

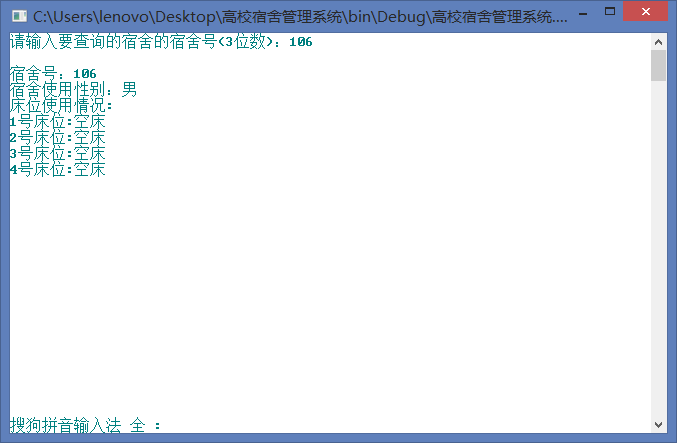


* 1. 添加宿舍失败情况：宿舍号重复

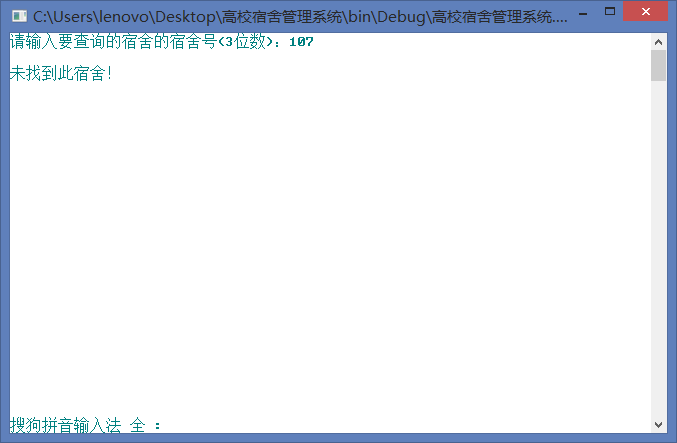


1. 查询宿舍信息：
   1. 查询宿舍信息：

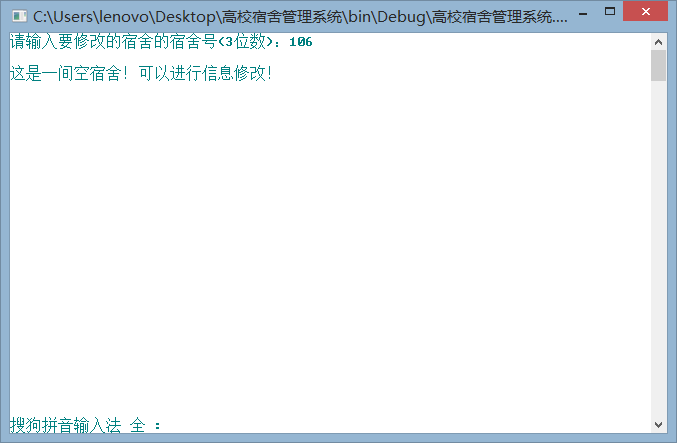


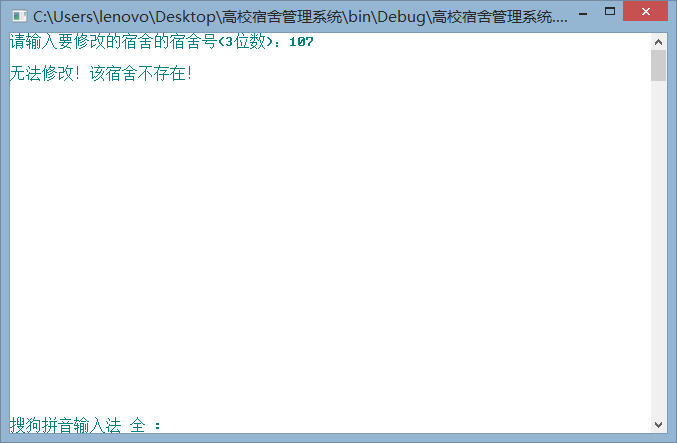


* 1. 查询宿舍信息失败情况：宿舍号不存在

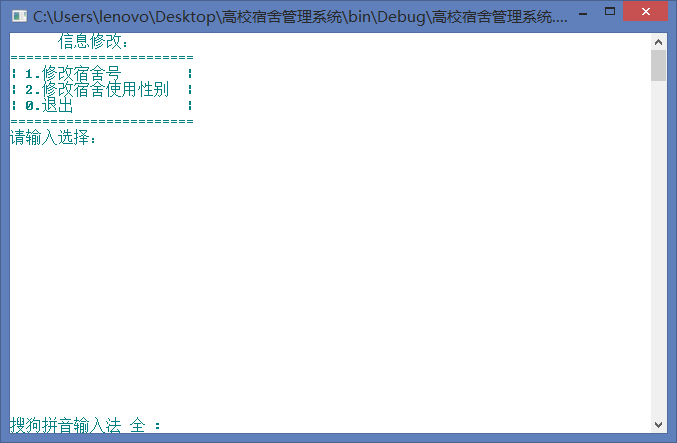


1. 修改宿舍信息：
   1. 查找宿舍号及失败情况：只有空宿舍才能修改信息

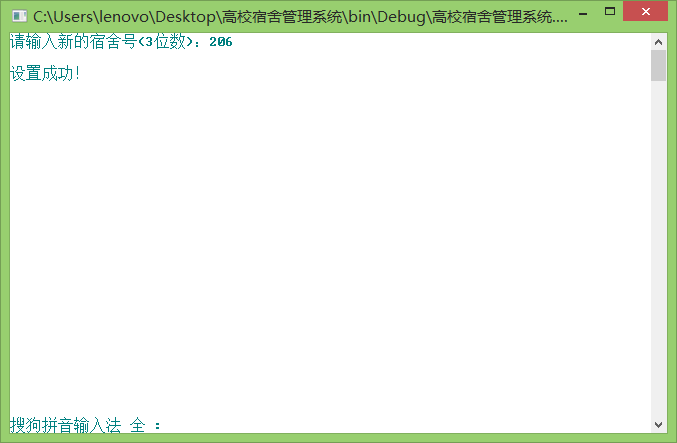




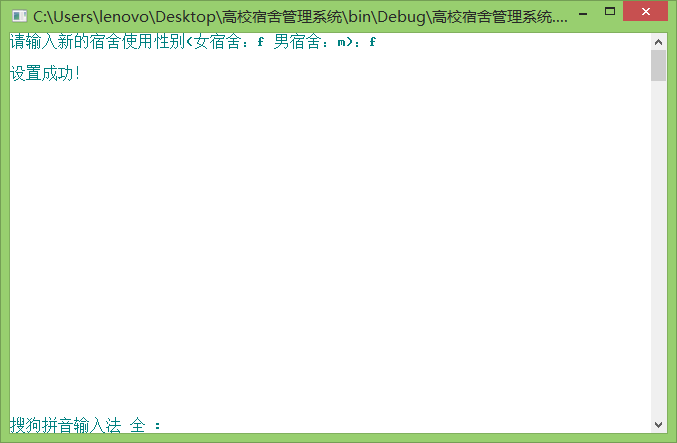
* 1. 修改宿舍信息菜单：



* 1. 修改宿舍号：



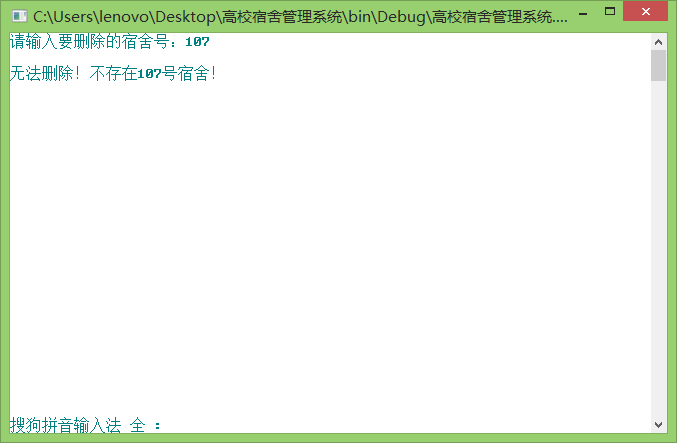
* 1. 修改宿舍使用性别：



1. 删除宿舍：
   1. 删除宿舍：

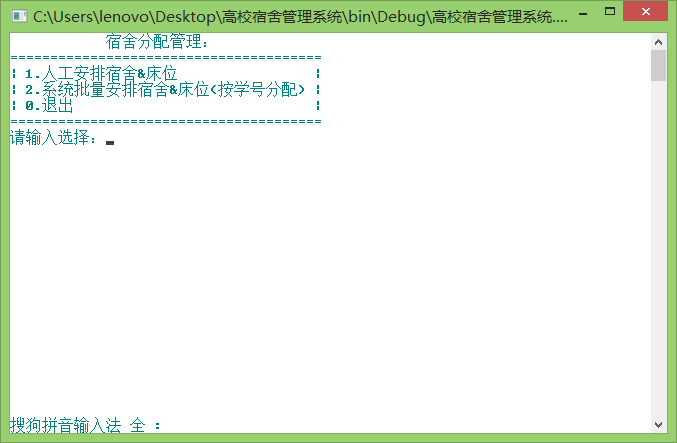


* 1. 删除宿舍失败情况：宿舍不存在以及宿舍不为空



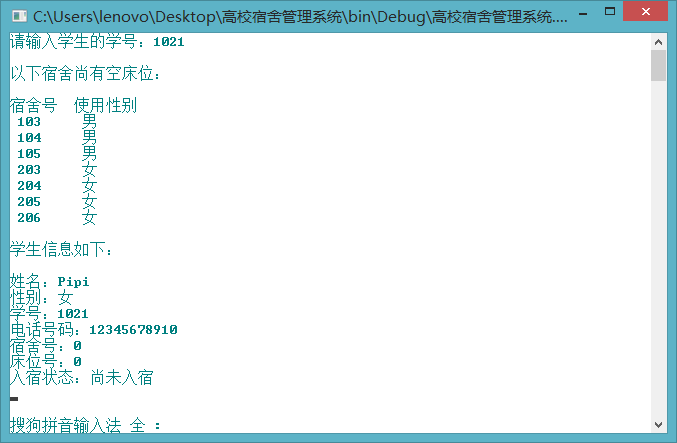


1. 宿舍分配管理：
2. 宿舍分配管理菜单：

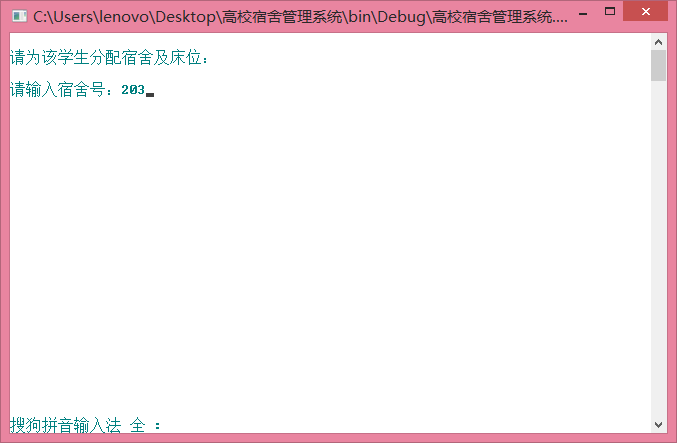


1. 人工安排宿舍＆床位：
   1. 人工安排宿舍＆床位：

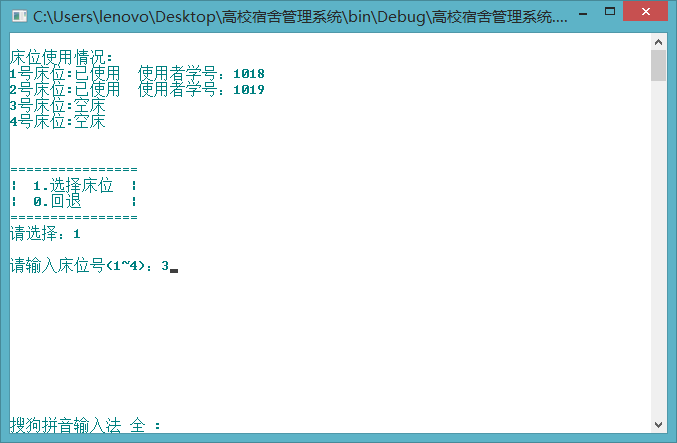
先选择学生，同时打印出学生信息和有空床位的宿舍信息。



选择宿舍。



打印出所选宿舍的床位信息，再进行床位选择。

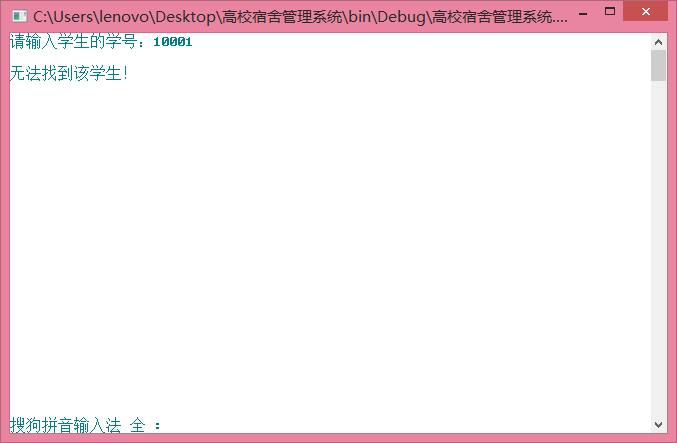


打印出新的学生信息。

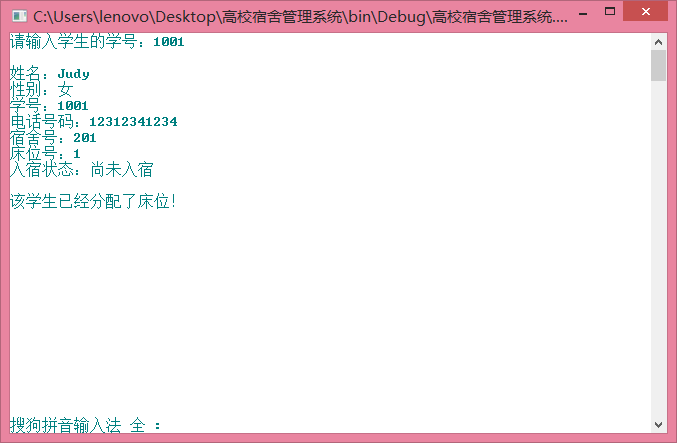


* 1. 人工安排宿舍＆床位失败情况：

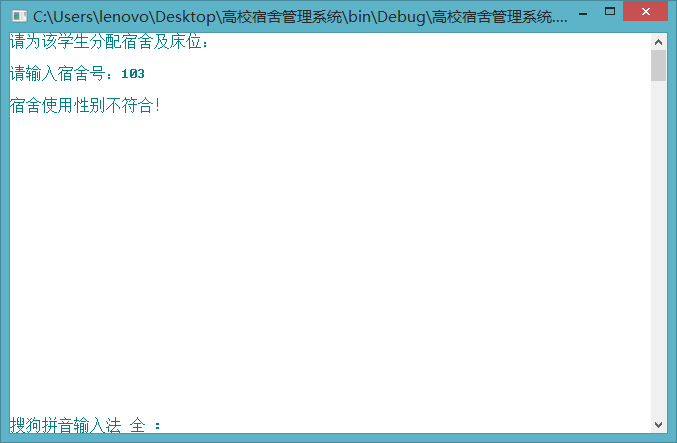
学号不存在的情况。



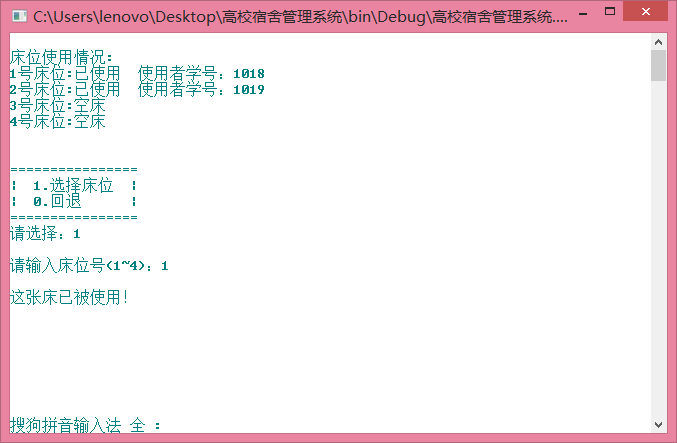
学生已经分配过床位的情况。



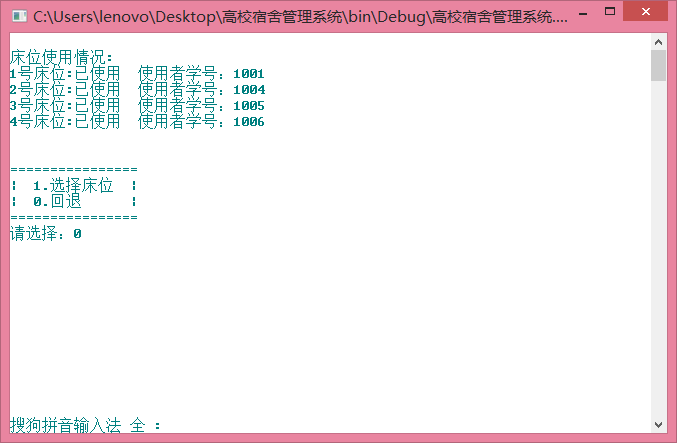
选择宿舍与学生的性别不符情况。



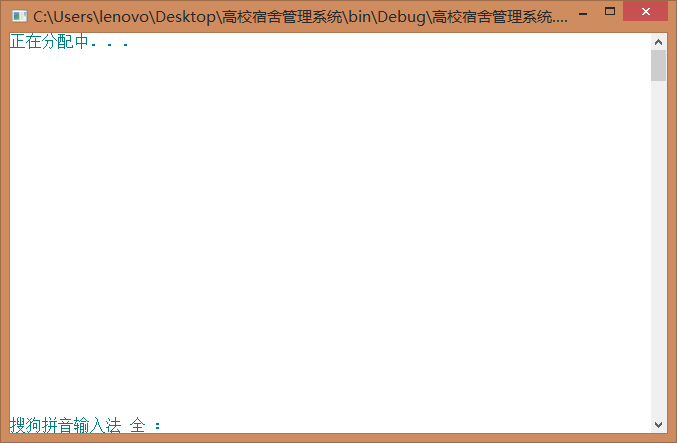
选择的床位已有学生的情况。



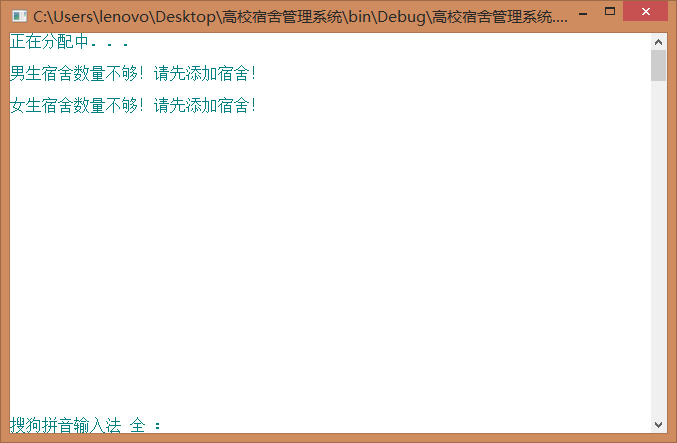
宿舍无空床位的情况。



1. 系统批量安排宿舍&床位：
2. 系统批量安排宿舍&床位：



1. 系统批量安排宿舍&床位失败情况：宿舍数量不足



1. 入宿登记：
   1. 入宿登记：

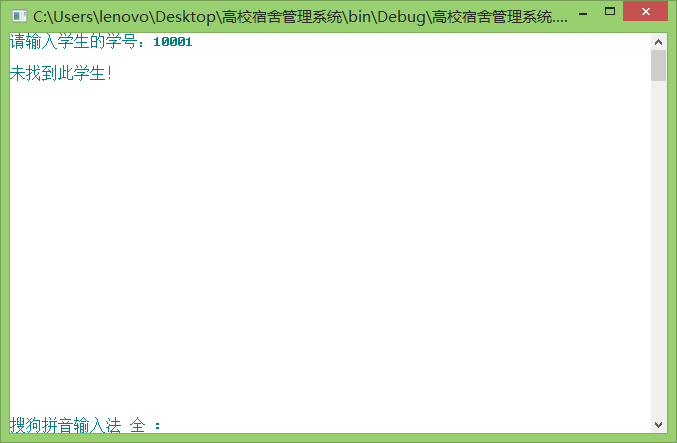


* 1. 入宿登记失败情况：

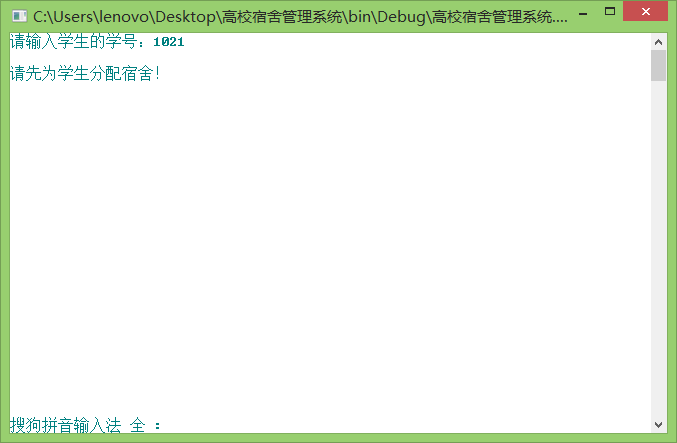
无法重复登记入宿。



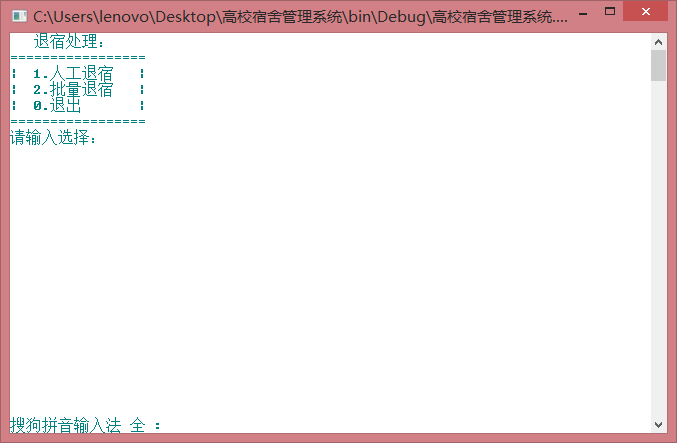
学号出错。



学生尚未分配宿舍。



1. 退宿处理：
2. 退宿处理菜单：

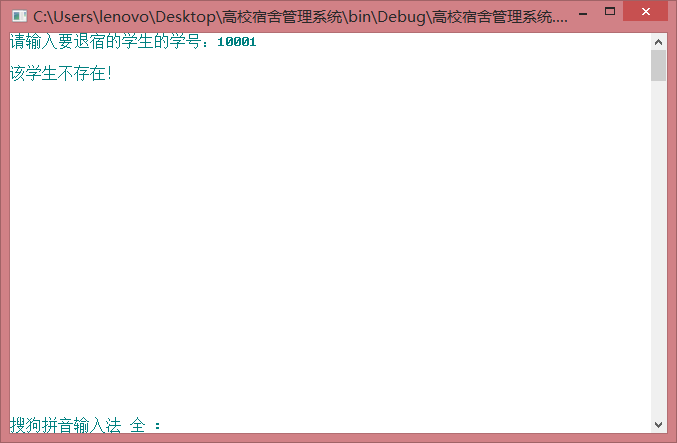


1. 人工退宿：
   1. 人工退宿：

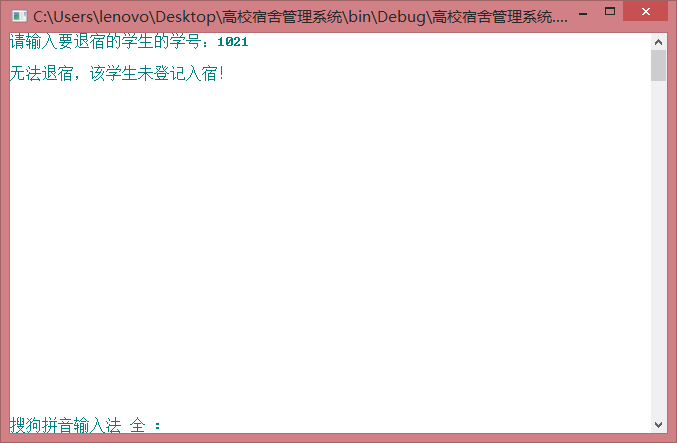


* 1. 人工退宿失败情况：

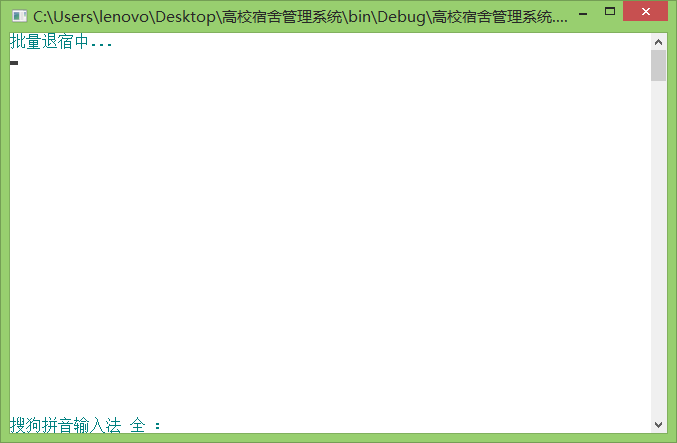
学号出错。



学生尚未入宿。



1. 批量退宿：

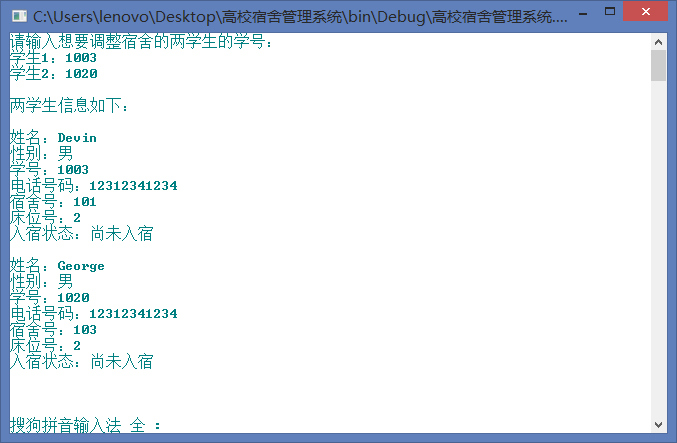


1. 调宿处理：
2. 调宿处理菜单：

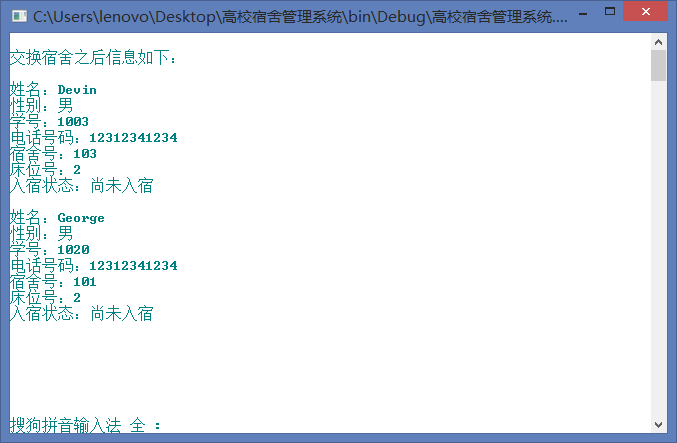


1. 两学生交换宿舍：
   1. 两学生交换宿舍：

输入学生学号后打印出两名学生的信息。

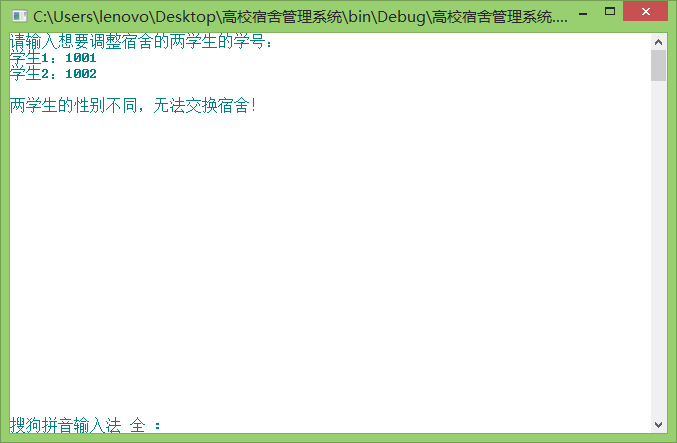


打印出交换宿舍后的两学生的信息。

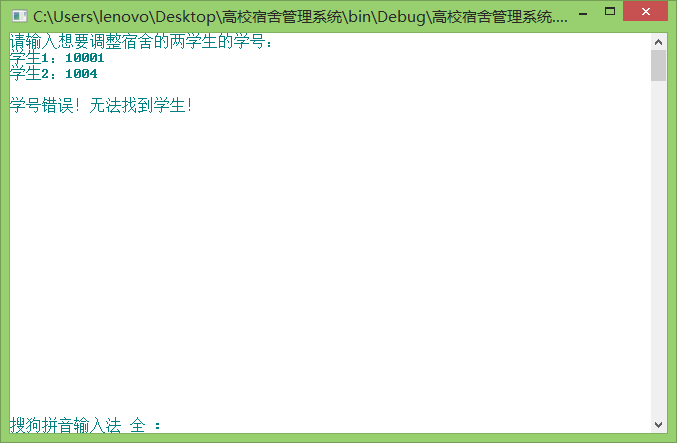


* 1. 两学生交换宿舍失败情况：

两学生的性别不同，无法交换宿舍。

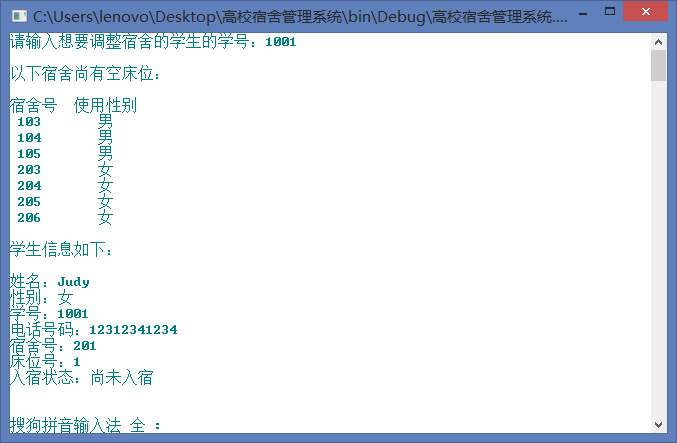


学号输入错误。

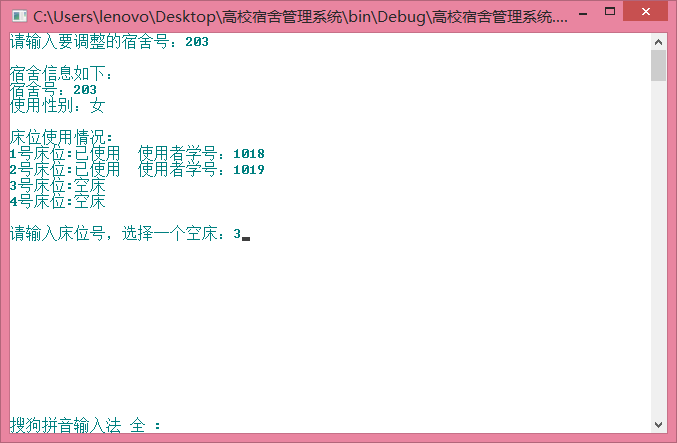


1. 调整一名学生的宿舍：
   1. 调整一名学生的宿舍：

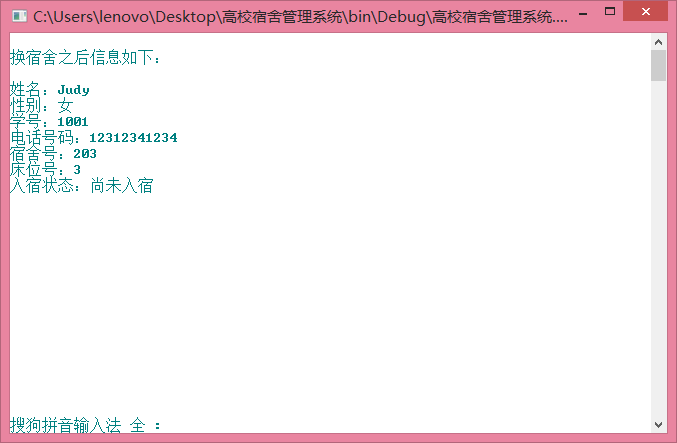
输入学生学号并打印出空宿舍和学生信息。



输入宿舍号并选择床位。



打印出调换宿舍后的学生信息。

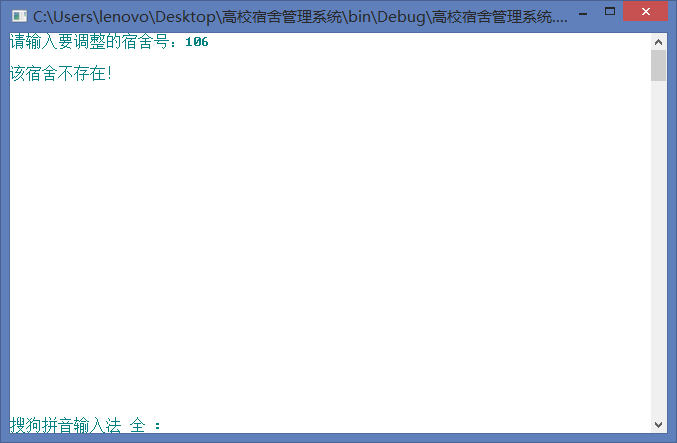


* 1. 调整一名学生的宿舍失败情况：

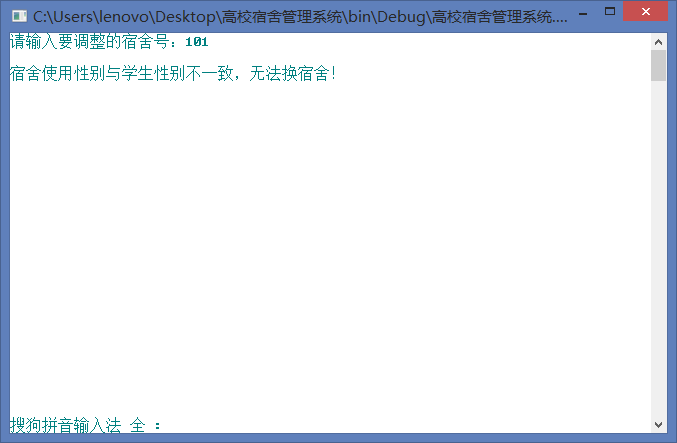
学号输入错误的情况。



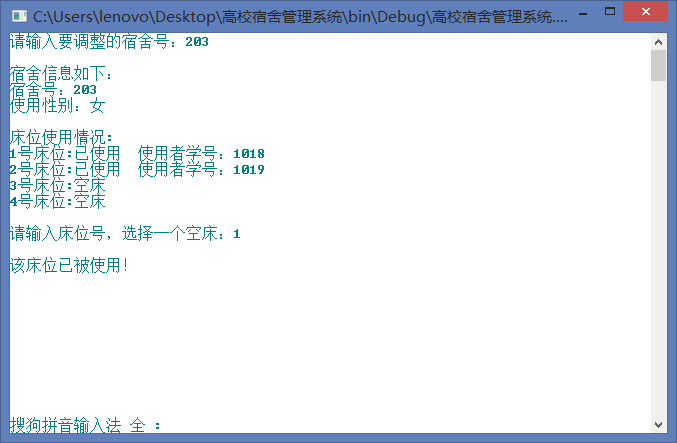
输入宿舍号出错的情况。



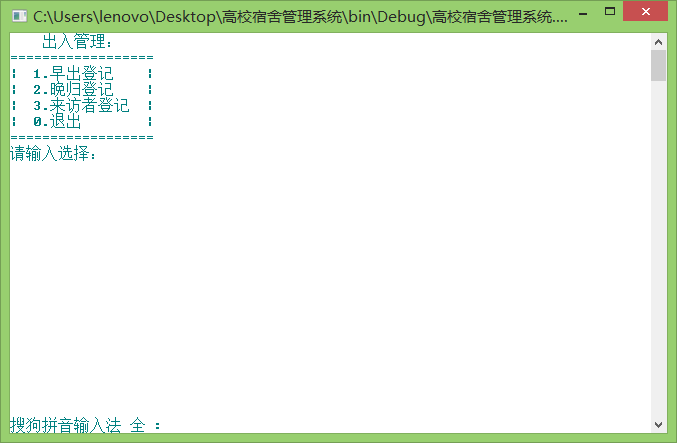
选择的宿舍与学生的性别不一致的情况。



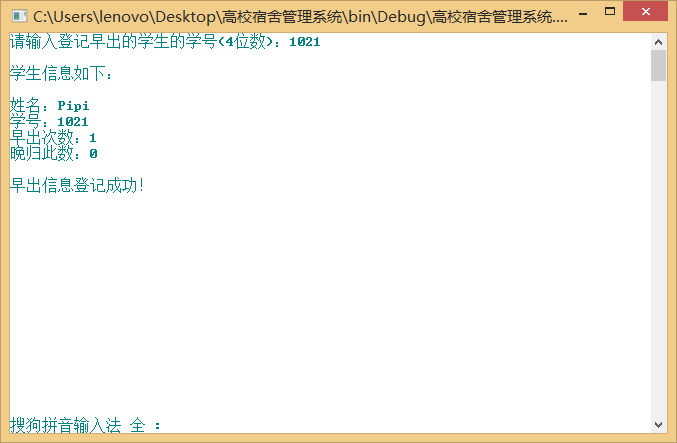
选择的床位不为空的情况。



1. **出入管理**
2. 出入管理菜单：

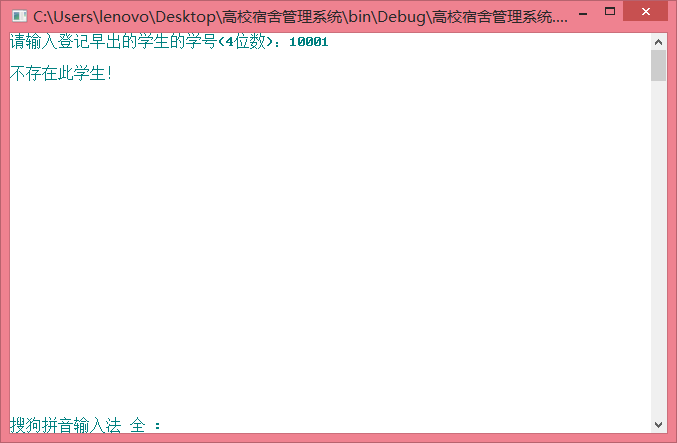


1. 早出登记：
   1. 早出登记：

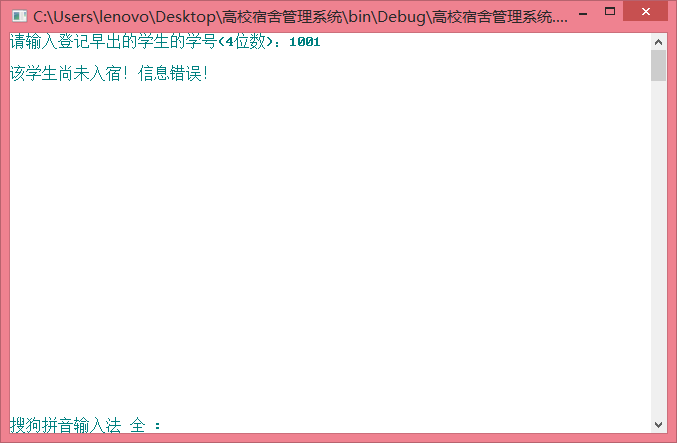


* 1. 早出登记失败情况：

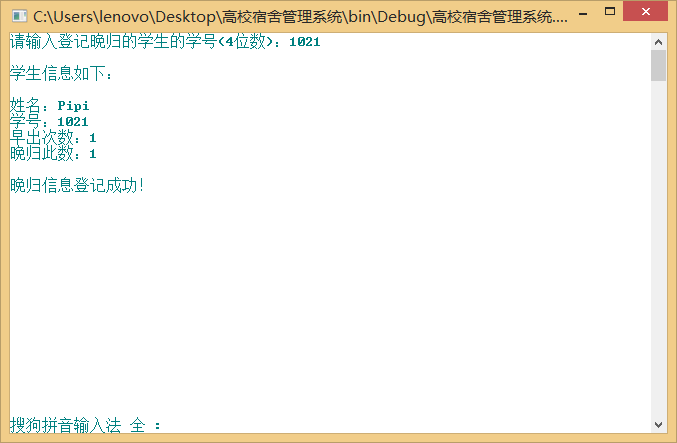
学号输入出错的情况。



学生尚未入宿的情况。

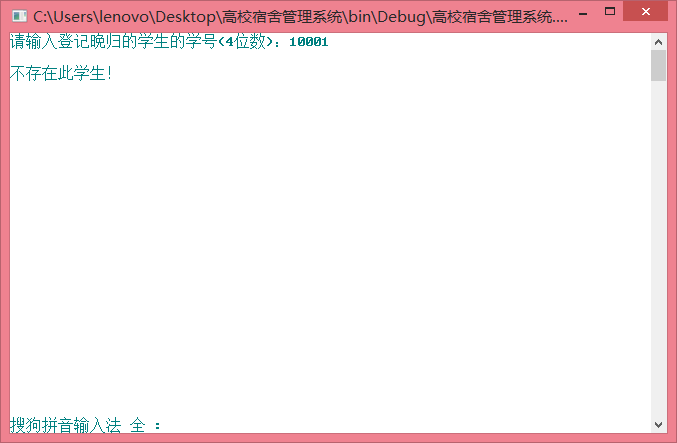


1. 晚归登记：
   1. 晚归登记：

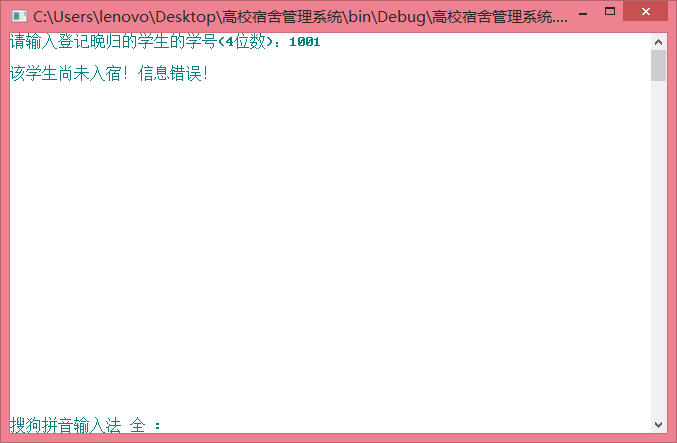


* 1. 晚归登记失败情况：

学号输入错误的情况。



学生尚未入宿的情况。



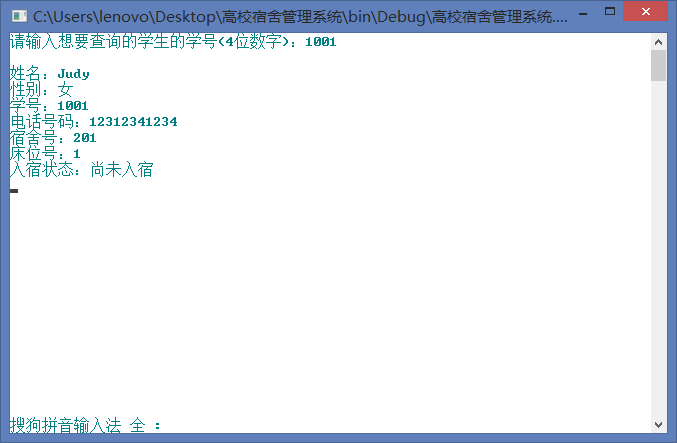
1. 来访者登记：含出错情况



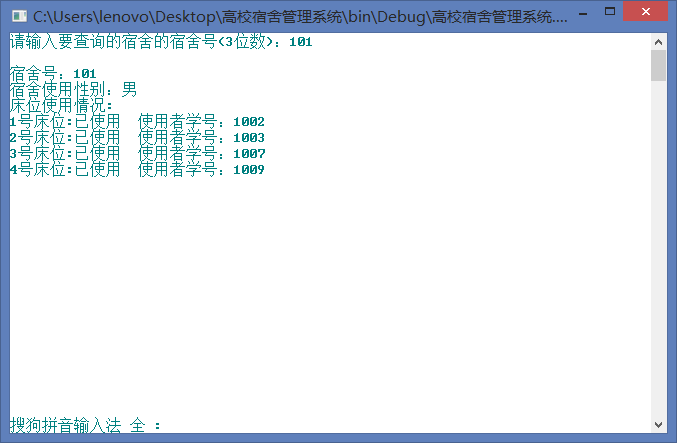
1. **信息查询**
2. 信息查询菜单：



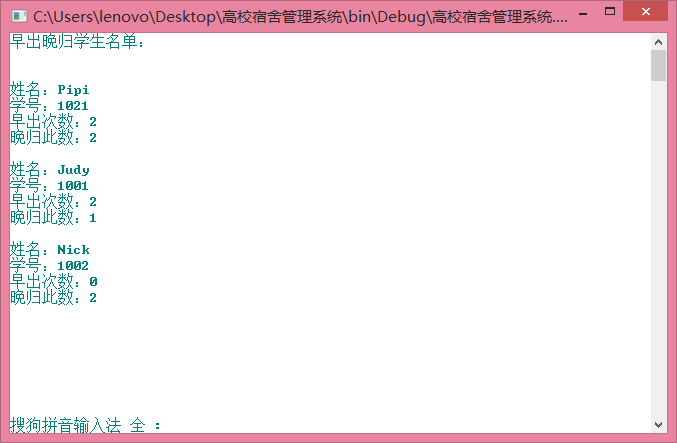
1. 查询学生信息含住宿信息：



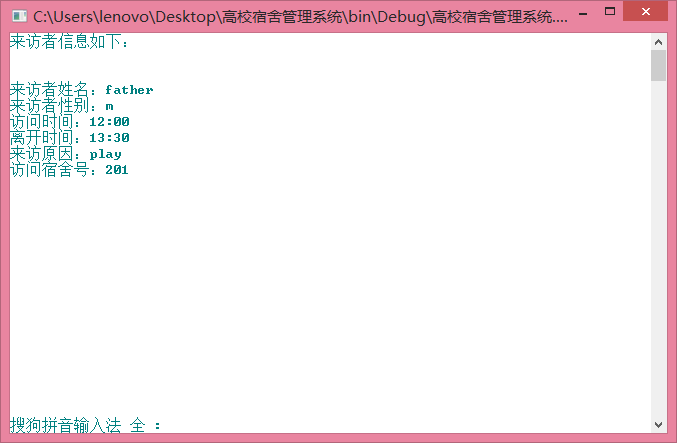
1. 查询宿舍信息含住宿信息：



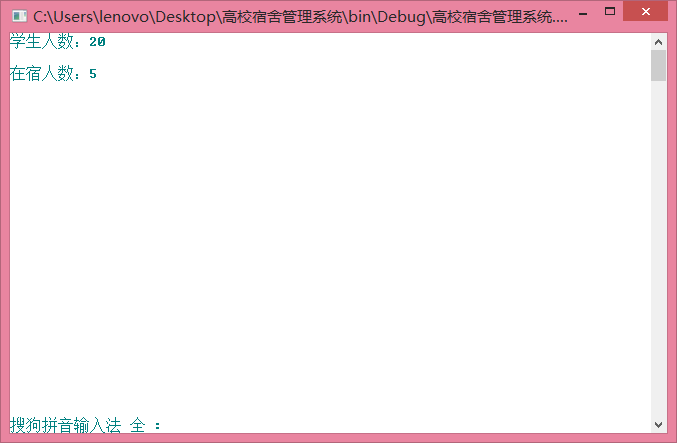
1. 查询早出晚归信息：



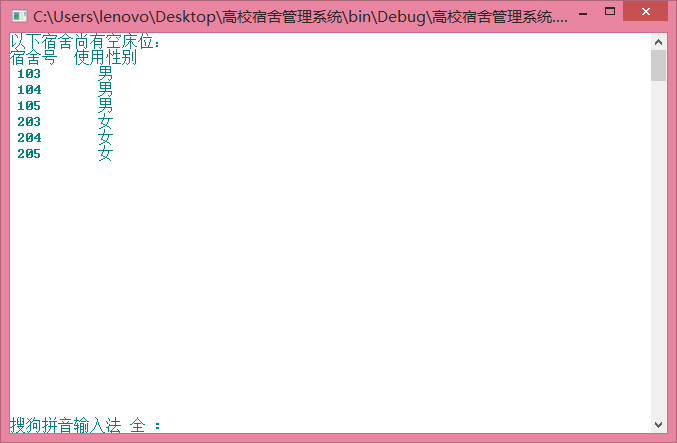
1. 查询来访者信息：



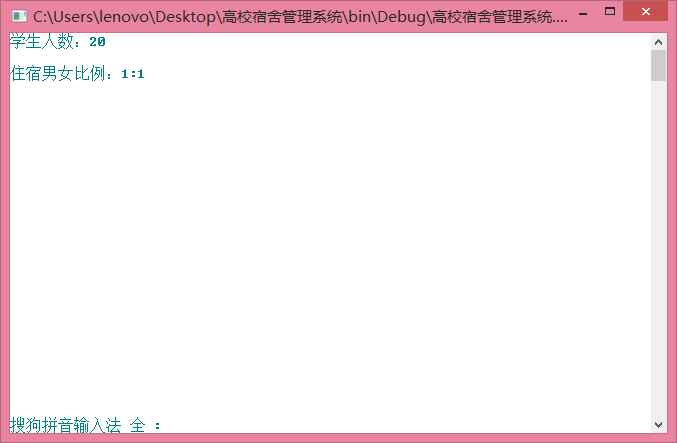
1. 查询入宿人数：



1. 查询宿舍空床情况：



1. 查询住宿男女比例：



**第五章 总结**

在做这次项目之前，我认为自己并没有完整做出一个项目的能力。起步总是比较艰难，一开始靠着阅读项目“小小图书馆”的代码去理解其功能的划分，但两个项目毕竟不同，难免多走弯路，项目也一再推翻重来。但通过多次的考虑学习，和同学相互讨论，慢慢去熟悉功能，琢磨各个模块的联系，最终项目也就得以完成了。

通过这次做项目的经验，我觉得自身学到了很多的东西。

这次的项目经验让我学到将一个项目划分模块的经验，真正去考虑到每个模块的功能的相互联系，懂得如何将各个模块联系起来。也懂得了如何去把一个类给建立起来，去考虑其中的各个要素。

这次的项目也使得我对文件的操作变得熟络起来。读写文件过去曾是困扰我的一大难题，在这次项目中，多次需要对文件进行操作，我也因此得以学习到使用文件的精髓，将数据保留下来。

我亦学会了逐个模块调试查找Bug，避免盲目调试整个程序，导致无从下手。在逐个破解Bug之后，我将整个项目的功能大致完善了。

我认识到了每一个产品的产生都是不容易的，也认识到了版权的重要性。每个人的劳动成果，都不应该被盗用而应该受到保护和尊重。

虽然这个项目不是完美的，但这个项目对于我而言是一次大的突破，是一次大的进步，增加了我对于项目的认识，也提高了我的兴趣，使我的考虑问题更加全面，并且相信自己做项目的能力。

学海无涯，日后我也会继续学习，争取做到更好。

**参考文献**

1.谭浩强编著.《C++程序设计（第二版）》.北京：清华大学出版社，2011.8

2.宋桂琴编著.《面型对象程序设计实验课程指导》，2008.8

**附录**

**程序代码：**

///////////////////////////////////////////////

//

//项目名称：高校宿舍管理系统的

//作者：Pipi——詹萍

//完成日期：2016.10.15

//

////////////////////////////////////////////////

* + 1. **(.h)头文件**

1. **administrator.h文件：**

#ifndef ADMINISTRATOR\_H\_INCLUDED

#define ADMINISTRATOR\_H\_INCLUDED

//////////////////////////////////////////////////////

//

//说明：类Administrator的函数成员的函数体在administrator.cpp中

//

//////////////////////////////////////////////////////

class Administrator

{

private:

char userName[20];

char password[20];

public:

char \*getUserName(); //获取用户名

char \*getPassword(); //获取密码

void addAdministrator(char \*na,char \*pass);//增加管理员

};

#endif // ADMINISTRATOR\_H\_INCLUDED

1. **student.h文件：**

#ifndef STUDENT\_H\_INCLUDED

#define STUDENT\_H\_INCLUDED

//////////////////////////////////////////////////////

//

//说明：类Student的函数成员的函数体在student.cpp中

//

//////////////////////////////////////////////////////

class Student

{

private:

char name[10]; //姓名

char sex; //性别

int number; //学号

char phone[15]; //电话号码

int roomNumber; //宿舍号

int bedNumber; //床位号

bool mark; //入宿标记（0：离宿状态 1：入宿状态）

bool note; //删除标记（0：已删 1：未删）

public:

Student(); //构造函数

char \*getname(); //获得姓名

char getsex(); //获得性别

int getnumber(); //获得学号

char \*getphone(); //获得电话号码

int getroomNumber(); //获得宿舍号

int getbedNumber(); //获得床位号

bool getmark(); //获得入宿状态

bool getnote(); //获得删除标记

void addstudent(char \*na,char s,int num,char \*ph);//加入新学生，登记信息

void setname(char \*na); //设置姓名

void setphone(char \*ph); //设置电话号码

void setsex(char s); //设置性别

void setroom\_bed(int room,int bed); //设置宿舍号和床位号

void delstudent(); //删除学生

void show(); //展示学生信息

void live(); //入宿

void leave(); //退宿

};

#endif // STUDENT\_H\_INCLUDED

1. **dormitory.h文件：**

#ifndef DORMITORY\_H\_INCLUDED

#define DORMITORY\_H\_INCLUDED

//////////////////////////////////////////////////////

//

//说明：类Dormitory的函数成员的函数体在dormitory.cpp中

//

//////////////////////////////////////////////////////

class Dormitory

{

private:

int dormitoryNumber; //宿舍号

int bedNum[4]; //每个床的床位号

bool bed[4]; //床位使用情况（0：空床 1：已使用）

char dormitorySex; //女宿舍：f 男宿舍：m

int beduser[4]; //每个床位的使用者的学号

bool delNote; //删除标记（0：删除 1：未删）

public:

Dormitory(); //构造函数

int getdormitoryNumber(); //获得宿舍号

bool getdelNote(); //获得删除标记

char getdormitorySex(); //获得宿舍的使用性别

int use\_dormitory(); //获得宿舍的使用情况

bool bed\_use\_case(int bedNum); //获得具体某个床位的使用情况

void showbed(); //展示全部床位的使用情况

void setdormitoryNumber(int dorNum);//设置宿舍号

void setdormitorySex(char sex); //设置宿舍使用性别

void deletedormitory(); //删除宿舍

void not\_use\_bed(int bedNum); //空出床位

void use\_bed(int bedNum,int num); //使用床位

void addDormitory(int num,char sex);//增加宿舍

};

#endif // DORMITORY\_H\_INCLUDED

1. **record.h文件：**

#ifndef RECORD\_H\_INCLUDED

#define RECORD\_H\_INCLUDED

//////////////////////////////////////////////////////

//

//说明：类Morning\_Evening\_record的函数成员的函数体在record.cpp中

//

//////////////////////////////////////////////////////

//学生早出晚归登记

class Morning\_Evening\_record

{

private:

char studentName[10];

int studentNumber;

int morningTimes; //早出次数

int eveningTimes; //晚归此数

public:

Morning\_Evening\_record(); //构造函数

int getNum(); //获取学号

void morningRecord(); //学生早出次数增加

void eveningRecord(); //学生晚归次数增加

void addStud(char \*na,int nu); //添加早出晚归的学生

void showStudentInformation(); //展示学生早出晚归信息

};

//////////////////////////////////////////////////////

//

//说明：类Visitor的函数成员的函数体在record.cpp中

//

//////////////////////////////////////////////////////

//来访者登记

class Visitor

{

private:

char visitorName[10];

char visitorSex;

char visitTime[8];

char leaveTime[8];

char reason[50];

int visitDorNum;

public:

void setVisitInformation(char \*na,char se,char \*v\_t,char \*l\_t,char \*re,int dorNum);//设置信息

void showVisitorInformation(); //展示来访信息

};

#endif // RECORD\_H\_INCLUDED

1. **function.h文件**：

#ifndef FUNCTION\_H\_INCLUDED

#define FUNCTION\_H\_INCLUDED

//////////////////////////////////////////////////////

//

//说明：类Function的函数成员的函数体在以下文件中:

//1.adjust\_dormitory.cpp 2.allot\_dormitory.cpp 3.construction.cpp 4.destructor.cpp

//5.dormitory\_IM.cpp 6.dormitory\_IM\_function.cpp 7.dormitory\_manage.cpp

//8.get\_in\_out\_manage.cpp 9.leave\_dormitory.cpp 10.live\_in\_dormitory.cpp

//11.login.cpp 12.query\_information.cpp 13.student\_IM.cpp 14.student\_IM\_function.cpp

//

//////////////////////////////////////////////////////

#include "administrator.h"

#include "student.h"

#include "dormitory.h"

#include "record.h"

using namespace std;

const int Maxa=20; //最多20位管理员

const int Maxs=500; //最多500名学生

const int Maxd=200; //最多200间宿舍

const int Maxv=100; //最多登记100个来访者

const int Max=500; //最多登记500个早出晚归信息

class Function

{

private:

//登录管理

Administrator administrator[Maxa];

int adNum;

//学生管理

Student student[Maxs];

int stuNum;

Student \*ps;

//宿舍管理

Dormitory dormitory[Maxd];

int dorNum;

Dormitory \*pd;

//出入管理

Visitor visitor[Maxv];

int visNum;

Morning\_Evening\_record stud[Max];

int studNum;

Morning\_Evening\_record \*pr;

public:

//Function()在文件construction.cpp中

Function(); //构造函数

//~Function()在文件destructor.cpp中

~Function(); //析构函数

//登录管理

//以下函数成员在文件login.cpp中

int queryUserName(char \*na); //查询用户名

int queryUser(char \*na,char \*pass); //查询用户

void Register(); //注册账号

int sign\_in(); //登录

int login(); //登录管理

//学生基本信息管理

//以下函数在文件student\_IM\_function.cpp中

void addstudent(char \*na,char s,int num,char \*ph);//增加学生

bool deletestudent(int num); //删除学生

void modify(Student \*p); //修改学生信息

Student \*queryNum(int num); //查找学生

//void studentdatabase()在文件student\_IM.cpp中

void studentdatabase(); //学生基本信息管理

//住宿管理

//以下函数在文件dormitory\_IM\_function.cpp中

void addDormitory(int num,char sex); //增加宿舍

void modifyDormitory(Dormitory \*p); //修改宿舍信息

void deleteDormitory(int room); //删除宿舍

Dormitory \*queryroom(int num); //查询宿舍信息

//void allotDormitory()在文件allot\_dormitory.cpp中

void allotDormitory(); //分配宿舍

//void liveDormitory()在文件live\_in\_dormitory.cpp中

void liveDormitory(); //入宿登记

//void leaveDormitory()在文件leave\_dormitory.cpp中

void leaveDormitory(); //退宿处理

//void adjustDormitory()在文件adjust\_dormitory.cpp中

void adjustDormitory(); //调宿处理

//void dormitorydatabase()在文件dormitory\_IM.cpp中

void dormitorydatabase(); //宿舍基本信息管理

//void dormitoryManage()在文件dormitory\_manage.cpp中

void dormitoryManage(); //住宿管理

//出入管理

//以下函数在文件get\_in\_out\_manage.cpp中

Morning\_Evening\_record \*query(int num); //查找早出晚归学生

void morning\_go\_out(); //早出登记

void evening\_go\_back(); //晚归登记

void visit\_record(); //来访者登记

void get\_in\_out\_manage(); //出入管理

//信息查询

//void query\_information()在文件query\_information.cpp中

void query\_information(); //信息查询

};

#endif // FUNCTION\_H\_INCLUDED

* + 1. **(.cpp)文件**

1. **main.cpp文件：**

////////////////////////////////////////////////////////

//

//文件名：main.cpp

//程序功能：

// 主函数，含初始界面登录入口和高校宿舍管理系统总菜单

//作者：Pipi

//日期：2016.10.15

//

////////////////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string.h>

#include <stdlib.h> //system函数头文件

#include <windows.h> //Sleep函数头文件

#include "administrator.h"

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

int main()

{

system("color F3"); //修改界面颜色

int choice=1;

Function function\_select; //定义一个学生数据库类的对象

//初始界面

cout<<"\n\n\n\n";

cout<<"--------------------------------------------------------------------------------"<<endl;

cout<<" D D OO R R M M I I I T T T OO R R Y Y"<<endl;

cout<<" D D O O R R M M M M I T O O R R Y Y "<<endl;

cout<<" D D O O R R M M M M I T O O R R Y "<<endl;

cout<<" D D O O R R M M M I T O O R R Y "<<endl;

cout<<" D D OO R R M M I I I T OO R R Y "<<endl;

cout<<"\n--------------------------------------------------------------------------------\n"<<endl;

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*正在进入“高校宿舍管理系统”\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\n--------------------------------------------------------------------------------"<<endl;

Sleep(2000); //停留时间单位是毫秒

system("cls"); //页面清除

//管理员登录、注册模块

if(function\_select.login()==0)

{

cout<<"\n正在退出系统..."<<endl;

exit(1);

}

//功能选择模块

while(choice!=0)

{

cout<<" 高校宿舍管理系统："<<endl;

cout<<"------------------------"<<endl;

cout<<"| 1.学生基本信息管理 |"<<endl;

cout<<"| 2.住宿管理 |"<<endl;

cout<<"| 3.出入管理 |"<<endl;

cout<<"| 4.信息查询 |"<<endl;

cout<<"| 0.退出系统 |"<<endl;

cout<<"------------------------"<<endl;

cout<<"请输入选择：";

cin>>choice;

system("cls");

if(choice<=4&&choice>=0)

{

switch(choice)

{

case 1://学生基本信息管理

function\_select.studentdatabase();

break;

case 2://住宿管理

function\_select.dormitoryManage();

break;

case 3://出入管理

function\_select.get\_in\_out\_manage();

break;

case 4://信息查询

function\_select.query\_information();

break;

case 0://退出系统

cout<<"正在退出系统..."<<endl;

Sleep(1000);

exit(1);

break;

}

}

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

return 0;

}

1. **administrator.cpp文件：**

//////////////////////////////////////////////

//

//文件名：administrator.cpp

//程序功能：管理员类的成员函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <string.h>

#include <conio.h> //getch函数头文件

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "administrator.h"

using namespace std;

/////////////////////////////////////////////

//获取用户名函数

//函数功能：获取用户名

//函数参数：无

//参数返回值：字符串指针，指向用户名

//

char \*Administrator::getUserName()

{return userName;}

/////////////////////////////////////////////

//获取密码函数

//函数功能：获取密码

//函数参数：无

//参数返回值：字符串指针，指向密码

//

char \*Administrator::getPassword()

{return password;}

/////////////////////////////////////////////

//增加管理员函数

//函数功能：增加管理员

//函数参数：字符指针，以用户名和密码作为参数来新增管理员

//参数返回值：无

//

void Administrator::addAdministrator(char \*na,char \*pass)

{

strcpy(userName,na);

strcpy(password,pass);

}

1. **login.cpp文件：**

//////////////////////////////////////////////

//

//文件名：login.cpp

//程序功能：登录管理模块

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <string.h>

#include <conio.h> //getch函数头文件

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "administrator.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//查询函数

//函数功能：查询用户名

//函数参数：字符型指针，指向要查找的用户名

//参数返回值：整型，1：查找成功 0：查找失败

//

int Function::queryUserName(char \*na)

{

for(int i=0;i<=adNum;i++)

{

//找到该用户名

if(strcmp(administrator[i].getUserName(),na)==0)

return 1;

}

return 0;

}

/////////////////////////////////////////////

//查询函数

//函数功能：查询用户

//函数参数：1.字符型指针\*na，指向要查找的用户名

// 2.字符型指针\*pass,指向要查找的用户密码

//参数返回值：整型，1：查找成功 0：查找失败

//

int Function::queryUser(char \*na,char \*pass)

{

for(int i=0;i<=adNum;i++)

{

//找到该用户名和密码

if(strcmp(administrator[i].getUserName(),na)==0&&strcmp(administrator[i].getPassword(),pass)==0)

return 1;

//用户名正确,密码错误

if(strcmp(administrator[i].getUserName(),na)==0&&strcmp(administrator[i].getPassword(),pass)!=0)

return 2;

}

return 0;

}

/////////////////////////////////////////////

//注册函数

//函数功能：管理员注册账号

//函数参数：无

//参数返回值：无

//

void Function::Register()

{

int temp;

char Name[20];

char Password[20];

while(1)

{

cout<<"用户名(20个字符以内)：";

cin>>Name;

temp=queryUserName(Name);

if(temp==1)

{

cout<<"\n此用户名已存在！"<<endl;

Sleep(1000);

system("cls");

}

else break;

}

cout<<"密 码(20个字符以内)：";

cin>>Password;

//注册账号

adNum++;

administrator[adNum].addAdministrator(Name,Password);

//验证是否注册成功

temp=queryUser(Name,Password);

if(temp==1)

{

cout<<"\n注册成功！"<<endl;

Sleep(1000);

system("cls");

}

else

{

cout<<"\n注册失败！"<<endl;

Sleep(1000);

system("cls");

}

}

/////////////////////////////////////////////

//登录函数

//函数功能：管理员登录

//函数参数：无

//参数返回值：整型，1.登录成功，0：登录失败

//

int Function::sign\_in()

{

int index=0; //用于将密码显示为\*号

int temp;

char Name[20];

char Password[20];

cout<<"用户名(20个字符以内)：";

cin>>Name;

cout<<"密 码(20个字符以内)：";

//以下代码块将输入的密码显示为\*号

while(1)

{

char ch;

ch=getch();

if(ch==8)//退格键'\b'

{

if(index!=0)

{

cout<<char(8)<<" "<<char(8);

index--;

}

}

else if(ch==13) //回车键'\r'

{

Password[index]='\0';

cout<<endl;

break;

}

else

{

cout<<"\*";

Password[index++]=ch;

}

}//密码星号end

//登录操作

temp=queryUser(Name,Password);

//验证是否登录成功

if(temp==1)

{

cout<<"\n登陆成功！"<<endl;

Sleep(1000);

system("cls");

return 1;

}

if(temp==2)

{

cout<<"\n密码错误！"<<endl;

Sleep(1000);

system("cls");

}

if(temp==0)

{

cout<<"\n用户名不存在！"<<endl;

Sleep(1000);

system("cls");

}

return 0;

}

/////////////////////////////////////////////

//登录管理函数

//函数功能：登录管理

// 1.登录

// 2.注册

// 0.退出系统

//函数参数：无

//参数返回值：整型，1.登录成功，0：退出系统

//

int Function::login()

{

int choice=1;

while(1)

{

cout<<"欢迎使用“高校宿舍管理系统”："<<endl;

cout<<"==========================="<<endl;

cout<<"| 1.登录 |"<<endl;

cout<<"| 2.注册 |"<<endl;

cout<<"| 0.退出系统 |"<<endl;

cout<<"==========================="<<endl;

cout<<"请选择：";

cin>>choice;

system("cls");

if(choice<=2&&choice>=0)

{

switch (choice)

{

case 1://登录

if(sign\_in()==1)return 1;

break;

case 2://注册

Register();

break;

case 0://退出系统

return 0;

break;

}//end switch

}//end if

else

{

cout<<"\n错误！没有此选项！"<<endl;

Sleep(500);

}

system("cls");

}//end while

}

1. **student.cpp文件：**

//////////////////////////////////////////////

//

//文件名：student.cpp

//程序功能：Student类的成员函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <string.h>

#include <iomanip>

#include "student.h"

using namespace std;

/////////////////////////////////////////////

//构造函数

//函数功能：初始化宿舍、床位信息为空

//函数参数：无

//参数返回值：无

//

Student::Student()

{

roomNumber=0;

bedNumber=0;

}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生姓名

//函数参数：无

//参数返回值：字符型指针，返回学生姓名的地址

//

char \*Student::getname()

{return name;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生性别

//函数参数：无

//参数返回值：字符型，返回学生性别

//

char Student::getsex()

{return sex;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生学号

//函数参数：无

//参数返回值：整型,返回学生学号

//

int Student::getnumber()

{return number;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生电话号码

//函数参数：无

//参数返回值：字符型指针,返回学生电话号码的地址

//

char \*Student::getphone()

{return phone;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生宿舍号

//函数参数：无

//参数返回值：整型,返回学生宿舍号

//

int Student::getroomNumber()

{return roomNumber;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生床位号

//函数参数：无

//参数返回值：整型,返回学生床位号

//

int Student::getbedNumber()

{return bedNumber;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生入宿状态

//函数参数：无

//参数返回值：布尔型，0：离宿状态 1：入宿状态

//

bool Student::getmark()

{return mark;}

/////////////////////////////////////////////

//get函数

//函数功能：获得学生删除标记

//函数参数：无

//参数返回值：布尔型，0：已删 1：未删

//

bool Student::getnote()

{return note;}

/////////////////////////////////////////////

//添加函数

//函数功能：加入新学生，登记信息，将删除标记记为1，表示未删

//函数参数：字符型指针、字符型，整型。分别代表学生各类信息

//参数返回值：无

//

void Student::addstudent(char \*na,char s,int num,char \*ph)

{

strcpy(name,na);

sex=s;

number=num;

strcpy(phone,ph);

note=1; //删除标记（0：已删 1：未删）

}

/////////////////////////////////////////////

//set函数

//函数功能：设置姓名

//函数参数：字符型指针，学生姓名

//参数返回值：无

//

void Student::setname(char \*na)

{strcpy(name,na);}

/////////////////////////////////////////////

//set函数

//函数功能：设置电话号码

//函数参数：字符型指针，学生电话号码

//参数返回值：无

//

void Student::setphone(char \*ph)

{strcpy(phone,ph);}

/////////////////////////////////////////////

//set函数

//函数功能：设置性别

//函数参数：字符型，学生性别

//参数返回值：无

//

void Student::setsex(char s)

{sex=s;}

/////////////////////////////////////////////

//删除函数

//函数功能：删除学生，note=0代表学生被删除

//函数参数：无

//参数返回值：无

//

void Student::delstudent()

{note=0;}

/////////////////////////////////////////////

//show函数

//函数功能：展示学生信息

//函数参数：无

//参数返回值：无

//

void Student::show()

{

cout<<"\n姓名："<<name<<endl;

cout<<"性别：";

if(sex=='f')cout<<"女"<<endl;

else cout<<"男"<<endl;

cout<<"学号："<<number<<endl;

cout<<"电话号码："<<phone<<endl;

cout<<"宿舍号："<<roomNumber<<endl;

cout<<"床位号："<<bedNumber<<endl;

cout<<"入宿状态：";

if(mark==0)cout<<"尚未入宿"<<endl;

else cout<<"已入宿"<<endl;

}

/////////////////////////////////////////////

//set函数

//函数功能：设置宿舍号和床位号

//函数参数：整型，宿舍号room和床位号bed

//参数返回值：无

//

void Student::setroom\_bed(int room,int bed)

{

roomNumber=room;

bedNumber=bed;

}

/////////////////////////////////////////////

//入宿函数

//函数功能：入宿，0：离宿状态 1：入宿状态

//函数参数：无

//参数返回值：无

//

void Student::live()

{mark=1;}

/////////////////////////////////////////////

//退宿函数

//函数功能：退宿，0：离宿状态 1：入宿状态

//函数参数：无

//参数返回值：无

//

void Student::leave()

{mark=0;}

1. **student\_IM\_function.cpp文件：**

//////////////////////////////////////////////

//

//文件名：student\_IM\_function.cpp

//程序功能：学生基本信息增删查改

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <string.h>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

#include "record.h"

using namespace std;

//学生信息管理的模块

/////////////////////////////////////////////

//查找函数

//函数功能：按学号查找学生

//函数参数：整型，num为学生学号

//参数返回值：Student型指针，返回查找到的学生对象的地址

//

Student \*Function::queryNum(int num)

{

for(int i=0;i<=stuNum;i++)

{

if(num==student[i].getnumber()&&student[i].getnote()==1)//宿舍号相同且未被删除(0：删除 1：未删除)

return &student[i];

}

return NULL;

}

/////////////////////////////////////////////

//添加函数

//函数功能：增加学生

//函数参数：1.字符型指针，\*na指向学生姓名

// 2.字符型，s为学生性别

// 3.整型，num为学生学号

// 4.字符型指针，\*ph指向学生电话

//参数返回值：无

//

void Function::addstudent(char \*na,char s,int num,char \*ph)

{

ps=queryNum(num);

if(ps==NULL)

{

stuNum++;

student[stuNum].addstudent(na,s,num,ph);

cout<<"\n学生信息添加成功！"<<endl;

}

else cout<<"\n该学号已存在，学生信息添加失败！"<<endl;

}

/////////////////////////////////////////////

//删除函数

//函数功能：按学号删除学生

//函数参数：整型，num为学生学号

//参数返回值：布尔型，0--删除失败 1--删除成功

//

bool Function::deletestudent(int num)

{

Dormitory \*p;

ps=queryNum(num);

if(ps==NULL)

{

cout<<"\n此学生不存在！"<<endl;

return 0;

}

else if(ps->getmark()==1)

{

cout<<"\n学生已经入宿！"<<endl;

return 0;

}

else

{

if(ps->getroomNumber()!=0)

{

p=queryroom(ps->getroomNumber());

if(p==NULL)

{

cout<<"出错，无法找到学生居住的宿舍！"<<endl;

return 0;

}

//空出床位

else p->not\_use\_bed(ps->getbedNumber());

}

ps->delstudent();

return 1;

}

}

/////////////////////////////////////////////

//修改函数

//函数功能：通过指向学生对象的指针修改学生信息

//函数参数：Student型指针

//参数返回值：无

//

void Function::modify(Student \*p)

{

char name[15];

char phone[15];

char sex;

int select=1;

while(select)

{

cout<<" 信息修改："<<endl;

cout<<"===================="<<endl;

cout<<"| 1.修改姓名 |"<<endl;

cout<<"| 2.修改电话 |"<<endl;

cout<<"| 3.修改性别 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"===================="<<endl;

cout<<"请输入选择：";

cin>>select;

cout<<endl;

if(select>=0&&select<=3)

{

switch(select)

{

case 1://修改姓名

cout<<"请输入姓名：";

cin>>name;

p->setname(name);

cout<<"\n设置成功！"<<endl;

Sleep(1000);

break;

case 2://修改电话

cout<<"请输入电话号码：";

cin>>phone;

p->setphone(phone);

cout<<"\n设置成功！"<<endl;

Sleep(1000);

break;

case 3://修改性别

if(p->getmark()==1)

{

cout<<"该学生已经入宿！无法修改性别！"<<endl;

Sleep(1000);

break;

}

cout<<"请输入性别(男：m 女：f)：";

cin>>sex;

if(sex=='f'||sex=='m')

{

p->setsex(sex);

cout<<"\n设置成功！"<<endl;

Sleep(1000);

}

else

{

cout<<"\n性别输入错误！"<<endl;

Sleep(1000);

break;

}

break;

}

system("cls");

}//end if

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}

}

1. **student\_IM.cpp文件：**

//////////////////////////////////////////////

//

//文件名：student\_IM.cpp

//程序功能：学生基本信息管理

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string.h>

#include <algorithm> //sort函数

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//学生基本信息管理函数

//函数功能：学生基本信息管理

// 1.添加学生

// 2.查询学生信息

// 3.修改学生信息

// 4.删除学生

//函数参数：无

//参数返回值：无

//

void Function::studentdatabase()

{

int choice=1;

int number;

char name[15];

char sex;

char phone[15];

while(choice!=0)

{

cout<<" 学生基本信息管理："<<endl;

cout<<"========================"<<endl;

cout<<"| 1.添加学生 |"<<endl;

cout<<"| 2.查询学生信息 |"<<endl;

cout<<"| 3.修改学生信息 |"<<endl;

cout<<"| 4.删除学生 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"========================"<<endl;

cout<<"请输入选择：";

cin>>choice;

system("cls");

if(choice<=4&&choice>=0)

{

switch(choice)

{

case 1://添加学生

cout<<"请输入学生的姓名：";

cin>>name;

cout<<"请输入学生的学号(4位数字)：";

cin>>number;

cout<<"请输入学生的性别(男生：m 女生：f)：";

cin>>sex;

cout<<"请输入学生的电话号码：";

cin>>phone;

//添加学生信息

addstudent(name,sex,number,phone);

Sleep(1000);

system("cls");

break;

case 2://查询学生信息

cout<<"请输入想要查询的学生的学号(4位数字)：";

cin>>number;

ps=queryNum(number);

if(ps==NULL)

{

cout<<"\n无法找到此学生！"<<endl;

Sleep(1000);

system("cls");

break;

}

ps->show();

Sleep(3000);

system("cls");

break;

case 3://修改学生信息

cout<<"请输入想要修改信息的学生的学号(4位数字)：";

cin>>number;

ps=queryNum(number);

if(ps==NULL)

{

cout<<"\n无法找到此学生！"<<endl;

Sleep(1000);

system("cls");

break;

}

system("cls");

//修改学生信息

modify(ps);

break;

case 4://删除学生

cout<<"请输入想要删除信息的学生的学号(4位数字)：";

cin>>number;

//删除学生信息

if(deletestudent(number)==1)

cout<<"\n删除成功！"<<endl;

else cout<<"\n删除失败！"<<endl;

Sleep(1000);

system("cls");

break;

}

}

else

{

cout<<"请输入正确的选择！"<<endl;

}

}

}

1. **dormitory.cpp文件：**

//////////////////////////////////////////////

//

//文件名：dormitory.cpp

//程序功能：Dormitory类的成员函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include "dormitory.h"

using namespace std;

/////////////////////////////////////////////

//构造函数

//函数功能：初始化床位信息为空

//函数参数：无

//参数返回值：无

//

Dormitory::Dormitory()

{

for(int i=0;i<4;i++)

{

bed[i]=0;

beduser[i]=0;

bedNum[i]=i+1;

}

}

/////////////////////////////////////////////

//get函数

//函数功能：获得宿舍号

//函数参数：无

//参数返回值：整型，宿舍号是整型变量

//

int Dormitory::getdormitoryNumber()

{return dormitoryNumber;}

/////////////////////////////////////////////

//get函数

//函数功能：获得宿舍的使用性别

//函数参数：无

//参数返回值：字符型，宿舍的使用性别是字符型变量

//

char Dormitory::getdormitorySex()

{return dormitorySex;}

/////////////////////////////////////////////

//get函数

//函数功能：获得删除标记

//函数参数：无

//参数返回值：布尔型，0：删除 1：未删

//

bool Dormitory::getdelNote()

{return delNote;}

/////////////////////////////////////////////

//get函数

//函数功能：获得宿舍的使用情况

//函数参数：无

//参数返回值：整型，返回一间宿舍床位已使用数量

//

int Dormitory::use\_dormitory()

{

int use\_count=0;

for(int i=0;i<4;i++)

if(bed[i]==1)use\_count++;

return use\_count;

}

/////////////////////////////////////////////

//get函数

//函数功能：获得具体某个床位的使用情况

//函数参数：整型，要获取信息的床位号

//参数返回值：布尔型，0：床位为空 1：床位不为空

//

bool Dormitory::bed\_use\_case(int bedNum)

{return bed[bedNum-1];}

/////////////////////////////////////////////

//展示函数

//函数功能：展示全部床位的使用情况

//函数参数：无

//参数返回值：无

//

void Dormitory::showbed()

{

cout<<"\n床位使用情况:"<<endl;

for(int i=0;i<4;i++)

{

if(bed[i])cout<<bedNum[i]<<"号床位:已使用 使用者学号："<<beduser[i]<<endl;

else cout<<bedNum[i]<<"号床位:空床"<<endl;

}

}

/////////////////////////////////////////////

//set函数

//函数功能：设置宿舍号

//函数参数：整型，宿舍号为整型变量

//参数返回值：无

//

void Dormitory::setdormitoryNumber(int dorNum)

{dormitoryNumber=dorNum;}

/////////////////////////////////////////////

//set函数

//函数功能：设置宿舍使用性别

//函数参数：字符型，宿舍使用性别为字符型变量

//参数返回值：无

//

void Dormitory::setdormitorySex(char sex)

{dormitorySex=sex;}

/////////////////////////////////////////////

//删除函数

//函数功能：删除宿舍

//函数参数：无

//参数返回值：无

//delNote为0：删除宿舍 1：未删除

//

void Dormitory::deletedormitory()

{delNote=0;}

/////////////////////////////////////////////

//空出床位函数

//函数功能：空出某个床位

//函数参数：整型，要空出床位的床位号

// 将床位使用情况和使用者均设置为0，即为空

//参数返回值：无

//

void Dormitory::not\_use\_bed(int bedNum)

{

bed[bedNum-1]=0;

beduser[bedNum-1]=0;

}

/////////////////////////////////////////////

//使用床位函数

//函数功能：使用某个床位

//函数参数：1.整型，bedNum为要空出床位的床位号

// 2.整型，num为使用者学号

// 将床位使用情况设置为1，表示不空，并填入使用者学号

//参数返回值：无

//

void Dormitory::use\_bed(int bedNum,int num)

{

bed[bedNum-1]=1;

beduser[bedNum-1]=num;

}

/////////////////////////////////////////////

//增加宿舍函数

//函数功能：增加宿舍

//函数参数：1.整型，num为宿舍号

// 2.字符型，sex为宿舍使用性别

//参数返回值：无

//

void Dormitory::addDormitory(int num,char sex)

{

dormitoryNumber=num;

dormitorySex=sex;

delNote=1; //删除标记（0：删除 1：未删）

}

1. **dormitory\_IM\_function.cpp文件：**

//////////////////////////////////////////////

//

//文件名：dormitory\_IM\_function.cpp

//程序功能：Function类关于dormitory\_IM的成员函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <string.h>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

#include "record.h"

using namespace std;

//宿舍信息管理模块

/////////////////////////////////////////////

//查询函数

//函数功能：查询宿舍信息

//函数参数：整型，num为要查询的宿舍号

//参数返回值：Dormitory型指针，指向查找到的宿舍对象

//

Dormitory \*Function::queryroom(int num)

{

for(int i=0;i<=dorNum;i++)

{

if(num==dormitory[i].getdormitoryNumber()&&dormitory[i].getdelNote()==1)

return &dormitory[i];

}

return NULL;

}

/////////////////////////////////////////////

//添加函数

//函数功能：增加宿舍

//函数参数：1.整型，num为新增宿舍的宿舍号

// 2.字符型，sex为新增宿舍的使用性别

//参数返回值：无

//

void Function::addDormitory(int num,char sex)

{

pd=queryroom(num);

if(pd==NULL)

{

dorNum++;

dormitory[dorNum].addDormitory(num,sex);

cout<<"\n添加成功！"<<endl;

}

else cout<<"\n添加失败！此宿舍已存在！"<<endl;

}

/////////////////////////////////////////////

//修改函数

//函数功能：修改宿舍信息

// 1.修改宿舍号

// 2.修改宿舍使用性别

//函数参数：Dormitory型指针，指向要修改信息的宿舍对象

//参数返回值：无

//

void Function::modifyDormitory(Dormitory \*p)

{

int select=1;

int number;

char sex;

//宿舍为空才能修改信息

if(p->use\_dormitory()==0)

{

cout<<"\n这是一间空宿舍！可以进行信息修改！\n"<<endl;

Sleep(1000);

system("cls");

while(select)

{

cout<<" 信息修改："<<endl;

cout<<"======================="<<endl;

cout<<"| 1.修改宿舍号 |"<<endl;

cout<<"| 2.修改宿舍使用性别 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"======================="<<endl;

cout<<"请输入选择：";

cin>>select;

cout<<endl;

system("cls");

if(select<=2||select>=0)

{

switch(select)

{

case 1://修改宿舍号

cout<<"请输入新的宿舍号(3位数)：";

cin>>number;

if(number<=0||number>=1000)

{

cout<<"\n宿舍号错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd=queryroom(number);

//判断宿舍号是否已经存在

if(pd==NULL)

{

p->setdormitoryNumber(number);

cout<<"\n设置成功！"<<endl;

}

else cout<<"\n此宿舍号已存在，无法修改！"<<endl;

Sleep(1000);

system("cls");

break;

case 2://修改宿舍使用性别

cout<<"请输入新的宿舍使用性别(女宿舍：f 男宿舍：m)：";

cin>>sex;

if(sex=='f'||sex=='m')

{

p->setdormitorySex(sex);

cout<<"\n设置成功！"<<endl;

Sleep(1000);

system("cls");

}

else

{

cout<<"\n性别输入错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

break;

}//end switch

}//end if

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

}//end if

else

{

cout<<"\n这间宿舍尚有学生居住！无法修改信息！"<<endl;

cout<<"宿舍信息如下："<<endl;

cout<<"\n宿舍号："<<pd->getdormitoryNumber()<<endl;

cout<<"宿舍使用性别：";

if(pd->getdormitorySex()=='f')cout<<"女";

else cout<<"男";

pd->showbed();

Sleep(4000);

system("cls");

}

}

/////////////////////////////////////////////

//删除函数

//函数功能：删除宿舍

//函数参数：整型，room为要删除的宿舍号

//参数返回值：无

//

void Function::deleteDormitory(int room)

{

pd=queryroom(room);

if(pd==NULL)

{

cout<<"\n无法删除！该宿舍不存在！"<<endl;

Sleep(1000);

}

else

{

if(pd->use\_dormitory()==0)

{

pd->deletedormitory();

cout<<"\n这是一间空宿舍！删除完毕！"<<endl;

Sleep(1000);

}

else

{

cout<<"\n这间宿舍尚有学生居住！无法删除！"<<endl;

cout<<"若要删除！请先调整这些学生的住所！"<<endl;

cout<<"宿舍信息如下："<<endl;

cout<<"\n宿舍号："<<pd->getdormitoryNumber()<<endl;

cout<<"宿舍使用性别：";

if(pd->getdormitorySex()=='f')cout<<"女";

else cout<<"男";

pd->showbed();

Sleep(4000);

}

}

}

1. **dormitory\_IM.cpp文件：**

//////////////////////////////////////////////

//

//文件名：dormitory\_IM.cpp

//程序功能：Function类的成员函数,宿舍基本信息管理

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//信息管理函数

//函数功能：宿舍基本信息管理菜单选项

// 1.添加宿舍

// 2.查询宿舍信息

// 3.修改宿舍信息

// 4.删除宿舍

// 0.退出

//函数参数：无

//参数返回值：无

//

void Function::dormitorydatabase()

{

int choice=1;

int roomNumber;

char sex;

while(choice!=0)

{

cout<<" 宿舍基本信息管理："<<endl;

cout<<"========================"<<endl;

cout<<"| 1.添加宿舍 |"<<endl;

cout<<"| 2.查询宿舍信息 |"<<endl;

cout<<"| 3.修改宿舍信息 |"<<endl;

cout<<"| 4.删除宿舍 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"========================"<<endl;

cout<<"请输入选择：";

cin>>choice;

system("cls");

if(choice<=4&&choice>=0)

{

switch(choice)

{

case 1://添加宿舍

cout<<"请输入要添加的宿舍号(3位数)：";

cin>>roomNumber;

if(roomNumber<=0||roomNumber>=1000)

{

cout<<"\n宿舍号错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

cout<<"请输入宿舍的使用性别(男生：m 女生：f)：";

cin>>sex;

//判断性别输入是否正确

if(sex=='f'||sex=='m')

{

addDormitory(roomNumber,sex);

Sleep(1000);

system("cls");

}

else

{

cout<<"\n性别输入错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

break;

case 2://查询宿舍信息

cout<<"请输入要查询的宿舍的宿舍号(3位数)：";

cin>>roomNumber;

if(roomNumber<=0||roomNumber>=1000)

{

cout<<"\n宿舍号错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd=queryroom(roomNumber);

if(pd==NULL)

{

cout<<"\n未找到此宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

//展示宿舍信息

cout<<"\n宿舍号："<<pd->getdormitoryNumber()<<endl;

cout<<"宿舍使用性别：";

if(pd->getdormitorySex()=='f')cout<<"女";

else cout<<"男";

pd->showbed();

Sleep(4000);

system("cls");

break;

case 3://修改宿舍信息

cout<<"请输入要修改的宿舍的宿舍号(3位数)：";

cin>>roomNumber;

if(roomNumber<=0||roomNumber>=1000)

{

cout<<"\n宿舍号错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd=queryroom(roomNumber);

if(pd==NULL)

{

cout<<"\n无法修改！该宿舍不存在！"<<endl;

Sleep(1000);

system("cls");

break;

}

modifyDormitory(pd);

break;

case 4://删除宿舍

cout<<"请输入要删除的宿舍号：";

cin>>roomNumber;

if(roomNumber<0)

{

cout<<"\n宿舍号错误！"<<endl;

break;

}

deleteDormitory(roomNumber);

system("cls");

break;

}

}

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}

}

1. **dormitory\_manage.cpp文件：**

//////////////////////////////////////////////

//

//文件名：dormitory\_manage.cpp

//程序功能：Function类的成员函数，住宿管理菜单

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string.h>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//住宿管理函数

//函数功能：住宿管理

// 1.宿舍基本信息管理

// 2.宿舍分配管理

// 3.入宿登记

// 4.退宿处理

// 5.调宿处理

//函数参数：无

//参数返回值：无

//

void Function::dormitoryManage()

{

int choice=1;

while(choice!=0)

{

cout<<" 住宿管理："<<endl;

cout<<"========================"<<endl;

cout<<"| 1.宿舍基本信息管理 |"<<endl;

cout<<"| 2.宿舍分配管理 |"<<endl;

cout<<"| 3.入宿登记 |"<<endl;

cout<<"| 4.退宿处理 |"<<endl;

cout<<"| 5.调宿处理 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"========================"<<endl;

cout<<"请输入选择：";

cin>>choice;

system("cls");

if(choice<=5&&choice>=0)

{

switch(choice)

{

case 1://宿舍基本信息管理

dormitorydatabase();

break;

case 2://宿舍分配管理

allotDormitory();

break;

case 3://入宿登记

liveDormitory();

break;

case 4://退宿处理

leaveDormitory();

break;

case 5://调宿处理

adjustDormitory();

break;

}//end switch

}//end if

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

}

1. **live\_in\_dormitory.cpp文件：**

//////////////////////////////////////////////

//

//文件名：live\_in\_dormitory.cpp

//程序功能：学生登记入宿

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//入宿登记函数

//函数功能：入宿登记

//函数参数：无

//参数返回值：无

//

void Function::liveDormitory()

{

int studentNumber;

cout<<"请输入学生的学号：";

cin>>studentNumber;

ps=queryNum(studentNumber);

if(ps==NULL)cout<<"\n未找到此学生！"<<endl;

else if(ps->getmark()==1) cout<<"\n该学生已经入宿，无法重复登记入宿！"<<endl;

else if(ps->getroomNumber()==0) cout<<"\n请先为学生分配宿舍！"<<endl;

else

{

//记上入宿标记

ps->live();

if(ps->getmark()==1)cout<<"\n学生成功登记入宿！"<<endl;

else cout<<"\n错误！学生登记入宿失败！"<<endl;

}

Sleep(1000);

system("cls");

}

1. **leave\_dormitory.cpp文件：**

//////////////////////////////////////////////

//

//文件名：leave\_dormitory.cpp

//程序功能：处理退宿相关功能

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//退宿处理函数

//函数功能：退宿处理

// 1.人工退宿

// 2.批量退宿

//函数参数：无

//参数返回值：无

//

void Function::leaveDormitory()

{

int select=1;

int studentNumber;

while(select)

{

cout<<" 退宿处理："<<endl;

cout<<"================="<<endl;

cout<<"| 1.人工退宿 |"<<endl;

cout<<"| 2.批量退宿 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"================="<<endl;

cout<<"请输入选择：";

cin>>select;

system("cls");

if(select>=0&&select<=2)

{

switch(select)

{

case 1://人工退宿

cout<<"请输入要退宿的学生的学号：";

cin>>studentNumber;

ps=queryNum(studentNumber);

//先判断是否可以找到该学生

if(ps==NULL)

{

cout<<"\n该学生不存在！"<<endl;

Sleep(1000);

system("cls");

break;

}

//再判断该学生是否已经有入宿登记

else if(ps->getmark()==0)

{

cout<<"\n无法退宿，该学生未登记入宿！"<<endl;

Sleep(1000);

system("cls");

break;

}

//退宿并删除学生

ps->leave();

if(ps->getmark()==0)

{

//找到宿舍，空出床位

pd=queryroom(ps->getroomNumber());

if(pd==NULL)

{

cout<<"\n错误！无法找到该学生所在宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd->not\_use\_bed(ps->getbedNumber());

ps->delstudent();

cout<<"\n退宿成功！"<<endl;

Sleep(1000);

}

else

{

cout<<"\n退宿失败！"<<endl;

Sleep(1000);

}

system("cls");

break;

case 2://批量退宿

cout<<"批量退宿中..."<<endl;

Sleep(1000);

//学生标上退宿标记并删除

for(int i=0;i<stuNum;i++)

{

//判断学生是否已经入宿

if(student[i].getmark()==1)

{

student[i].leave();

if(student[i].getmark()==0)

{

//将学生的信息从宿舍中删除

pd=queryroom(student[i].getroomNumber());

if(pd==NULL)

{

cout<<"\n错误！无法找到该学生所在宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd->not\_use\_bed(ps->getbedNumber());

}

student[i].delstudent();

}

}

system("cls");

break;

}//end switch

}//end if

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

}

1. **allot\_dormitory.cpp文件：**

//////////////////////////////////////////////

//

//文件名：allot\_dormitory.cpp

//程序功能：分配学生宿舍

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <algorithm> //sort函数

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//分配宿舍函数

//函数功能：分配宿舍

//函数参数：无

//参数返回值：无

//

void Function::allotDormitory()

{

int tempBoyNum[Maxs],tempGirlNum[Maxs],girlNum,boyNum;//用于批量分配时分性别学号排序

int roomNum,count;

int studentNumber;

int select=1,opt=1;

int room,bed;

while(select)

{

cout<<" 宿舍分配管理："<<endl;

cout<<"======================================="<<endl;

cout<<"| 1.人工安排宿舍&床位 |"<<endl;

cout<<"| 2.系统批量安排宿舍&床位(按学号分配) |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"======================================="<<endl;

cout<<"请输入选择：";

cin>>select;

cout<<endl;

system("cls");

if(select<=2&&select>=0)

{

switch(select)

{

case 1://人工安排床位

cout<<"请输入学生的学号：";

cin>>studentNumber;

//查找并判断学生是否存在

ps=queryNum(studentNumber);

if(ps==NULL)

{

cout<<"\n无法找到该学生！"<<endl;

Sleep(1000);

break;

}

//若学生尚未分配到床位

if(ps->getroomNumber()==0||ps->getbedNumber()==0)

{

//找到所有有空床位的宿舍

cout<<"\n以下宿舍尚有空床位：\n"<<endl;

cout<<"宿舍号 使用性别"<<endl;

for(int i=0;i<=dorNum;i++)

{

if(dormitory[i].use\_dormitory()<4)

{

cout<<' '<<dormitory[i].getdormitoryNumber()<<" ";

if(dormitory[i].getdormitorySex()=='f')cout<<"女"<<endl;

else cout<<"男"<<endl;

}

}

//打印出学生信息

cout<<"\n学生信息如下："<<endl;

ps->show();

Sleep(5000);

system("cls");

while(1)

{

cout<<"请为该学生分配宿舍及床位：\n"<<endl;

cout<<"请输入宿舍号：";

cin>>room;

//查找这间宿舍

pd=queryroom(room);

if(pd==NULL)

{

cout<<"\n不存在此宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

else

{

//学生性别与宿舍使用性别一致才能分配床位

if(ps->getsex()==pd->getdormitorySex())

{

system("cls");

//展示床位使用情况，再做出选择

pd->showbed();

cout<<"\n\n================"<<endl;

cout<<"| 1.选择床位 |"<<endl;

cout<<"| 0.回退 |"<<endl;

cout<<"================"<<endl;

cout<<"请选择：";

cin>>opt;

if(opt==1)

{

//选择床位

cout<<"\n请输入床位号(1~4)：";

cin>>bed;

//判断床位是否已被使用

if(pd->bed\_use\_case(bed)==1)

{

cout<<"\n这张床已被使用！"<<endl;

Sleep(1000);

system("cls");

}

else

{

system("cls");

//使用床位

ps->setroom\_bed(room,bed);

pd->use\_bed(bed,ps->getnumber());

//显示学生信息

cout<<"分配成功！新的学生信息如下：\n"<<endl;

ps->show();

Sleep(4000);

system("cls");

break;

}

}

if(opt!=0&&opt!=1)

{

cout<<"\n请输入正确的选择！"<<endl;

Sleep(500);

}

system("cls");

}//end if 判断性别是否相符

else

{

cout<<"\n宿舍使用性别不符合！"<<endl;

Sleep(1000);

system("cls");

}

}//end else 判断宿舍是否存在

}//end while

}//end if 判断学生是否已经分配了床位

else

{

//打印出学生信息

ps->show();

cout<<"\n该学生已经分配了床位！"<<endl;

Sleep(2000);

system("cls");

}

break;

case 2://系统批量安排床位，按学号进行分配

//初始化学生数组的学生数量

boyNum=-1;

girlNum=-1;

cout<<"正在分配中. . ."<<endl;

Sleep(1000);

//先将学生们的学号按照性别存储在数组中，以便排序

for(int i=0;i<=stuNum;i++)

{

//挑出未被删除的学生

if(student[i].getnote()!=0)

{

if(student[i].getsex()=='m')

{

boyNum++;

tempBoyNum[boyNum]=student[i].getnumber();

}

else

{

girlNum++;

tempGirlNum[girlNum]=student[i].getnumber();

}

}

}

//按学号排序，从小到大

sort(tempBoyNum,tempBoyNum+boyNum+1);

sort(tempGirlNum,tempGirlNum+girlNum+1);

//分配宿舍&床位

//男生

roomNum=0;

count=1; //用于计数，每四个换一间宿舍

for(int i=0;i<=boyNum;i++)

{

//先判断宿舍够不够

if(roomNum>dorNum)

{

cout<<"\n男生宿舍数量不够！请先添加宿舍！"<<endl;

Sleep(1000);

break;

}

//先找到该学生

ps=queryNum(tempBoyNum[i]);

if(ps==NULL)

{

cout<<"\n出错！无法找到学生！"<<endl;

break;

}

if(count>4)

{

roomNum++;

count=1; //一间宿舍床位共4个

}

while(1)

{

//先判断宿舍够不够

if(roomNum>dorNum) break;

//分配宿舍&床位

if(dormitory[roomNum].getdormitorySex()=='m')

{

//将床位信息登记到学生的信息中

ps->setroom\_bed(dormitory[roomNum].getdormitoryNumber(),count);

dormitory[roomNum].use\_bed(count,tempBoyNum[i]);

count++;

break;

}

else roomNum++;

}

}

//女生

roomNum=0;

count=1; //用于计数，每四个换一间宿舍

for(int i=0;i<=girlNum;i++)

{

//先判断宿舍够不够

if(roomNum>dorNum)

{

cout<<"\n女生宿舍数量不够！请先添加宿舍！"<<endl;

Sleep(1000);

break;

}

//先找到该学生

ps=queryNum(tempGirlNum[i]);

if(ps==NULL)

{

cout<<"\n出错！无法找到学生！"<<endl;

break;

}

if(count>4)

{

roomNum++;

count=1; //一间宿舍床位共4个

}

while(1)

{

//判断宿舍够不够

if(roomNum>dorNum) break;

//分配宿舍&床位

if(dormitory[roomNum].getdormitorySex()=='f')

{

//将床位信息登记到学生的信息中

ps->setroom\_bed(dormitory[roomNum].getdormitoryNumber(),count);

dormitory[roomNum].use\_bed(count,tempGirlNum[i]);

count++;

break;

}

else roomNum++;

}//end while

}//end for

break;

}//end switch

system("cls");

}//end if

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

}

1. **adjust\_dormitory.cpp文件：**

//////////////////////////////////////////////

//

//文件名：adjust\_dormitory.cpp

//程序功能：调整学生宿舍

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

using namespace std;

/////////////////////////////////////////////

//调宿处理函数

//函数功能：调整学生宿舍

//函数参数：无

//参数返回值：无

//

void Function::adjustDormitory()

{

Student \*p1,\*p2;

int select=1;

int stuNum1,stuNum2;

int temp\_roomNum,temp\_bed;

int roomNum,bed;

while(select!=0)

{

cout<<" 调宿管理："<<endl;

cout<<"=========================="<<endl;

cout<<"| 1.两学生交换宿舍 |"<<endl;

cout<<"| 2.调整一名学生的宿舍 |"<<endl;

cout<<"| 0.回退 |"<<endl;

cout<<"=========================="<<endl;

cout<<"请输入选择：";

cin>>select;

system("cls");

if(select<=2&&select>=0)

{

switch(select)

{

case 1://两学生互换宿舍

while(1)

{

cout<<"请输入想要调整宿舍的两学生的学号：";

cout<<"\n学生1：";

cin>>stuNum1;

cout<<"学生2：";

cin>>stuNum2;

if(stuNum2!=stuNum1)break;

cout<<"\n两学生学号相同！出错！\n"<<endl;

Sleep(1000);

system("cls");

}

p1=queryNum(stuNum1);

p2=queryNum(stuNum2);

if(p1!=NULL&&p2!=NULL)

{

//先确认两学生性别相同

if(p1->getsex()==p2->getsex())

{

//展示目前的情况

cout<<"\n两学生信息如下："<<endl;

p1->show();

p2->show();

Sleep(6000);

system("cls");

//交换两个人的宿舍

temp\_roomNum=p1->getroomNumber();

temp\_bed=p1->getbedNumber();

p1->setroom\_bed(p2->getroomNumber(),p2->getbedNumber());

p2->setroom\_bed(temp\_roomNum,temp\_bed);

//修改两间宿舍的信息

//学生1

pd=queryroom(p1->getroomNumber());

if(pd==NULL)

{

cout<<"\n无法找到学生所属宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd->use\_bed(p1->getbedNumber(),p1->getnumber());

//学生2

pd=queryroom(p2->getroomNumber());

if(pd==NULL)

{

cout<<"\n无法找到学生所属宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd->use\_bed(p2->getbedNumber(),p2->getnumber());

//展示新的情况

cout<<"\n交换宿舍之后信息如下："<<endl;

p1->show();

p2->show();

Sleep(6000);

system("cls");

}//end if 判断两学生性别是否相同

else

{

cout<<"\n两学生的性别不同，无法交换宿舍！"<<endl;

Sleep(1000);

system("cls");

}

}//end if 判断是否可以找到学生

else

{

cout<<"\n学号错误！无法找到学生！"<<endl;

Sleep(1000);

system("cls");

}

break;

case 2://调整一学生宿舍

cout<<"请输入想要调整宿舍的学生的学号：";

cin>>stuNum1;

ps=queryNum(stuNum1);

if(ps==NULL)

{

cout<<"\n未找到此学生！"<<endl;

Sleep(1000);

system("cls");

break;

}

//调整前

//找到所有有空床位的宿舍

cout<<"\n以下宿舍尚有空床位：\n"<<endl;

cout<<"宿舍号 使用性别"<<endl;

for(int i=0;i<=dorNum;i++)

{

if(dormitory[i].use\_dormitory()<4)

{

cout<<' '<<dormitory[i].getdormitoryNumber()<<" ";

if(dormitory[i].getdormitorySex()=='f')cout<<"女"<<endl;

else cout<<"男"<<endl;

}

}

//打印学生信息

cout<<"\n学生信息如下："<<endl;

ps->show();

Sleep(5000);

system("cls");

cout<<"请输入要调整的宿舍号：";

cin>>roomNum;

pd=queryroom(roomNum);

if(pd==NULL)

{

cout<<"\n该宿舍不存在！"<<endl;

Sleep(1000);

system("cls");

break;

}

//判断宿舍使用性别与学生性别是否一致

if(pd->getdormitorySex()==ps->getsex())

{

cout<<"\n宿舍信息如下："<<endl;

cout<<"宿舍号："<<pd->getdormitoryNumber()<<endl;

cout<<"使用性别：";

if(pd->getdormitorySex()=='f')cout<<"女"<<endl;

else cout<<"男"<<endl;

pd->showbed();

cout<<"\n请输入床位号，选择一个空床：";

cin>>bed;

if(bed>4||bed<1)

{

cout<<"\n床位号出错！"<<endl;

Sleep(1000);

system("cls");

break;

}

if(pd->bed\_use\_case(bed)==0)

{

//调整宿舍

temp\_bed=ps->getbedNumber();

temp\_roomNum=ps->getroomNumber();

ps->setroom\_bed(roomNum,bed);

//将学生信息填入新的宿舍中

pd->use\_bed(bed,ps->getnumber());

//将原来宿舍的信息删去

pd=queryroom(temp\_roomNum);

pd->not\_use\_bed(temp\_bed);

system("cls");

//调整后

cout<<"\n换宿舍之后信息如下："<<endl;

ps->show();

Sleep(3000);

system("cls");

}//end if 判断该床位是否已被使用

else

{

cout<<"\n该床位已被使用！"<<endl;

Sleep(1000);

system("cls");

}

}//end if 判断宿舍使用性别与学生性别是否一致

else

{

cout<<"\n宿舍使用性别与学生性别不一致，无法换宿舍！"<<endl;

Sleep(1000);

system("cls");

}

break;

}//end switch

}//end if

else

{

cout<<"请输入正确选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

}

1. **record.cpp文件：**

//////////////////////////////////////////////

//

//文件名：record.cpp

//程序功能：Morning\_Evening\_record类的成员函数

// Visitor类的成员函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <string.h>

#include "record.h"

using namespace std;

//学生早出晚归登记

/////////////////////////////////////////////

//构造函数

//函数功能：初始化早出晚归此数为0

//函数参数：无

//参数返回值：无

//

Morning\_Evening\_record::Morning\_Evening\_record()

{

morningTimes=0;

eveningTimes=0;

}

/////////////////////////////////////////////

//get函数

//函数功能：获取学号

//函数参数：无

//参数返回值：整型，返回学生学号

//

int Morning\_Evening\_record::getNum()

{return studentNumber;}

/////////////////////////////////////////////

//早出登记函数

//函数功能：学生早出次数增加

//函数参数：无

//参数返回值：无

//

void Morning\_Evening\_record::morningRecord()

{morningTimes++;}

/////////////////////////////////////////////

//晚归登记函数

//函数功能：学生晚归次数增加

//函数参数：无

//参数返回值：无

//

void Morning\_Evening\_record::eveningRecord()

{eveningTimes++;}

/////////////////////////////////////////////

//添加函数

//函数功能：添加早出晚归的学生

//函数参数：字符型指针：指向姓名，整型：学号

//参数返回值：无

//

void Morning\_Evening\_record::addStud(char \*na,int nu)

{

strcpy(studentName,na);

studentNumber=nu;

}

/////////////////////////////////////////////

//show函数

//函数功能：展示学生早出晚归信息

//函数参数：无

//参数返回值：无

//

void Morning\_Evening\_record::showStudentInformation()

{

cout<<"\n姓名："<<studentName<<endl;

cout<<"学号："<<studentNumber<<endl;

cout<<"早出次数："<<morningTimes<<endl;

cout<<"晚归此数："<<eveningTimes<<endl;

}

//来访者登记

/////////////////////////////////////////////

//set函数

//函数功能：设置来访者信息

//函数参数：字符型指针，字符型，指向来访者的各种信息

//参数返回值：无

//

void Visitor::setVisitInformation(char \*na,char se,char \*v\_t,char \*l\_t,char \*re,int dorNum)

{

strcpy(visitorName,na);

visitorSex=se;

strcpy(visitTime,v\_t);

strcpy(leaveTime,l\_t);

strcpy(reason,re);

visitDorNum=dorNum;

}

/////////////////////////////////////////////

//show函数

//函数功能：展示来访信息

//函数参数：无

//参数返回值：无

//

void Visitor::showVisitorInformation()

{

cout<<"\n来访者姓名："<<visitorName<<endl;

cout<<"来访者性别："<<visitorSex<<endl;

cout<<"访问时间："<<visitTime<<endl;

cout<<"离开时间："<<leaveTime<<endl;

cout<<"来访原因："<<reason<<endl;

cout<<"访问宿舍号："<<visitDorNum<<endl;

}

1. **get\_in\_out\_manage.cpp文件：**

//////////////////////////////////////////////

//

//文件名：get\_in\_out\_manage.cpp

//程序功能：Function类的成员函数,出入管理

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <string.h>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

#include "record.h"

using namespace std;

/////////////////////////////////////////////

//查找函数

//函数功能：查找早出晚归学生

//函数参数：整型，num是要查找的学生学号

//参数返回值：Morning\_Evening\_record型指针

//

Morning\_Evening\_record \*Function::query(int num)

{

for(int i=0;i<=studNum;i++)

{

if(num==stud[i].getNum())

return &stud[i];

}

return NULL;

}

/////////////////////////////////////////////

//早出登记函数

//函数功能：登记早出学生信息

//函数参数：无

//参数返回值：无

//

void Function::morning\_go\_out()

{

int num;

cout<<"请输入登记早出的学生的学号(4位数)：";

while(1)

{

cin>>num;

ps=queryNum(num);

if(ps==NULL)

{

cout<<"\n不存在此学生！"<<endl;

Sleep(1000);

system("cls");

}

else if(ps->getmark()==0)

{

cout<<"\n该学生尚未入宿！信息错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

else

{

pr=query(num);

//判断学生是否已经存在，若不存在先添加学生

cout<<"\n学生信息如下："<<endl;

if(pr==NULL)

{

studNum++;

stud[studNum].addStud(ps->getname(),num);

stud[studNum].morningRecord();

stud[studNum].showStudentInformation();

}

else

{

pr->morningRecord();

pr->showStudentInformation();

}

cout<<"\n早出信息登记成功！"<<endl;

Sleep(4000);

system("cls");

break;

}

cout<<"请重新输入学号(4位数)：";

}

}

/////////////////////////////////////////////

//晚归登记函数

//函数功能：登记晚归学生信息

//函数参数：无

//参数返回值：无

//

void Function::evening\_go\_back()

{

int num;

cout<<"请输入登记晚归的学生的学号(4位数)：";

while(1)

{

cin>>num;

ps=queryNum(num);

if(ps==NULL)

{

cout<<"\n不存在此学生！"<<endl;

Sleep(1000);

system("cls");

}

else if(ps->getmark()==0)

{

cout<<"\n该学生尚未入宿！信息错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

else

{

pr=query(num);

//判断学生是否已经存在，若不存在先添加学生

cout<<"\n学生信息如下："<<endl;

if(pr==NULL)

{

studNum++;

stud[studNum].addStud(ps->getname(),num);

stud[studNum].eveningRecord();

stud[studNum].showStudentInformation();

}

else

{

pr->eveningRecord();

pr->showStudentInformation();

}

cout<<"\n晚归信息登记成功！"<<endl;

Sleep(4000);

system("cls");

break;

}

cout<<"请重新输入学号(4位数)：";

}

}

/////////////////////////////////////////////

//来访者登记函数

//函数功能：登记来访者信息

//函数参数：无

//参数返回值：无

//

void Function::visit\_record()

{

char name[10];

char sex;

char visitTime[8];

char leaveTime[8];

char reason[50];

int visitDorNum;

cout<<"请输入来访者姓名：";

cin>>name;

cout<<"请输入来访者性别(男：m 女：f)：";

cin>>sex;

cout<<"请输入访问时间(24小时制,格式：xx:xx)：";

cin>>visitTime;

cout<<"请输入结束时间(24小时制,格式：xx:xx)：";

cin>>leaveTime;

cout<<"请简述来访理由：";

cin>>reason;

cout<<"请输入访问宿舍号：";

while(1)

{

cin>>visitDorNum;

pd=queryroom(visitDorNum);

if(pd==NULL) cout<<"\n宿舍号错误！\n\n请重新输入：";

else if(pd->use\_dormitory()==0) cout<<"\n这是一间空宿舍！\n\n请重新选择：";

else break;

}

visNum++;

visitor[visNum].setVisitInformation(name,sex,visitTime,leaveTime,reason,visitDorNum);

cout<<"\n来访者信息登记成功！"<<endl;

Sleep(1000);

system("cls");

}

/////////////////////////////////////////////

//出入管理函数

//函数功能：出入管理

// 1.早出登记

// 2.晚归登记

// 3.来访者登记

//函数参数：无

//参数返回值：无

//

void Function::get\_in\_out\_manage()

{

int choice=1;

while(choice!=0)

{

cout<<" 出入管理："<<endl;

cout<<"=================="<<endl;

cout<<"| 1.早出登记 |"<<endl;

cout<<"| 2.晚归登记 |"<<endl;

cout<<"| 3.来访者登记 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"=================="<<endl;

cout<<"请输入选择：";

cin>>choice;

system("cls");

if(choice<=3&&choice>=0)

{

switch(choice)

{

case 1://早出登记

morning\_go\_out();

break;

case 2://晚归登记

evening\_go\_back();

break;

case 3://来访者登记

visit\_record();

break;

}

}

}

}

1. **query\_information.cpp文件：**

//////////////////////////////////////////////

//

//文件名：query\_information.cpp

//程序功能：实现各种信息查询

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h> //system函数头文件

#include <windows.h> //sleep函数头文件

#include "student.h"

#include "dormitory.h"

#include "function.h"

#include "record.h"

using namespace std;

/////////////////////////////////////////////

//求最大公约数函数

//函数功能：辗转相除法求最大公约数

//函数参数：整型m，n

//参数返回值：整型，返回最大公约数

//

int greatest\_common\_divisor(int m,int n)

{

int a,b;

if(n>m) {a=m;m=n;n=a;}//确保大数除小数

while(m%n!=0)

{b=m%n;m=n;n=b;}

return n;

}

/////////////////////////////////////////////

//信息查询函数

//函数功能：信息查询

// 1.查询学生信息含住宿信息

// 2.查询宿舍信息含住宿信息

// 3.查询早出晚归信息

// 4.查询来访者信息

// 5.查询入宿人数

// 6.查询宿舍空床情况

// 7.查询住宿男女比例

//函数参数：无

//参数返回值：无

//

void Function::query\_information()

{

int choice=1;

int studentNum;

int roomNum;

int countStu;//在宿人数

int boyNum=0,girlNum=0;

int n; //最大公约数

while(choice!=0)

{

cout<<" 信息查询："<<endl;

cout<<"================================="<<endl;

cout<<"| 1.查询学生信息含住宿信息 |"<<endl;

cout<<"| 2.查询宿舍信息含住宿信息 |"<<endl;

cout<<"| 3.查询早出晚归信息 |"<<endl;

cout<<"| 4.查询来访者信息 |"<<endl;

cout<<"| 5.查询入宿人数 |"<<endl;

cout<<"| 6.查询宿舍空床情况 |"<<endl;

cout<<"| 7.查询住宿男女比例 |"<<endl;

cout<<"| 0.退出 |"<<endl;

cout<<"================================="<<endl;

cout<<"请输入选择：";

cin>>choice;

system("cls");

if(choice<=7&&choice>=0)

{

switch(choice)

{

case 1://查询学生信息

cout<<"请输入想要查询的学生的学号(4位数字)：";

cin>>studentNum;

ps=queryNum(studentNum);

if(ps==NULL)

{

cout<<"\n无法找到此学生！"<<endl;

Sleep(1000);

system("cls");

break;

}

ps->show();

Sleep(5000);

system("cls");

break;

case 2://查询宿舍信息

cout<<"请输入要查询的宿舍的宿舍号：";

cin>>roomNum;

if(roomNum<=0)

{

cout<<"\n宿舍号错误！"<<endl;

Sleep(1000);

system("cls");

break;

}

pd=queryroom(roomNum);

if(pd==NULL)

{

cout<<"\n未找到此宿舍！"<<endl;

Sleep(1000);

system("cls");

break;

}

cout<<"\n宿舍信息如下："<<endl;

cout<<"\n宿舍号："<<pd->getdormitoryNumber()<<endl;

cout<<"\n宿舍使用性别：";

if(pd->getdormitorySex()=='f')cout<<"女"<<endl;

else cout<<"男"<<endl;

pd->showbed();

Sleep(5000);

system("cls");

break;

case 3://查询早出晚归信息

cout<<"早出晚归学生名单：\n"<<endl;

for(int i=0;i<=studNum;i++)

stud[i].showStudentInformation();

Sleep(8000);

system("cls");

break;

case 4://查询来访者信息

cout<<"来访者信息如下：\n"<<endl;

for(int i=0;i<=visNum;i++)

visitor[i].showVisitorInformation();

Sleep(10000);

system("cls");

break;

case 5://查询入宿人数

cout<<"学生人数："<<stuNum+1<<endl;

//将统计入宿人数的计数机初始化为0

countStu=0;

for(int i=0;i<=stuNum;i++)

if(student[i].getnote()==1&&student[i].getmark()==1)//未删除并且处于入宿状态

countStu++;

cout<<"\n在宿人数："<<countStu<<endl;

Sleep(1000);

system("cls");

break;

case 6://查询宿舍空床情况

cout<<"以下宿舍尚有空床位："<<endl;

cout<<"宿舍号 使用性别"<<endl;

for(int i=0;i<=dorNum;i++)

if(dormitory[i].use\_dormitory()<4)

{

cout<<' '<<dormitory[i].getdormitoryNumber()<<" ";

if(dormitory[i].getdormitorySex()=='f')cout<<"女"<<endl;

else cout<<"男"<<endl;

}

Sleep(10000);

system("cls");

break;

case 7://查询住宿男女比例

cout<<"学生人数："<<stuNum+1<<endl;

for(int i=0;i<=stuNum;i++)

{

if(student[i].getsex()=='f')girlNum++;

else boyNum++;

}

n=greatest\_common\_divisor(boyNum,girlNum);

cout<<"\n住宿男女比例："<<boyNum/n<<':'<<girlNum/n<<endl;

Sleep(1000);

system("cls");

break;

}//end switch

}//end if

else

{

cout<<"请输入正确的选择！"<<endl;

Sleep(500);

system("cls");

}

}//end while

}

1. **construction.cpp文件：**

//////////////////////////////////////////////

//

//文件名：construction.cpp

//程序功能：Function类的构造函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include "student.h"

#include "dormitory.h"

#include "function.h"

#include "record.h"

using namespace std;

/////////////////////////////////////////////

//构造函数

//函数功能：

//1.从student.dat文件读入Student类的对象数据

//2.从dormitory.dat文件读入Dormitory类的对象数据

//3.从visitor.dat文件读入Visitor类的对象数据

//4.从morning\_evening\_record.dat文件读入Morning\_evening\_record类的对象数据

//5.从administrator.dat文件读入Administrator类的对象数据

//函数参数：无

//参数返回值：无

//

Function::Function()

{

//学生管理

Student s;

stuNum=-1;

//以读入方式打开文件

fstream file1("student.dat",ios::in|ios::binary);

if(!file1)

{

cerr<<"student.dat无法正常打开!"<<endl;

exit(1);

}

//读入学生的数据

while(1)

{

file1.read((char\*)&s,sizeof(s));

if(!file1)break; //如果读入完毕，终止循环

stuNum++;

student[stuNum]=s;

}

file1.close();

//宿舍管理

Dormitory d;

dorNum=-1;

//以读入方式打开文件

fstream file2("dormitory.dat",ios::in|ios::binary);

if(!file2)

{

cout<<"dormitory.dat无法正常打开！"<<endl;

exit(1);

}

//读入宿舍的数据

while(1)

{

file2.read((char\*)&d,sizeof(d));

if(!file2)break; //如果读入完毕，终止循环

dorNum++;

dormitory[dorNum]=d;

}

file2.close();

//出入管理

//来访者

Visitor v;

visNum=-1;

//以读入方式打开文件

fstream file3("visitor.dat",ios::in|ios::binary);

if(!file3)

{

cout<<"visitor.dat无法正常打开！"<<endl;

exit(1);

}

//读入来访者的数据

while(1)

{

file3.read((char\*)&v,sizeof(v));

if(!file3)break; //如果读入完毕，终止循环

visNum++;

visitor[visNum]=v;

}

file3.close();

//学生早出晚归

Morning\_Evening\_record st;

studNum=-1;

//以读入方式打开文件

fstream file4("morning\_evening\_record.dat",ios::in|ios::binary);

if(!file4)

{

cout<<"morning\_evening\_record.dat无法正常打开！"<<endl;

exit(1);

}

//读入早出晚归的数据

while(1)

{

file4.read((char\*)&st,sizeof(st));

if(!file4)break; //如果读入完毕，终止循环

studNum++;

stud[studNum]=st;

}

file4.close();

//登录管理

Administrator a;

adNum=-1;

//以读入方式打开文件

fstream file5("administrator.dat",ios::in|ios::binary);

if(!file5)

{

cout<<"administrator.dat无法正常打开！"<<endl;

exit(1);

}

//读入管理员的数据

while(1)

{

file5.read((char\*)&a,sizeof(a));

if(!file5)break; //如果读入完毕，终止循环

adNum++;

administrator[adNum]=a;

}

file5.close();

}

1. **destructor.cpp文件：**

//////////////////////////////////////////////

//

//文件名：destructor.cpp

//程序功能：Function类的析构函数

//作者：Pipi

//日期：2016.10.15

//

//////////////////////////////////////////////

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include "student.h"

#include "dormitory.h"

#include "function.h"

#include "record.h"

using namespace std;

/////////////////////////////////////////////

//析构函数

//函数功能：

//1.将Student类的对象数据写入student.dat文件

//2.将Dormitory类的对象数据写入dormitory.dat文件

//3.将Visitor类的对象数据写入visitor.dat文件

//4.将Morning\_evening\_record类的对象数据写入morning\_evening\_record.dat文件

//5.将Administrator类的对象数据写入administrator.dat文件

//函数参数：无

//参数返回值：无

//

Function::~Function()

{

//学生管理

fstream file1("student.dat",ios::out|ios::binary);

if(!file1)

{

cout<<"student.dat文件无法正常打开！"<<endl;

exit(1);

}

for(int i=0;i<=stuNum;i++)

if(student[i].getnote()==1)

file1.write((char\*)&student[i],sizeof(student[i]));

file1.close();

//宿舍管理

fstream file2("dormitory.dat",ios::out|ios::binary);

if(!file2)

{

cout<<"dormitory.dat文件无法正常打开！"<<endl;

exit(1);

}

for(int i=0;i<=dorNum;i++)

if(dormitory[i].getdelNote()==1)

file2.write((char\*)&dormitory[i],sizeof(dormitory[i]));

file2.close();

//出入管理

//来访者

fstream file3("visitor.dat",ios::out|ios::binary);

if(!file3)

{

cout<<"visitor.dat文件无法正常打开！"<<endl;

exit(1);

}

for(int i=0;i<=visNum;i++)

file3.write((char\*)&visitor[i],sizeof(visitor[i]));

file3.close();

//学生早出晚归

fstream file4("morning\_evening\_record.dat",ios::out|ios::binary);

if(!file4)

{

cout<<"morning\_evening\_record.dat文件无法正常打开！"<<endl;

exit(1);

}

for(int i=0;i<=studNum;i++)

file4.write((char\*)&stud[i],sizeof(stud[i]));

file4.close();

//登录管理

fstream file5("administrator.dat",ios::out|ios::binary);

if(!file5)

{

cout<<"administrator.dat文件无法正常打开！"<<endl;

exit(1);

}

for(int i=0;i<=adNum;i++)

file5.write((char\*)&administrator[i],sizeof(administrator[i]));

file5.close();

}