云南大学软件学院 实验报告

课程： 软件工程概论 学期： 2024 - 2025学年 第一学期 任课教师： 莫启

专业： 软件工程 学号： 20221120044 姓名： 唐嘉骏 成绩：

**实验2**

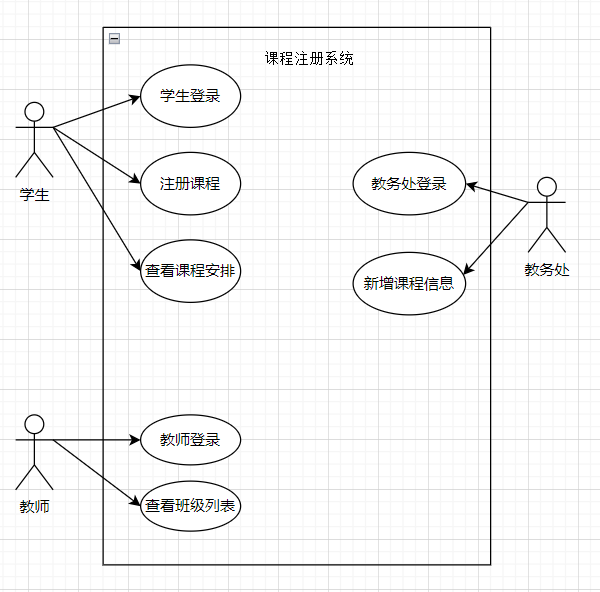
针对实验1中的课程注册系统，利用面向对象技术对其进行设计和开发：

（1）利用UML中用例图对其进行需求分析，每个用例需含有详细的用例说明文档；

（2）对其进行面向对象的设计，即：1）设计其对应体系结构，以UML中组件图刻画（用Visio或draw.io绘制）；2）确定组件图中每个组件中含有对象、画出对象间的交互及其每个对象的类图；3）进行数据库设计；

（3）对其进行简单开发，包括代码实现及测试（单元测试和集成测试）等部分。

（1）**用例图**：



### **用例说明文档**

### 用例编号: 001

| **项目** | **内容** |
| --- | --- |
| **用例名称** | 学生登录 |
| **参与者** | 学生 |
| **用例说明** | 学生通过系统登录账号以访问课程注册系统。 |
| **前置条件** | 学生已注册账号。 |
| **基本事件流** | 1. 学生输入账号。  2. 学生输入密码。  3. 学生按回车键登录。  4. 系统验证账号和密码是否匹配。  5. 系统将学生引导至学生菜单。 |
| **异常事件流** | 4a. 系统检测到账号或密码不正确，系统提示“密码或用户名错误！”。  4b. 系统检测到故障，系统提示“登录失败，请稍后重试”。 |
| **后置条件** | 学生登录成功，系统进入到主页。 |

### 用例编号: 002

| **项目** | **内容** |
| --- | --- |
| **用例名称** | 课程注册 |
| **参与者** | 学生 |
| **用例说明** | 学生选择并注册课程。 |
| **前置条件** | 学生已登录系统。 |
| **基本事件流** | 1. 学生在菜单选择“课程注册”。  2. 学生输入选课课程名和教师工号。  3. 系统将学生添加到课程的学生名单中。  4. 系统提示学生选课成功。 |
| **异常事件流** | 4a. 数据库未能找到对应的课程，系统提示“未找到课程”  4b. 数据库保存时出现故障，系统提示“选课失败，请稍后重试”。 |
| **后置条件** | 学生成功选课，系统更新课程的学生名单。 |

### 用例编号: 003

| **项目** | **内容** |
| --- | --- |
| **用例名称** | 查看课程安排 |
| **参与者** | 学生 |
| **用例说明** | 学生可以查看自己的课程安排。 |
| **前置条件** | 学生已登录系统并已选课。 |
| **基本事件流** | 1. 学生在菜单选择“课程安排”。  2. 系统显示学生的课程安排，包括课程号，课程名称，教师姓名等信息。 |
| **异常事件流** | 1a. 系统获取数据库数据时发生故障，系统提示“无法加载课程安排，请稍后重试”。 |
| **后置条件** | 学生成功查看自己的课程安排。 |

### 用例编号: 004

| **项目** | **内容** |
| --- | --- |
| **用例名称** | 教师登录 |
| **参与者** | 教师 |
| **用例说明** | 教师通过系统登录以访问课程注册系统。 |
| **前置条件** | 教师已注册账号。 |
| **基本事件流** | 1. 教师输入账号。  2. 教师入密码。  3. 教师按回车键登录。  4. 系统验证账号和密码是否匹配。  5. 系统将教师引导至菜单。 |
| **异常事件流** | 4a. 系统检测到账号或密码不正确，系统提示“密码或用户名错误！”。  4b. 系统检测到故障，系统提示“登录失败，请稍后重试”。 |
| **后置条件** | 教师登录成功，系统进入到教师菜单。 |

### 用例编号: 005

| **项目** | **内容** |
| --- | --- |
| **用例名称** | 查看班级列表 |
| **参与者** | 教师 |
| **用例说明** | 教师查看自己负责的班级列表。 |
| **前置条件** | 教师已登录系统。 |
| **基本事件流** | 1. 教师选择“查看班级列表”。  2. 系统显示教师所负责的班级列表，包括班级名称和学生名单等信息。 |
| **异常事件流** | 1a. 系统获取数据时发生故障，系统提示“无法加载班级列表，请稍后重试”。 |
| **后置条件** | 教师成功查看自己负责的班级列表。 |

### 用例编号: 006

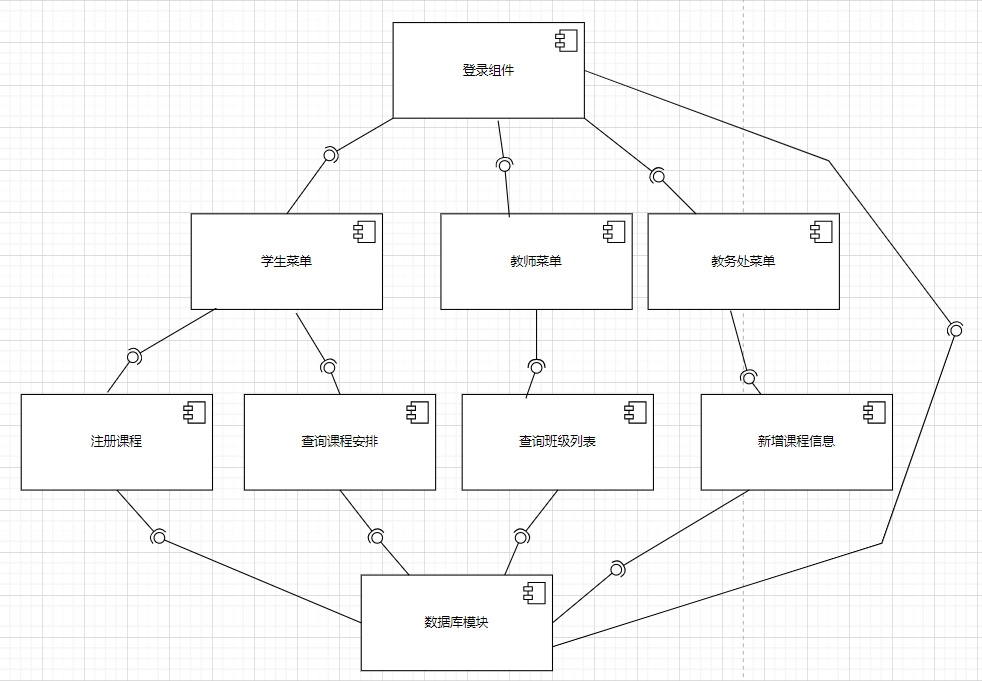
| **项目** | **内容** |
| --- | --- |
| **用例名称** | 教务处登录 |
| **参与者** | 教务处 |
| **用例说明** | 教务处通过系统登录以管理课程信息。 |
| **前置条件** | 教务处工作人员账号已录入。 |
| **基本事件流** | 1. 教务处工作人员输入账号。  2. 教务处工作人员输入密码。  3. 教务处工作人员按回车键登录。  4. 系统验证账号和密码是否匹配。  5. 系统将教务处工作人员引导至菜单。 |
| **异常事件流** | 4a. 系统检测到账号或密码不正确，系统提示“密码或用户名错误！”。  4b. 系统检测到故障，系统提示“登录失败，请稍后重试”。 |
| **后置条件** | 教务处工作人员登录成功，系统进入到教务处菜单。 |

### 用例编号: 007

| **项目** | **内容** |
| --- | --- |
| **用例名称** | 新增课程信息 |
| **参与者** | 教务处 |
| **用例说明** | 教务处新增课程信息以供学生选择。 |
| **前置条件** | 教务处工作人员已登录系统。 |
| **基本事件流** | 1. 教务处工作人员选择“新增课程”。  2. 教务处工作人员输入课程号、课程名称、上课教师工号等信息。  3. 教务处工作人员确认信息无误后按回车保存。  4. 系统将新课程信息保存到数据库。  5. 系统提示课程新增成功。 |
| **异常事件流** | 3a. 教务处工作人员输入的教师信息不存在，系统提示“找不到该教师”。  4a. 数据库保存时出现故障，系统提示“新增课程失败，请稍后重试”。 |
| **后置条件** | 新课程信息成功添加到系统中，学生可选该课程。 |

（2）

1. **组件图**



1. **组件图含有对象：**

登录组件：数据库对象

学生菜单：学生对象

教师菜单：教师对象

教务处菜单：教务处对象

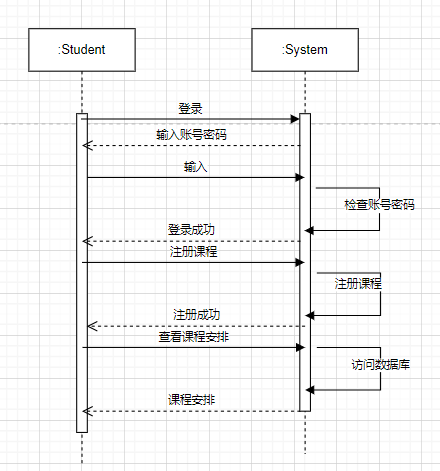
注册课程：学生对象，数据库对象

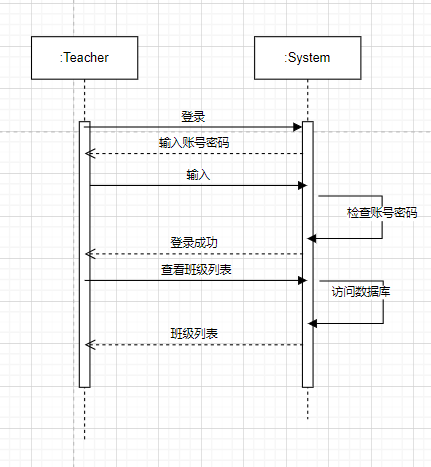
查询课程安排：学生对象，课程对象，数据库对象

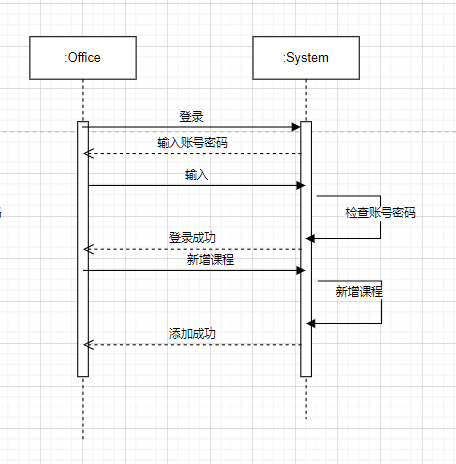
查询班级列表：教师对象，课程对象，班级对象，数据库对象

新增课程信息：课程对象，数据库对象

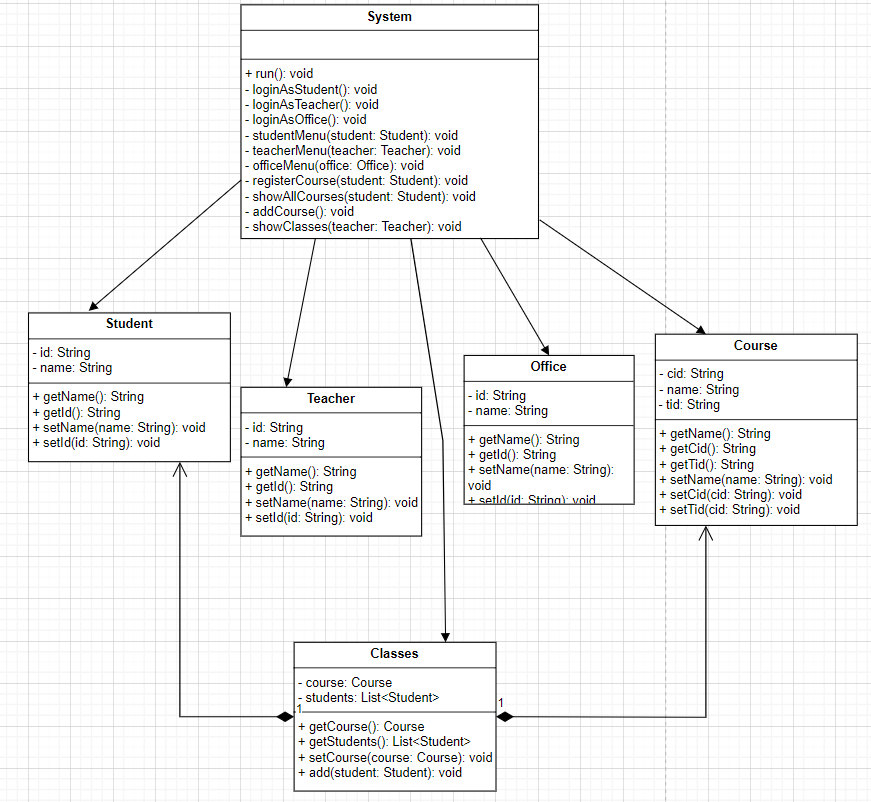
**交互：**

****

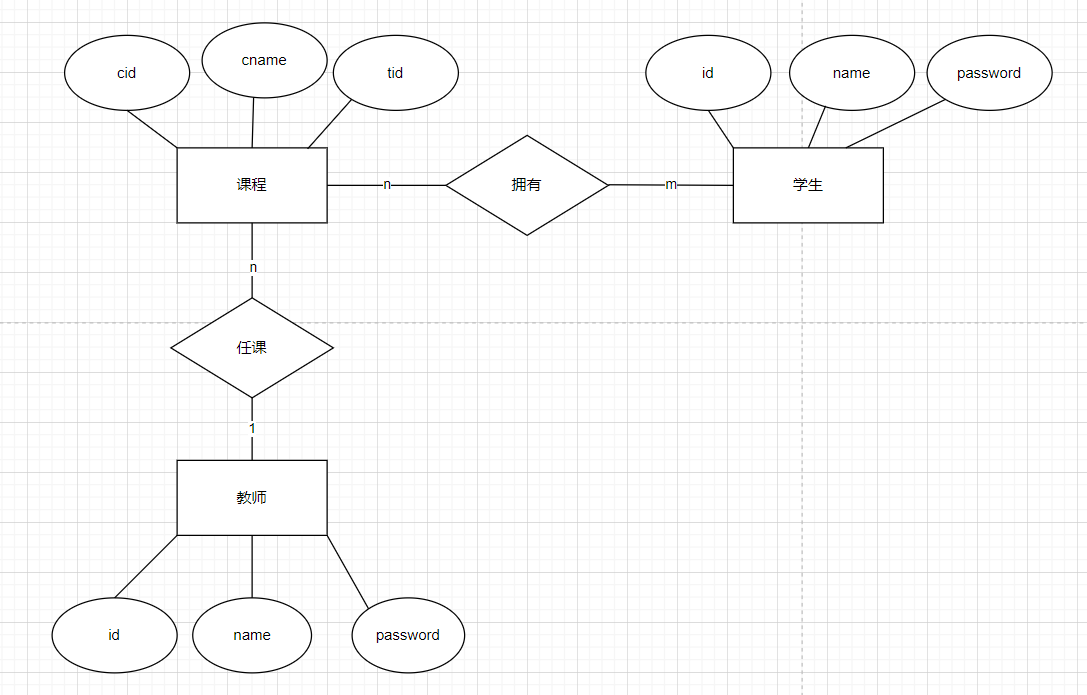
****

****

**UML类图：**

****

1. **数据库ER图**



**数据库建表语句设计**

Student：

1. **CREATE** **TABLE** `student` (
2. `id` **varchar**(255) NOT NULL,
3. `**name**` **varchar**(255) NOT NULL,
4. `**password**` **varchar**(255) NOT NULL,
5. **PRIMARY** **KEY** (`id`)
6. );

Teacher:

1. **CREATE** **TABLE** `teacher` (
2. `id` **varchar**(255) NOT NULL,
3. `**name**` **varchar**(255) NOT NULL,
4. `**password**` **varchar**(255) NOT NULL,
5. **PRIMARY** **KEY** (`id`)
6. );

Office:

1. **CREATE** **TABLE** `office` (
2. `id` **varchar**(255) NOT NULL,
3. `**name**` **varchar**(255) NOT NULL,
4. `**password**` **varchar**(255) NOT NULL,
5. **PRIMARY** **KEY** (`id`)
6. );

Course:

1. **CREATE** **TABLE** `course` (
2. `cname` **varchar**(255) NOT NULL,
3. `cid` **varchar**(255) NOT NULL,
4. `tid` **varchar**(255) **DEFAULT** NULL,
5. **PRIMARY** **KEY** (`cid`),
6. **KEY** `course\_teacher\_id\_fk` (`tid`),
7. **CONSTRAINT** `course\_teacher\_id\_fk` **FOREIGN** **KEY** (`tid`) **REFERENCES** `teacher` (`id`)
8. );

Registration:

1. **CREATE** **TABLE** `registration` (
2. `id` **int** NOT NULL AUTO\_INCREMENT,
3. `cid` **varchar**(255) **DEFAULT** NULL,
4. `sid` **varchar**(255) **DEFAULT** NULL,
5. **PRIMARY** **KEY** (`id`),
6. **KEY** `registration\_Course\_cid\_\_fk` (`cid`),
7. **KEY** `registration\_student\_id\_fk` (`sid`),
8. **CONSTRAINT** `registration\_Course\_cid\_\_fk` **FOREIGN** **KEY** (`cid`) **REFERENCES** `course` (`cid`),
9. **CONSTRAINT** `registration\_student\_id\_fk` **FOREIGN** **KEY** (`sid`) **REFERENCES** `student` (`id`)
10. );
11. **代码实现**：

Classes.java

1. **package** com.krowfeather.entity;
3. **import** java.util.ArrayList;
4. **import** java.util.List;
6. **public** **class** Classes {
7. **private** Course course;
8. **private** List<Student> students;
10. **public** Classes(Course course) {
11. **this**.course = course;
12. students = **new** ArrayList<Student>();
13. }
15. **public** Course getCourse() {
16. **return** course;
17. }
19. **public** **void** setCourse(Course course) {
20. **this**.course = course;
21. }
23. **public** List<Student> getStudents() {
24. **return** students;
25. }
27. **public** **void** add(Student student) {
28. **this**.students.add(student);
29. }
30. }

Course.java

1. **package** com.krowfeather.entity;
3. **public** **class** Course {
4. **private** String cid;
5. **private** String name;
6. **private** String tid;
8. **public** String getCid() {
9. **return** cid;
10. }
12. **public** **void** setCid(String cid) {
13. **this**.cid = cid;
14. }
16. **public** String getName() {
17. **return** name;
18. }
20. **public** **void** setName(String name) {
21. **this**.name = name;
22. }
24. **public** String getTid() {
25. **return** tid;
26. }
28. **public** **void** setTid(String tid) {
29. **this**.tid = tid;
30. }
32. **public** Course(String cid, String name, String tid) {
33. **this**.cid = cid;
34. **this**.name = name;
35. **this**.tid = tid;
36. }
37. }

Office.java

1. **package** com.krowfeather.entity;
3. **public** **class** Office {
4. **private** String id;
5. **private** String name;
7. **public** Office(String id, String name) {
8. **this**.id = id;
9. **this**.name = name;
10. }
12. **public** String getId() {
13. **return** id;
14. }
16. **public** **void** setId(String id) {
17. **this**.id = id;
18. }
20. **public** String getName() {
21. **return** name;
22. }
24. **public** **void** setName(String name) {
25. **this**.name = name;
26. }
27. }

Student.java

1. **package** com.krowfeather.entity;
3. **public** **class** Student {
4. **private** String id;
5. **private** String name;
7. **public** String getId() {
8. **return** id;
9. }
11. **public** **void** setId(String id) {
12. **this**.id = id;
13. }
15. **public** String getName() {
16. **return** name;
17. }
19. **public** **void** setName(String name) {
20. **this**.name = name;
21. }
23. **public** Student(String id, String name) {
24. **this**.id = id;
25. **this**.name = name;
26. }
27. }

Teacher.java

1. **package** com.krowfeather.entity;
3. **public** **class** Teacher {
4. **private** String id;
5. **private** String name;
7. **public** Teacher(String id, String name) {
8. **this**.id = id;
9. **this**.name = name;
10. }
12. **public** String getId() {
13. **return** id;
14. }
16. **public** **void** setId(String id) {
17. **this**.id = id;
18. }
20. **public** String getName() {
21. **return** name;
22. }
24. **public** **void** setName(String name) {
25. **this**.name = name;
26. }
28. }

CourseManageSystem.java

1. **package** com.krowfeather;
3. **import** com.krowfeather.entity.\*;
5. **import** java.sql.ResultSet;
6. **import** java.sql.Statement;
7. **import** java.util.ArrayList;
8. **import** java.util.List;
9. **import** java.util.Scanner;
11. **public** **class** CourseManageSystem {
13. **public** **void** run() {
14. **while** (**true**) {
15. System.out.println("-----欢迎来到课程注册系统-----");
16. System.out.println("1. 教务处登录");
17. System.out.println("2. 学生登录");
18. System.out.println("3. 教师登录");
19. System.out.println("0. 退出");
20. Scanner input = **new** Scanner(System.in);
21. **int** choice = input.nextInt();
22. **switch** (choice) {
23. **case** 1:
24. loginAsOffice();
25. **break**;
26. **case** 2:
27. loginAsStudent();
28. **break**;
29. **case** 3:
30. loginAsTeacher();
31. **break**;
32. **case** 0:
33. System.exit(0);
34. **break**;
35. **default**:
36. System.out.println("INVALID CHOICE");
37. }
38. }
39. }
41. **private** **void** loginAsTeacher() {
42. System.out.println("-----教师登录-----");
43. **boolean** flag = **false**;
44. Scanner input = **new** Scanner(System.in);
45. Teacher teacher = **null**;
46. **do** {
47. String id;
48. String password;
49. System.out.println("请输入id");
50. id = input.nextLine();
51. System.out.println("请输入密码");
52. password = input.nextLine();
53. Statement statement = Database.getStatement();
54. **try** {
55. **if** (statement != **null**) {
56. ResultSet resultSet = statement.executeQuery("select \* from teacher where id = '" + id + "'");
57. **while** (resultSet.next()) {
58. **if** (resultSet.getString("password").equals(password)) {
59. teacher = **new** Teacher(id, resultSet.getString("name"));
60. flag = **true**;
61. }
62. }
63. }
64. } **catch** (Exception e) {
65. System.out.println("登录失败，请稍后重试");
66. e.printStackTrace();
67. }
68. **if** (!flag) {
69. System.out.println("密码或用户名错误！");
70. }
71. } **while** (!flag);
72. System.out.println("-----login successful!-----");
73. teacherMenu(teacher);
74. }
76. **private** **void** teacherMenu(Teacher teacher) {
77. Scanner input = **new** Scanner(System.in);
78. **while** (**true**) {
79. System.out.println("------教师界面------");
80. System.out.println("1. 查看班级列表");
81. System.out.println("0. 退出");
82. **int** choice = input.nextInt();
83. **switch** (choice) {
84. **case** 0:
85. **return**;
86. **case** 1:
87. showClasses(teacher);
88. **break**;
89. **default**:
90. System.out.println("INVALID CHOICE");
91. }
92. }
93. }
95. **private** **void** showClasses(Teacher teacher) {
96. List<Course> userCourses = **new** ArrayList<>();
97. List<Classes> userClasses = **new** ArrayList<>();
98. **try** {
99. Statement statement = Database.getStatement();
100. ResultSet resultSet = statement.executeQuery("select \* from course where tid = '" + teacher.getId() + "'");
101. **while** (resultSet.next()) {
102. userCourses.add(**new** Course(resultSet.getString("cid"), resultSet.getString("cname"), resultSet.getString("tid")));
103. }
104. }**catch** (Exception e) {
105. e.printStackTrace();
106. }
107. **try**{
108. **for**(Course course : userCourses) {
109. Statement statement = Database.getStatement();
110. ResultSet resultSet = statement.executeQuery("select \* from registration where cid = '" + course.getCid() + "'");
111. Classes cls = **new** Classes(course);
112. **while** (resultSet.next()) {
113. Statement statement1 = Database.getStatement();
114. ResultSet resultSet1 = statement1.executeQuery("select \* from student where id = '" + resultSet.getString("sid") + "'");
115. **while** (resultSet1.next()) {
116. cls.add(**new** Student(resultSet1.getString("id"),resultSet1.getString("name")));
117. }
118. }
119. userClasses.add(cls);
120. }
121. }**catch** (Exception e){
122. System.out.println("无法加载班级列表，请稍后重试");
123. e.printStackTrace();
124. }
125. **for** (Classes cls : userClasses) {
126. System.out.println("课程号："+cls.getCourse().getCid()+"，课程名："+cls.getCourse().getName());
127. System.out.println("学生列表：");
128. **for**(Student student : cls.getStudents()) {
129. System.out.println("学号："+student.getId()+"，姓名 "+student.getName());
130. }
131. }
132. }
134. **private** **void** loginAsStudent() {
135. System.out.println("-----学生登录-----");
136. **boolean** flag = **false**;
137. Scanner input = **new** Scanner(System.in);
138. Student student = **null**;
139. **do** {
140. String id;
141. String password;
142. System.out.println("请输入id");
143. id = input.nextLine();
144. System.out.println("请输入密码");
145. password = input.nextLine();
146. Statement statement = Database.getStatement();
147. **try** {
148. **if** (statement != **null**) {
149. ResultSet resultSet = statement.executeQuery("select \* from student where id = '" + id + "'");
150. **while** (resultSet.next()) {
151. **if** (resultSet.getString("password").equals(password)) {
152. flag = **true**;
153. student = **new** Student(id, resultSet.getString("name"));
154. }
155. }
156. }
157. } **catch** (Exception e) {
158. System.out.println("登录失败，请稍后重试");
159. e.printStackTrace();
160. }
161. **if** (!flag) {
162. System.out.println("密码或用户名错误！");
163. }
164. } **while** (!flag);
165. System.out.println("-----login successful!-----");
166. studentMenu(student);
167. }
169. **private** **void** studentMenu(Student student) {
170. Scanner input = **new** Scanner(System.in);
171. **while** (**true**) {
172. System.out.println("-----学生界面-----");
173. System.out.println("1. 注册课程");
174. System.out.println("2. 查看课程安排");
175. System.out.println("0. 退出");
176. **int** choice = input.nextInt();
177. **switch** (choice) {
178. **case** 0:
179. **return**;
180. **case** 1:
181. registerCourse(student);
182. **break**;
183. **case** 2:
184. showAllCourses(student);
185. **break**;
186. **default**:
187. System.out.println("INVALID CHOICE");
188. }
189. }
190. }
192. **private** **void** showAllCourses(Student student) {
193. List<Course> userCourses = **new** ArrayList<>();
194. Statement statement = Database.getStatement();
195. **try** {
196. ResultSet resultSet = statement.executeQuery("select \* from registration where sid = '" + student.getId() + "'");
197. **while** (resultSet.next()) {
198. Statement statement1 = Database.getStatement();
199. ResultSet resultSet1 = statement1.executeQuery("select \* from course where cid = '" + resultSet.getString("cid") + "'");
200. **while** (resultSet1.next()) {
201. userCourses.add(**new** Course(resultSet1.getString("cid"), resultSet1.getString("cname"), resultSet1.getString("tid")));
202. }
203. }
204. } **catch** (Exception e) {
205. e.printStackTrace();
206. }
207. **for** (Course course : userCourses) {
208. Statement statement2 = Database.getStatement();
209. String tname = "";
210. **try** {
211. ResultSet resultSet = statement2.executeQuery("select name from teacher where id='" + course.getTid() + "'");
212. **while** (resultSet.next()) {
213. tname = resultSet.getString("name");
214. }
215. } **catch** (Exception e) {
216. System.out.println("无法加载课程安排，请稍后重试");
217. e.printStackTrace();
218. }
219. System.out.println("课程编号：" + course.getCid() + "，课程名：" + course.getName() + "，教师名：" + tname);
220. }
222. }
224. **private** **void** registerCourse(Student student) {
225. **boolean** flag = **false**;
226. **do** {
227. String cname;
228. String tid;
229. Scanner input = **new** Scanner(System.in);
230. System.out.println("请输入课程名称");
231. cname = input.nextLine();
232. System.out.println("请输入老师工号");
233. tid = input.nextLine();
234. Statement statement = Database.getStatement();
235. **try** {
236. ResultSet resultSet = statement.executeQuery("select \* from course where cname = '" + cname + "' and tid = '" + tid + "'");
237. **while** (resultSet.next()) {
238. String cur\_cid = resultSet.getString("cid");
239. flag = **true**;
240. Statement statement1 = Database.getStatement();
241. statement1.execute("insert into registration (cid, sid) values ('" + cur\_cid + "','" + student.getId() + "')");
242. }
243. } **catch** (Exception e) {
244. System.out.println("选课失败，请稍后重试");
245. }
246. **if** (!flag) {
247. System.out.println("未找到课程");
248. }
249. } **while** (!flag);
250. System.out.println("选课成功");
251. }
253. **private** **void** loginAsOffice() {
254. System.out.println("-----教务处登录-----");
255. **boolean** flag = **false**;
256. Scanner input = **new** Scanner(System.in);
257. Office office = **null**;
258. **do** {
259. String id;
260. String password;
261. System.out.println("请输入id");
262. id = input.nextLine();
263. System.out.println("请输入密码");
264. password = input.nextLine();
265. Statement statement = Database.getStatement();
266. **try** {
267. **if** (statement != **null**) {
268. ResultSet resultSet = statement.executeQuery("select \* from office where id = '" + id + "'");
269. **while** (resultSet.next()) {
270. **if** (resultSet.getString("password").equals(password)) {
271. office = **new** Office(resultSet.getString("id"), resultSet.getString("name"));
272. flag = **true**;
273. }
274. }
275. }
276. } **catch** (Exception e) {
277. System.out.println("登录失败，请稍后重试");
278. e.printStackTrace();
279. }
280. **if** (!flag) {
281. System.out.println("密码或用户名错误！");
282. }
283. } **while** (!flag);
284. System.out.println("-----login successful!-----");
285. officeMenu(office);
286. }
288. **private** **void** officeMenu(Office office) {
289. Scanner input = **new** Scanner(System.in);
290. **while** (**true**) {
291. System.out.println("------教务处界面------");
292. System.out.println("1. 新增课程信息");
293. System.out.println("0. 退出");
294. **int** choice = input.nextInt();
295. **switch** (choice) {
296. **case** 0:
297. **return**;
298. **case** 1:
299. addCourse();
300. **break**;
301. **default**:
302. System.out.println("INVALID CHOICE");
303. }
304. }
305. }
307. **private** **void** addCourse() {
308. Scanner input = **new** Scanner(System.in);
309. **boolean** flag = **false**;
310. System.out.println("-----添加课程-----");
311. **do** {
312. System.out.println("请输入课程号：");
313. String cid = input.nextLine();
314. System.out.println("请输入课程名：");
315. String courseName = input.nextLine();
316. System.out.println("请输入任课教师工号：");
317. String tid = input.nextLine();
318. Course newCourse = **new** Course(cid, courseName, tid);
319. Statement statement = Database.getStatement();
320. **try** {
321. ResultSet resultSet = statement.executeQuery("select \* from teacher where id = '" + tid + "'");
322. **while** (resultSet.next()) {
323. **if** (resultSet.getString("id").equals(tid)) {
324. newCourse.setTid(tid);
325. Statement statement1 = Database.getStatement();
326. statement1.execute("insert into course (cid, cname, tid) values ('"+newCourse.getCid()+"','"+newCourse.getName()+"','"+newCourse.getTid()+"')");
327. flag = **true**;
328. }
329. }
330. } **catch** (Exception e) {
331. System.out.println("新增课程失败，请稍后重试");
332. e.printStackTrace();
333. }
334. **if** (!flag) {
335. System.out.println("找不到该教师");
336. }
337. } **while** (!flag);
338. System.out.println("添加成功！");
339. }
340. }

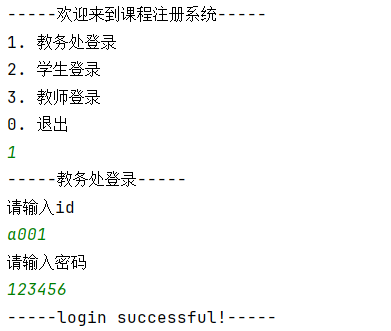
Database.java

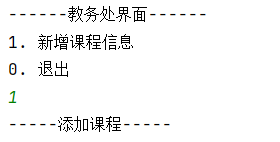
1. **package** com.krowfeather;
3. **import** java.sql.Connection;
4. **import** java.sql.DriverManager;
5. **import** java.sql.SQLException;
6. **import** java.sql.Statement;
8. **public** **class** Database {
9. **private** **static** **final** String USER = "root";
10. **private** **static** **final** String PASSWORD = "123456";
11. **public** **static** Connection getConnection() {
12. Connection con = **null**;
13. **try** {
14. Class.forName("com.mysql.cj.jdbc.Driver");
15. } **catch** (ClassNotFoundException e) {
16. e.printStackTrace();
17. }
18. **try** {
19. con = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", USER, PASSWORD);
20. } **catch** (SQLException e) {
21. e.printStackTrace();
22. }
23. **return** con;
24. }
25. **public** **static** Statement getStatement() {
26. **try** {
27. Connection connection = getConnection();
28. **return** connection.createStatement();
29. } **catch** (SQLException e) {
30. e.printStackTrace();
31. }
32. **return** **null**;
33. }
34. }

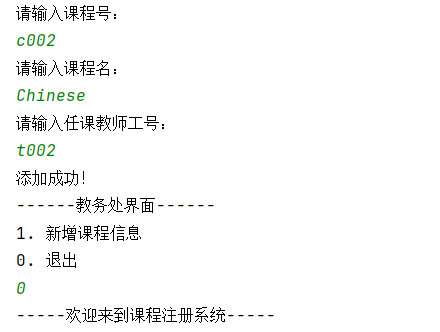
Main.java

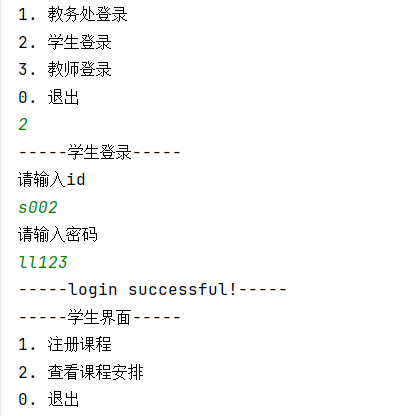
1. **package** com.krowfeather;
3. **public** **class** Main {
4. **public** **static** **void** main(String[] args) {
5. CourseManageSystem courseManageSystem = **new** CourseManageSystem();
6. courseManageSystem.run();
7. }
8. }

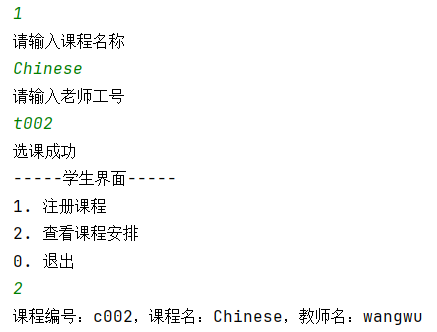
**测试：**

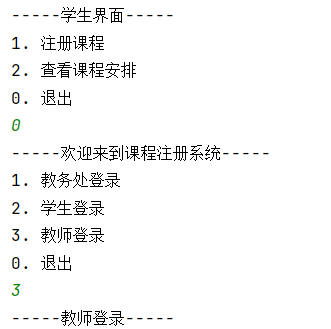
****

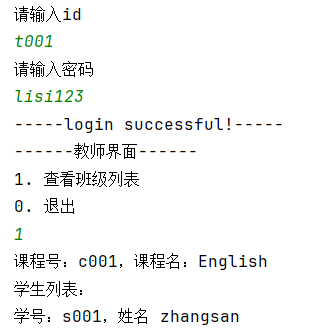
****

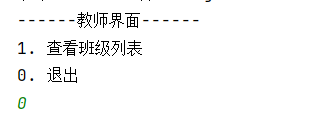
****

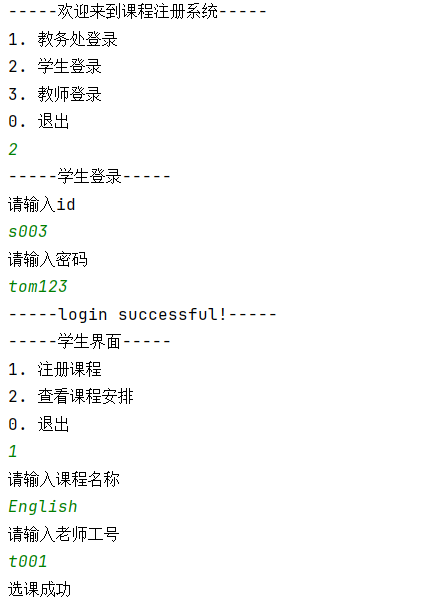
****

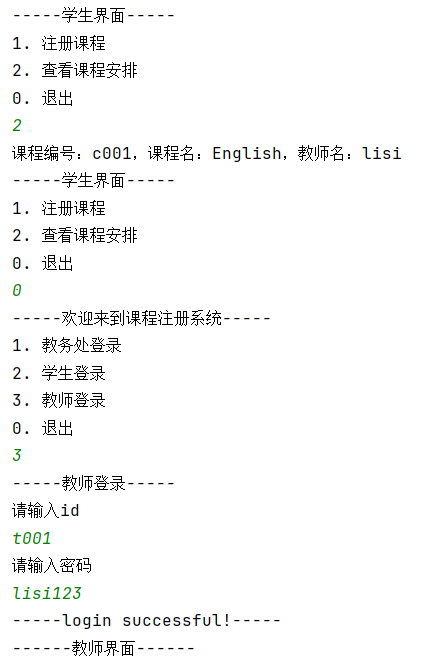
****

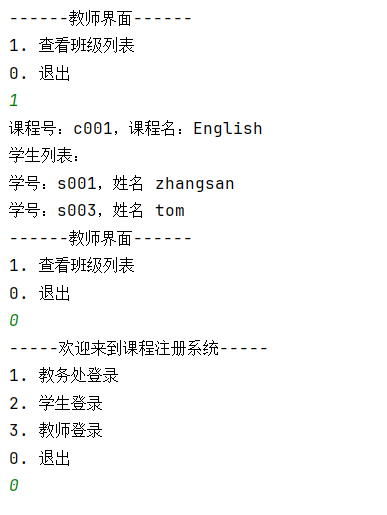
****

****

****

****

****

****