

算法与数据结构设计

（2017/ 2018 学年 第 一 学期）

|  |  |
| --- | --- |
| **题 目：** | **实验室安排系统** |

|  |  |
| --- | --- |
| **专 业** | 计算机科学与技术 |
| **学 生 姓 名** | 袁富志 |
| **班 级 学 号** | B15040125 |
| **指 导 教 师** | 朱洁 |
| **指 导 单 位** | 计算机学院计算机科学与技术系 |
| **日 期** | 2017.10.30 – 2017.11.10 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **评 分 细 则** | **评分项** | **优秀** | **良好** | **中等** | **差** |
| **遵守机房规章制度** |  |  |  |  |
| **上机时的表现** |  |  |  |  |
| **学习态度** |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |
| **程序设计能力** |  |  |  |  |
| **解决问题能力** |  |  |  |  |
| **课题功能实现情况** |  |  |  |  |
| **算法设计合理性** |  |  |  |  |
| **算法效能评价** |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |
| **内容详实程度** |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |
| **回答问题准确度** |  |  |  |  |
| **评 分 等 级** | **教师签名：**  **年 月 日** | | | | |
| **备 注** |  | | | | |

注：评分等级有五种：优秀、良好、中等、及格、不及格

**实验室安排系统**

1. **课题内容和要求**

课题内容：

为《算法与数据结构设计》课程开发实验室安排系统。该课程开课时间为2周，在两周内需要上6次课，每次连续4个学时。每天上午下午可各安排一次课，周末不安排上课。该系统包含两个程序：管理员程序和教师申请程序。

在教师申请界面中，教师录入所授课班级ID、班级人数、不希望的时间安排（例如不安排周一上午和周四下午）；信息提交后将被存储到后台数据库中。

在管理员界面中，管理员录入现有的实验室信息，包括实验室地址、实验室所能容纳学生数量。管理员可以选择“课表生成”，实验室安排算法读入已有申请和实验室信息，生成总课程安排，并写入数据库。

当总课程安排生成后，教师再次进入教师申请程序时，可以看到自己的课表。

请使用C/C++或Java实现。数据库可随意选择。

课题要求：

(1) 两个程序可以由同一个界面进行入口，或者分别两个界面，不要求身份验证；用户只需要输入ID即可，ID事先已写入数据库。

(2) 算法生成的课程安排，要求尽可能为每个老师安排6次课，所安排实验室能容纳所上课班级；按照教师申请的时间次序，尽可能优先满足先申请教师的期望实验安排；当申请较多而管理员录入的实验室较少，允许出现有的教师申请无法满足的情况；

(3) 实物演示时要求能够说明所采用算法思想；

(4) 程序操作友好、健壮；操作界面简洁美观。

(5) 完善程序功能。例如身份验证、教师与程序员可以修改自己的录入信息等。

(6) 先进而高效的算法。

(7) 界面美观，课表整齐美观

课题理解：

该课题主要设计一个分别面向管理员和老师的实验室安排体统，该系统包含管理员程序和教师申请程序，其中教师申请程序里面，教师录入自己所授课班级ID、班级人数、不希望的时间安排（例如不安排周一上午和周四下午），而在管理员程序中，管理员录入现有的实验室信息，包括实验室地址、实验室所能容纳学生数量。管理员根据当前老师提出的申请，合理为每个老师安排教学任务。老师可在管理员生成课表之后，查看自己的教学计划。因此：

系统要实现的功能：

验证老师登录

实现老师的功能

验证管理员登录

实现管理员功能

管理员要实现的功能：

录入实验室地址，实验室容量

修改录入的信息

实现排课表算法，尽可能为每个老师安排6次课，所安排实验室能容纳所上课班级，按照教师申请的时间次序，尽可能优先满足先申请教师的期望实验安排，当申请较多而管理员录入的实验室较少，允许出现有的教师申请无法满足的情况。

老师要实现的功能:

录入授课班级教ID、班级人数、不希望的时间安排

修改自己录入的信息

查看自己的课表

本次课程设计采用Java编程语言，借助Eclipse为开发工具，使用Java GUI,JDBC,MySQL数据库等技术来完成这次课题，具体实现方式如下：

1. **数据结构说明**

2.1系统结构：

该系统主要包含登录程序，管理员程序，教师程序。管理员程序又包含插入，修改，生成课程表程序等。而教师程序包含插入，修改，查看课程表等。具体的系统结构图如图2.1所示：

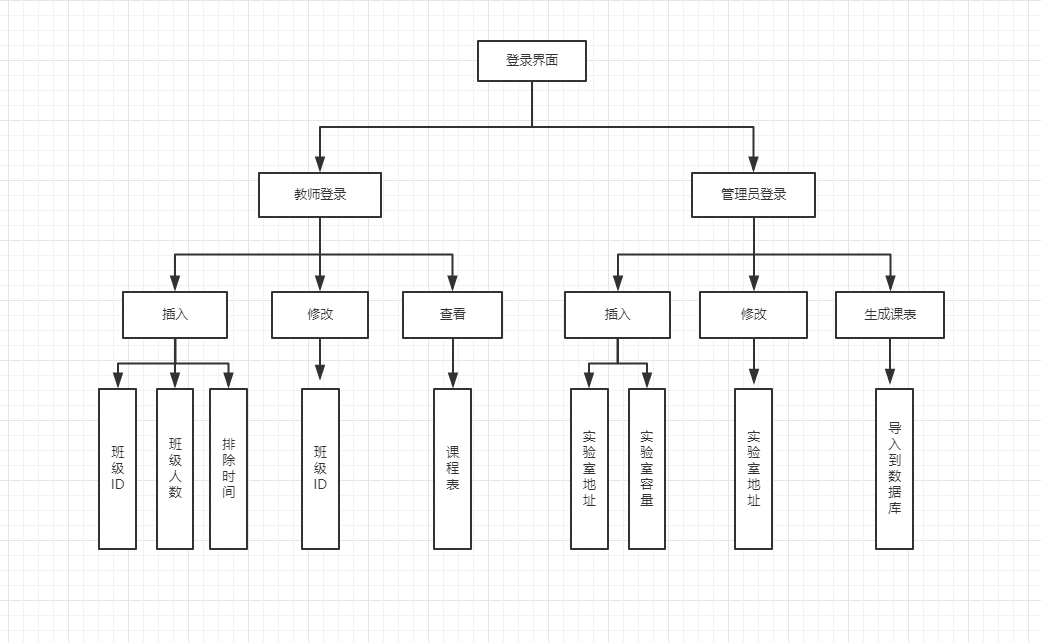
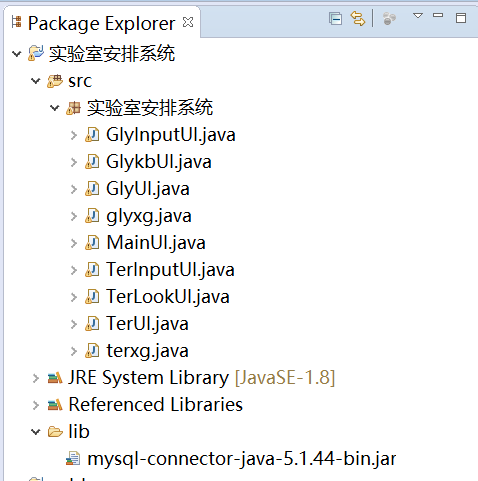


图2.1

2.2 函数结构:

该系统包含了九大板块，每个板块都定义了自己的函数，具体的函数说明如图2.2所示：



说明：

MainUI: 系统主界面

GlyUI: 管理员界面

GlyInputUI: 管理员输入界面

GlykbUI: 管理员生成课表函数

Glyxg: 管理员修改界面

TerUI: 教师登录界面

TerInputUI； 教师输入页面

TerLookUI: 教师查看课表界面

Terxg: 教师修改界面

Terxg: 教师修改页面

图2.2

2.3教师数据结构：

我们将教师存储在数据库teacher表中，每一条记录代表了一位老师录入的信息，包括教师姓名，教授的班级ID，班级人数，以及自己不想被安排排课的时间片，具体的结构如图2.3所示：

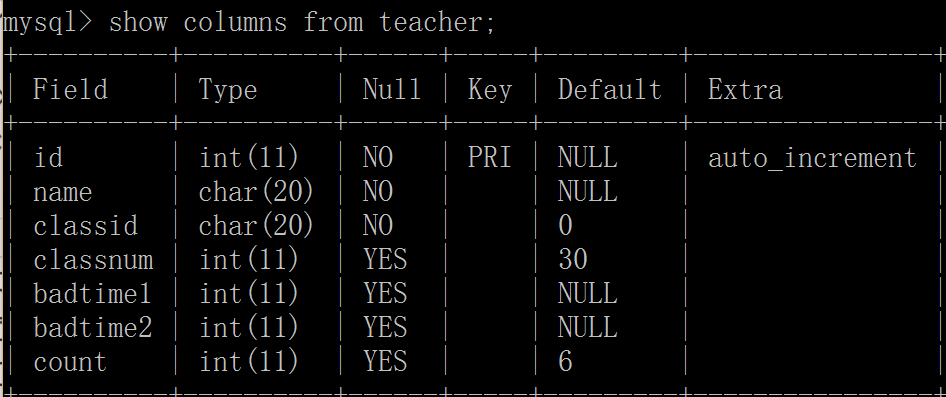


图2.3

（注：id为记录在数据表中的位置，与优先级有关，name为教师姓名，classid为教师所授班名称，classnum为班级人数，badtime1为不想上课的时间片1，badtime2位不想上课的时间片2，count为授课次数是6次）

2.4实验室数据结构：

我们将实验室的相关信息存储在表administrator中，具体包括实验室地址，实验室容量。结构图如图2.4所示：

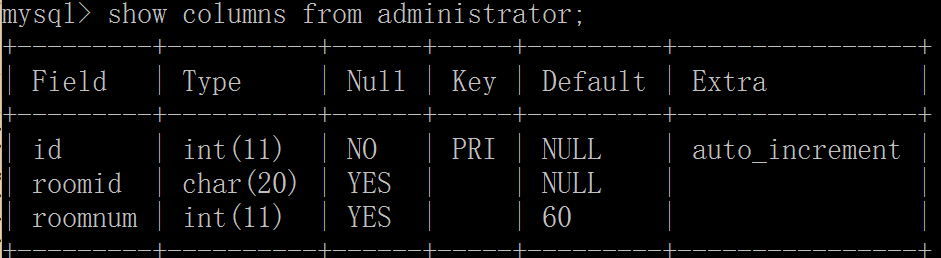


图2.4

（注：id为记录在数据表中的位置，roomid为实验室地址，roomnum为实验室容量）

2.5 课程表结构

我们将课程表的相关信息存储在表timetable中，每一条记录都包含了教师姓名，上课的地点，不想被排课的时间以及被排课的时间片，具体的结构如图2.5所示：

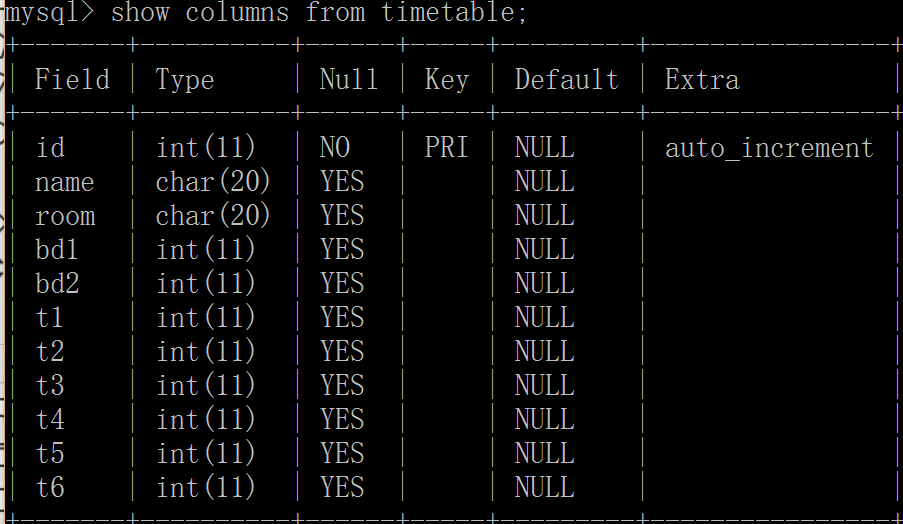


图2.5

（注：id为记录在数据表中的位置，与优先级有关，name为教师姓名，room为上课地点，bd1为不想上课的时间片1，bd2位不想上课的时间2,t1~t6为教师上课的时间片）

1. **算法设计**

该课题主要的算法就是课程表生成算法，排课表算法的大致思路如下：

1) 同一班级的学生在同一时间不能安排在两个实验室

2) 同一教师在同一时间不能教授两个班级

3) 同一实验室在同一时间不能安排两个班级

4) 某一实验室上课的人数不应大于实验室的容纳量

5) 课程在一周上多次时, 要有一定的间隔性

6) 对同一教师, 同一上课对象应尽量选择相对固定的几个实验室

7) 优先满足一些特殊要求（比如有些老师喜欢上上午的课，可以优先满足，有些老师不喜欢上某个时间片的课，我们尽量避免给他们在这个时间段排课）

因此，我将两周的时间分为20个时间片，以老师的喜好时间为基础来给他们安排合适的时间片来开展教学活动。将20个时间片中可用的时间片分成六部分，每部分生成一个上课的时间片，最后将生成的时间片作为该老师的上课时间，并写入到课程表中。时间片的排布如图3.1所示：



图3.1

生成课表算法：要生成A老师的课表，先通过访问数据库获得A老师不想被安排上课的时间片a和b，和A老师要上课的实验室roomid，在A老师的可用时间片里，为A老师生成6个上课的时间片t1~t6，并写入到数据库课程表表单中，供老师查询。我们定义以下几个函数来帮助我们实现该算法：

①获取当前教师人数 int Getternum()

②获取当前以及生成的课程表记录数int Gettimetablenum()

③返回表<tablename>中第i条记录 ResultSet find(int id,String tablename)

④得到id为i的老师上课的时间片 Set<Integer> getrandom(int id,int a,int b)

⑤生成姓名为<name>的老师的课程表记录，将其添加到数据库administrator表中：

void addtimetable(int id,String name,String room,int bd1,int bd2,int time[])String room,int bd1,int bd2,int time[])

算法的流程图如图3.2所示：

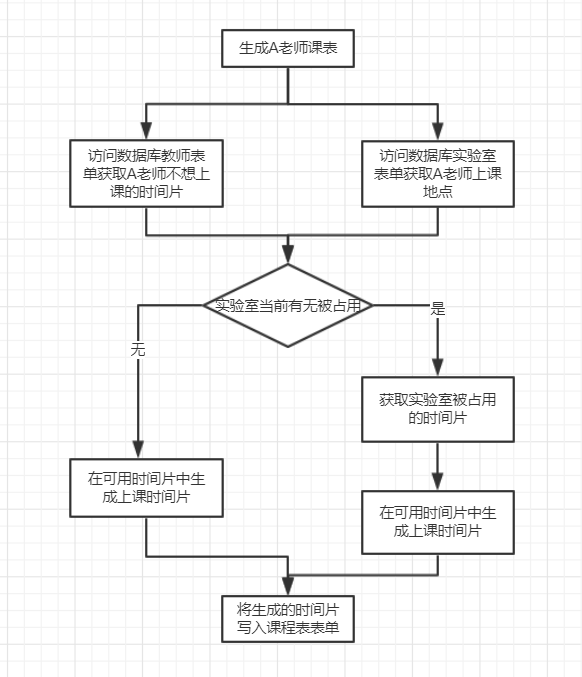


图3.2

**四、详细设计**

4.1 设计Getternum()函数用来返回教师的人数：

**public** **static** **int** Getternum(){

Connection con;//与数据库建立连接

String driver = "com.mysql.jdbc.Driver";

String url = "jdbc:mysql://localhost:3306/Demo";

String user = "root";

String password = "yfz";

**int** rowcount=0;

**try** {

Class.forName(driver);

con=(Connection) DriverManager.getConnection(url,user,password);

Statement statement = (Statement) con.createStatement();

String sql = "select count(\*) from teacher";

ResultSet rs = statement.executeQuery(sql); //执行SQL语句来获取表teacher中的记录个数

**while**(rs.next()){

rowcount=rs.getInt(1);

}

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** rowcount;//返回表teacher中的记录数（教师人数）

}

4.2设计ResultSet find(**int** i,String s) 用来返回数据库中表名称为s的第i条记录

**public** **static** ResultSet find(**int** i,String s) **throws** SQLException{

Connection con;//与数据库建立连接

String driver = "com.mysql.jdbc.Driver";

String url = "jdbc:mysql://localhost:3306/Demo";

String user = "root";

String password = "yfz";

ResultSet rs=**null**;

**try** {

Class.forName(driver);

con = (Connection)DriverManager.getConnection(url,user,password);//登录到数据库

PreparedStatement stmt=**null**;

String sql="select \* from "+s+" where id = "+String.valueOf(i);//获取数据库中表名称为s的第i条记录

stmt=(PreparedStatement) con.prepareStatement(sql);//执行SQL语句

rs = stmt.executeQuery();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

**return** rs; //返回结果集rs

}

4.3设计Set<Integer> getrandom(**int** i,**int** a,**int** b)得到id为i的老师上课的时间片

**public** **static** Set<Integer> getrandom(**int** i,**int** a,**int** b) **throws** SQLException{

Random ran=**new** Random(i);

Set<Integer> set =**new** TreeSet<>();//定义set集来存放为该老师生成的上课时间片

**if**(i<=10){

/\*

如果i<=10,该老师可以优先获得指定实验室的使用权，只需要在20个时间片中可用的16个时间片里面生成6个上课用的时间片即可，为了使结果合理，我们将可用的时间片分成6部分，每一个部分生成一个上课的时间片，如下：

\*/

**while**(**true**){

**int** r1=ran.nextInt(3)+1;

**if**(r1!=a&&r1!=b)

set.add(r1);

**if**(set.size()==1)

**break**;

}

**while**(**true**){

**int** r2=ran.nextInt(3)+4;

**if**(r2!=a&&r2!=b)

set.add(r2);

**if**(set.size()==2)

**break**;

}

**while**(**true**){

**int** r3=ran.nextInt(4)+7;

**if**(r3!=a&&r3!=b)

set.add(r3);

**if**(set.size()==3)

**break**;

}

System.***out***.println(set);

**while**(**true**){

**int** r4=ran.nextInt(3)+11;

**if**(r4!=(a+10)&&r4!=(b+10))

set.add(r4);

**if**(set.size()==4)

**break**;

}

**while**(**true**){

**int** r5=ran.nextInt(3)+14;

**if**(r5!=(a+10)&&r5!=(b+10))

set.add(r5);

**if**(set.size()==5)

**break**;

}

**while**(**true**){

**int** r6=ran.nextInt(4)+17;

**if**(r6!=(a+10)&&r6!=(b+10))

set.add(r6);

**if**(set.size()==6)

**break**;

}

System.***out***.println(set);

}**else**{

/\*

如果当前老师的id>10，那么他对指定实验室的优先权要低于该实验室的第一个指定老师，即他只能从该实验室未被第一个老师占用的时间片里面选取合适的时间片来进行教学活动，具体的实现如下：

\*/

Connection con;

String driver = "com.mysql.jdbc.Driver";

String url = "jdbc:mysql://localhost:3306/Demo";

String user = "root";

String password = "yfz";

ResultSet rs=**null**;//定义t1~t6存放实验室被占用的时间片

**int** t1=0;

**int** t2=0;

**int** t3=0;

**int** t4=0;

**int** t5=0;

**int** t6=0;

**try** {

Class.*forName*(driver);

con = (Connection)DriverManager.*getConnection*(url,user,password);

PreparedStatement stmt=**null**;

String sql="select \* from timetable where id = "+String.*valueOf*(i-10);

stmt=(PreparedStatement) con.prepareStatement(sql);

rs = stmt.executeQuery();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

**while**(rs.next()){

t1=rs.getInt(6);

t2=rs.getInt(7);

t3=rs.getInt(8);

t4=rs.getInt(9);

t5=rs.getInt(10);

t6=rs.getInt(11);

}

**int** n[]=**new** **int**[21];//定义n[]来存放被占用的时间片合集

**int** m[]=**new** **int** [21];//定义m[]来存放可用的时间片合集

**int** num=0;

**for**(**int** j=0;j<20;j++){

n[j]=1;

}

n[a]=0;n[b]=0;n[t1]=0;n[t2]=0;n[t3]=0;n[t4]=0;n[t5]=0;n[t6]=0;

**for**(**int** j=0;j<20;j++){

**if**(n[j]!=0){

m[num++]=j;

}

}

**int** r1=ran.nextInt(2)+0;

set.add(m[r1]);

**int** r2=ran.nextInt(2)+2;

set.add(m[r2]);

**int** r3=ran.nextInt(2)+4;

set.add(m[r3]);

**int** r4=ran.nextInt(2)+6;

set.add(m[r4]);

**int** r5=ran.nextInt(2)+8;

set.add(m[r5]);

**int** r6=ran.nextInt(num-10)+10;

set.add(m[r6]);

}

**return** set; //返回生成的时间片集合set

}

4.4 定义addtimetable(**int** id,String name,String room,**int** bd1,**int** bd2,**int** time[])来向表timetable插入生成的课程表记录

**public** **static** **void** addtimetable(**int** id,String name,String room,**int** bd1,**int** bd2,**int** time[]){

Connection con;

String driver = "com.mysql.jdbc.Driver";

String url = "jdbc:mysql://localhost:3306/Demo";

String user = "root";

String password = "yfz";

**try** {

Class.*forName*(driver);

con=(Connection) DriverManager.*getConnection*(url,user,password);

String sql = "insert into timetable values(?,?,?,?,?,?,?,?,?,?,?)";//向表timetable中插入记录

PreparedStatement pstmt;

**try** {//插入记录的初始化

pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setInt(1, id);

pstmt.setString(2, name);

pstmt.setString(3, room);

pstmt.setInt(4, bd1);

pstmt.setInt(5, bd2);

pstmt.setInt(6, time[0]);

pstmt.setInt(7, time[1]);

pstmt.setInt(8, time[2]);

pstmt.setInt(9, time[3]);

pstmt.setInt(10, time[4]);

pstmt.setInt(11, time[5]);

pstmt.executeUpdate();

pstmt.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

con.close();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

4.5定义Glykb类来实现生成课程表算法

public class Glykb{

public static void main(String[] args){

new Glykb();

}

Public Glykb()throws SQLException{

**int** ternum=*Getternum*();//得到当前教师人数

**int** timetabenum=*Gettimetablenum*();//得到当前课程表记录数

**for**(**int** i=timetabenum+1;i<=ternum;i++){//对还未生成课表的老师生成课表

String tname=**null**,roomid=**null**;

**int** tb1=0,tb2=0;

ResultSet rs1=*find*(i, "teacher");

**while**(rs1.next()){

tname=rs1.getString(2);

tb1=rs1.getInt(5);//获取不想上课的时间片t1,t2

tb2=rs1.getInt(6);

}

**if**(i<=10){

ResultSet rs2=*find*(i, "administrator");

**while**(rs2.next()){

roomid=rs2.getString(2);//获取指定的上课实验室名称

}

}**else**{

ResultSet rs2=*find*(i-10, "administrator");

**while**(rs2.next()){

roomid=rs2.getString(2);

}

}

Set<Integer> set=*getrandom*(i,tb1, tb2);

**int** time[]=**new** **int**[6];//time[]用来存放时间片

**int** j=0;

Iterator<Integer> it=set.iterator();

**while**(it.hasNext()){

time[j++]=it.next();

}

*addtimetable*(i, tname, roomid, tb1, tb2, time);//添加记录

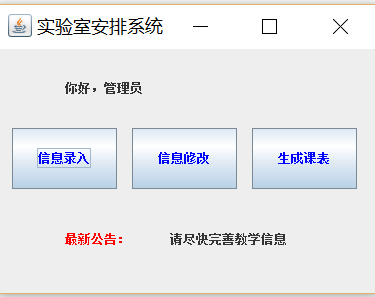
}

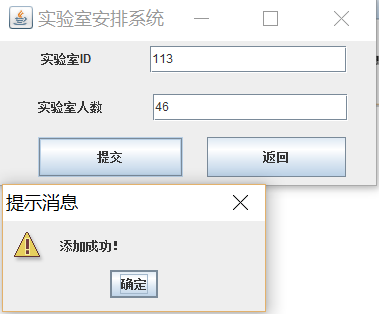
}

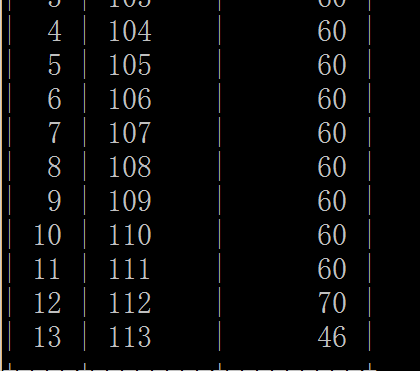
}

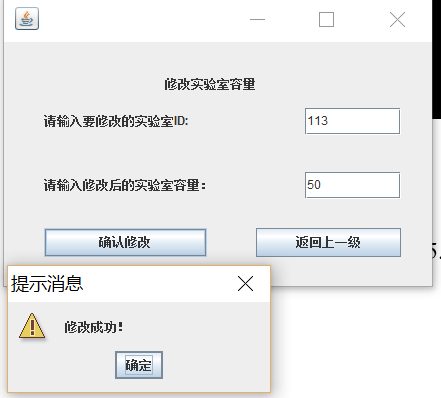
**五、测试数据及其结果分析（格式：宋体，4号，加粗，两端对齐）**

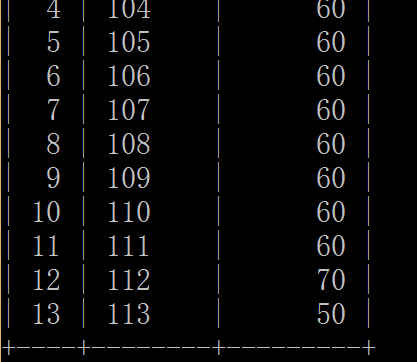
管理员登录界面（图5.1） 登录成功界面(图5.2)

  
图5.1 图5.2

信息录入界面（图5.3） 录入成功后台数据（图5.4）

  
 图5.3 图5.4

信息修改界面（图5.4） 修改成功后台界面（图5.5）

  
 图5.4 图5.5

教师登录界面（图5.6） 登录成功界面（图5.7）

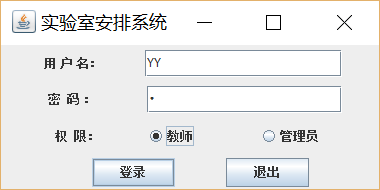
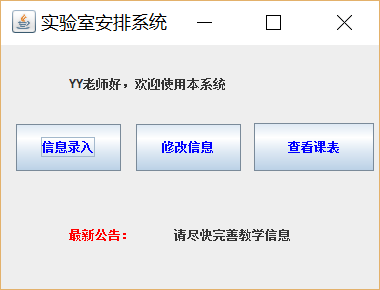
 

图5.6 图5.7

信息录入界面（图5.8） 录入成功后台显示（图5.9）

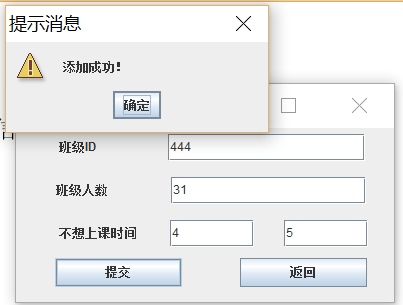
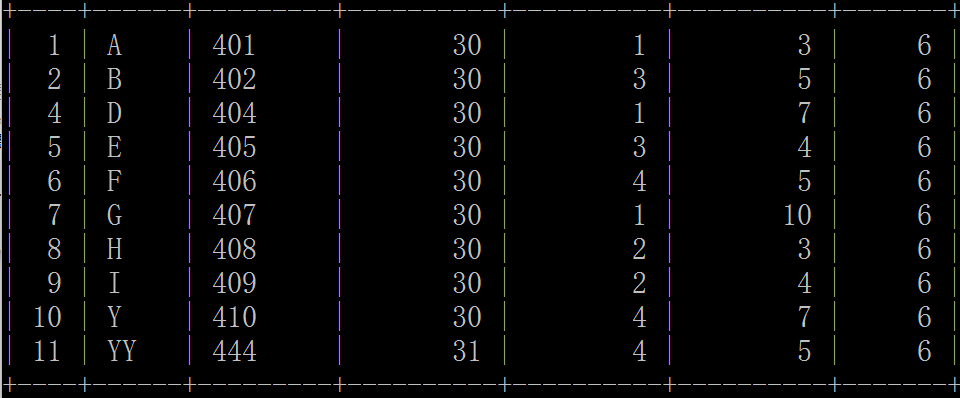
 

图5.8 图5.9

信息修改界面（图5.10） 修改成功后台显示（图5.11）

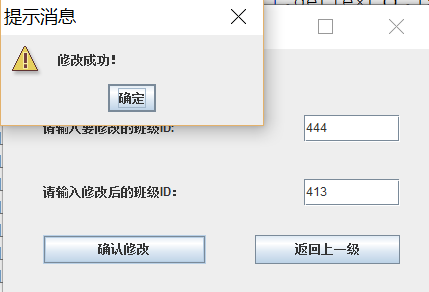
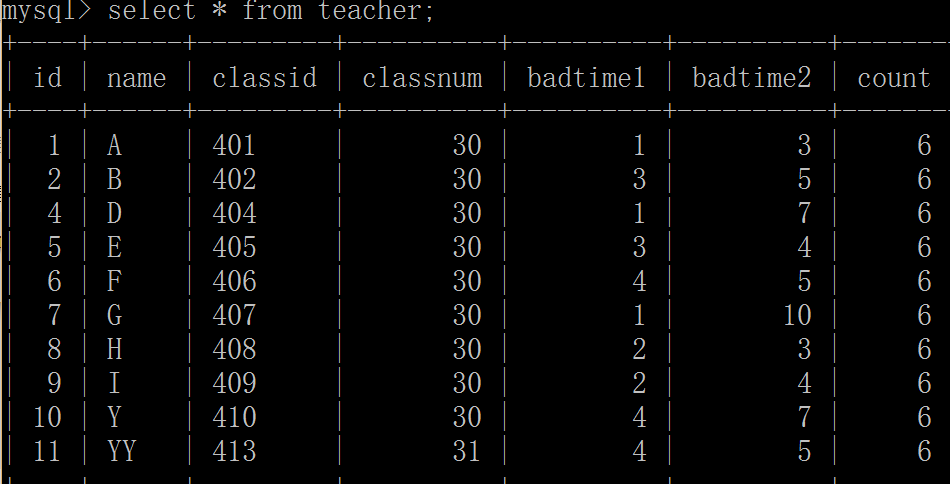
 

图5.10 图5.11

显示课程表界面（图5.11）



图5.11

**六、算法设计和程序调试过程中的问题**

在这次算法设计中，我遇到的问题有以下：

1. 一开始我的思路就是先把界面做好，在完成相关的算法。但用Java不会做界面跳转，在查阅了很多资料后，我用了一种类的关联方式来解决了这一问题。例如我要从A界面跳转到B界面：

Class A(){ class B(){

A(){ B(){

new B(); }

} }

}

2,开始的时候用Java连接MySQL数据库，代码没有问题但是无法连接，而且我已经把jdbc-mysql的驱动包添加到工程当中，后来发现了问题所在，虽然导入了驱动包，但是我忘记了在eclipse里面配置它，路径添加好了以后，解决了这一问题。

3，在老师登录成功后的界面，由于界面间并没有联系，所以无法知道当前是哪一位老师在使用该系统，为了解决这一问题，我在与老师有关的界面的类的构造函数的参数里面，加入了string类型的参数，这样每次进入界面，就把老师的名字传入进去，就可以获取当前是哪一位老师在使用该系统。

4，有一个是我想从数据库里面根据姓名来获取这条记录，我开始写的SQL语句是：

String sql="select \* from "+s+" where name = "+name;（s为表名称，name为表中的列名称，类型为CHAR(20)）,系统会报错，最后在一论坛上发现了解决这个问题的办法。需要给name加上引号：String sql="select \* from "+s+" where name = '"+name+"'";这样才解决了问题。

5，在生成上课时间片的算法中，本来用的random类来做的，在1~20生成6个随机数，但发现生成的数太过于集中，不符合预期，于是我把这个范围分成了6个部分，每个部分生成一个时间片，这样既可以满足要求，而且结果也比较合理。

**七、课程设计总结**

本次课程设计重点就是排课表算法，虽然最后算是满足了基本的需求，但是还有好多不完善的地方，设计的时候就是按照10个教室，每个老师固定一个教室这样的方式来分配的，这与现实生活中排课的情况还存在明显的差距。所以，算法还有待改进的地方，比如老师上课的地点不固定，给老师安排课程的优先级，原来是按照教师申请的优先级来排课，改进后可以按照教师的需求为优先级，要求高的老师可优先安排，需求集中的时间段优先安排，可能出现部分老师需求较高而无法满足的情况等等。

在这次课程设计当中，我了解到了我的不足，如算法的不完善、不细心和耐心不是很好等等。在这次课程设计时，有时候常常因为一些变量名，函数名大小写，拼写不注意，出错耽误了一些时间，以后一定要仔仔细细的做这些基本的工作，减少因为非能力之外的因素影响工作进度。不仅如此，在进行程序设计时，还要注意想好思路。即要有恰当模块名、变量名、常量名、子程序名等。将每个功能的模块，即函数名要清晰的表述出来，使用户能够一目了然此程序的功能。同时也要学会试着写注释，不仅方便自己回头检查，而且也便于别人看。

通过本次课程设计也加强了我的动手能力，提升了我考虑问题的思维方式。这短短两周的时间的确令我受益良多，也给我留下了很好的编程体验。从最初拿到题目的手足无措到一步步扩宽思路，不断学习相关的编程知识，不断地测试，调试，修改，完善，直到最后完成整个编程，可以说不仅巩固了以前所学过的知识，而且学到了很多在书本上所没有学到过的知识。而且，这两周的课程设计也让我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，才能认识到自己的不足，从而提高自己的实际动手能力和独立思考的能力。

最后，感谢朱洁老师在这两周的悉心指导，不仅帮助我解决了很多头疼的难题，也给我提出来许多的建设性建议，让我受益颇多，在此表达我真诚的谢意！