**中国矿业大学计算机学院**

**2020 级本科生实验报告**

课程名称 软件工程实践

报告时间 2022-12-6

学生姓名 李建业

学 号 12203743

专 业 计算机科学与技术

任课教师 刘迎春

# 实验（二）面向对象的软件分析与设计

实验目的

学习UML和UML软件工具，掌握面向对象分析与设计方法。

实验步骤

熟悉软件操作：在StarUML中默认打开的“toolbox”工具中就是类相关的一下基础功能组件，以及组件的功能简介基础组件中有类实例，以及描述类的各个功能组件。如图所示。

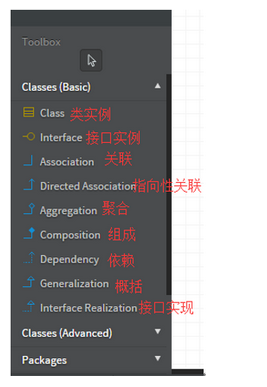


图1.类实例的功能组件

单击选择Class，至工作区域内单机左键，即可创建一个class实例。创建实例后，周围会出现一些功能按钮，如图所示。

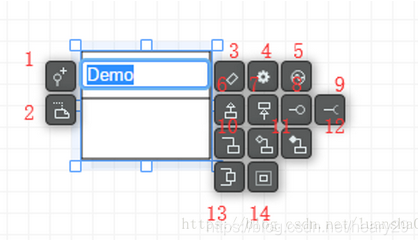


图2. 功能按钮

功能介绍：

1. 用来标识Class的可见性（默认为public. 2. 用来添加note的，比如：类的说明 3. 增加类的属性 4. 增加类的操作方法。 5. 增加Reception 6. 增加子类 7. 增加父类 8. 添加已有的接口 9. 添加需要的接口 10. 添加关联 11. 添加聚合 12. 添加组合 13. 添加端口14. 添加部件

## 基于UML对目标系统进行分析

### 系统用例图

基于作业14学生注册课程系统的用例图，本实验将绘制一个与之类似，较为完备的选课系统的用例图。

首先根据需求分析可知：**管理员**维护课程信息，对其进行增删改查等操作。**学生**可以在线查询课程信息，并进行选课，也可以在规定时间内更改选修的课程。我们发现系统中的参与者有：管理员和学生，然后从参与者的角度就可以发现系统的用例，并绘制出系统的用例图，如图1所示：

为了方便后期编程和双向工程，usecase图中所有的文字选用了英文。

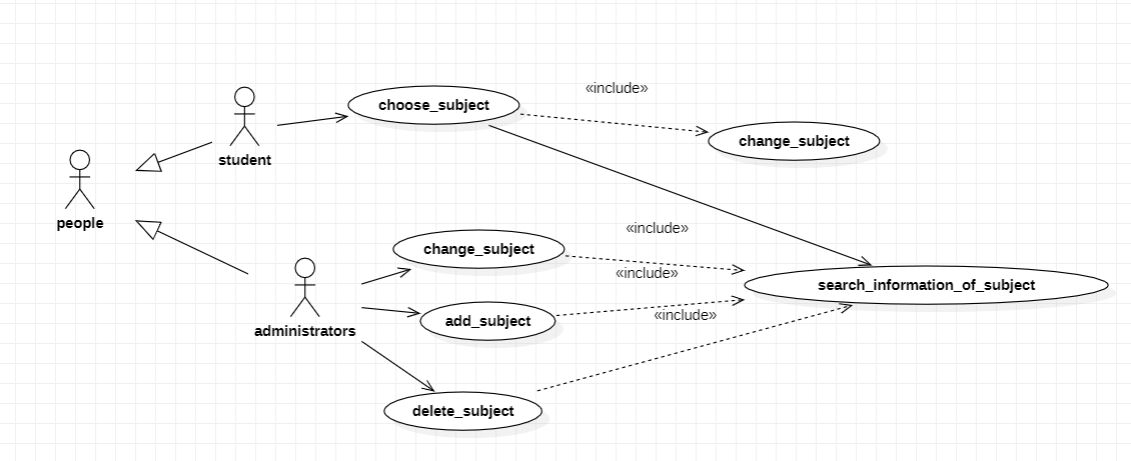


图3. Usecase图

2.对部分用例进行描述：

**“添加课程”用例(add\_subject)**

用例名：添加课程

执行者：管理员

目的：管理员通过系统界面进入，添加所要开设的课程，确认无误后将其信息保存到数据库中，以供学生选择。

过程描述：

1.管理员选择进入管理界面，用例开设

2.系统提示输入管理密码

3.管理员输入密码

4.系统验证密码

5.A1:密码错误

6.进入管理界面，系统显示目前所建立的全部课程信息

7.管理员选择添加课程

8.系统提示输入新课程信息

9.管理员输入信息

10.系统验证是否和已有的课程冲突

11.A2：有冲突

12.系统添加新课程，提示课程添加成功

13.系统重新进入管理界面，显示所有课程

14.用例结束

**异常事件流处理：**

A1：密码错误 ：1.系统提示再次输入。2.用户确认后进入第5)步。

A2：有冲突 ：1.系统提示冲突，显示冲突的课程信息。2.用户重新输入，验证无误后进入第6步。

**“选课”用例(choose subject)**

用例名：选课

执行者：学生

目的：学生进入选课系统界面，浏览的课程，最后选择一门自己喜欢的课程并提交。

过程描述：

1.学生进入选课登录界面，用例开始

2).系统提示输入学号与密码

3.学生输入学号与密码

4.系统验证

A1：验证错误

5.进入选课主界面

6.学生点击选课

7.系统显示所有课程信息

8.学生选择课程

9.系统验证课程是否可选

A2：不可选

10.系统提示课程选择成功

11.用例结束

异常事件流处理：

A1：验证错误： 1.系统提示验证错误，提示重新输入。2.验证成功，进入第5.步

A2：不可选 1.系统提示课程不可选及原因。2.学生重新选课。3.验证成功后进入第10步

**“修改”用例(change\_subject)**

管理员选择进入管理界面，用例开设

系统提示输入管理密码

管理员输入密码

系统验证密码

A1:密码错误

进入修改主界面，系统显示目前所建立的全部课程信息

管理员选择要修改的课程

是否确定修改该课程

A2：不修改

系统显示全部课程信息

管理员输入新信息

验证是否有课程冲突

A3：冲突

系统修改课程，提示修改成功

系统重新进入主界面，显示所有课程

用例结束

异常事件流处理：

A1:密码错误 ：1.系统提示再次输入。2)用户确认后进入第5)步。

A2：不修改：1) 系统提示重新选择。2)用户确认后进入第8)步。

A3：有冲突 ：1)系统提示冲突，显示冲突的课程信息。2.用户重新输入，验证无误后进入第11.步。

### 类图

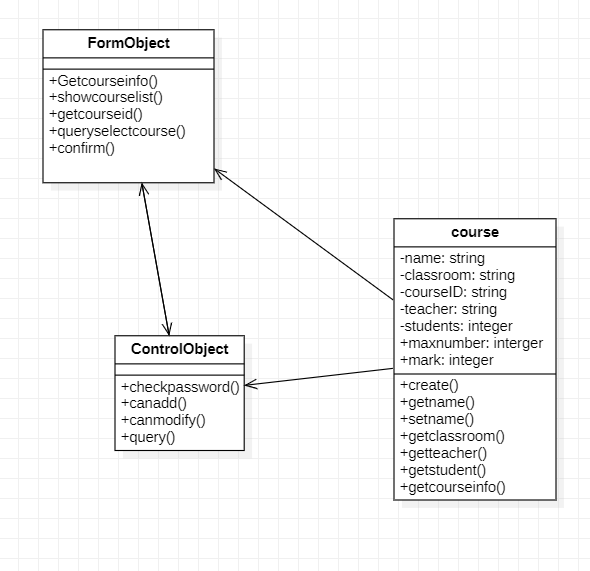


图4.类图

## 双向工程

需要下载java拓展

To install an extension manually:

1.Get an extension as a .zip file.

2.Extract the file on the extensions path.

/Users/<user>/Library/Application Support/StarUML/extensions/user for Mac OS.

C:\Users\<user>\AppData\Roaming\StarUML\extensions\user for Windows.

~/.config/StarUML/extensions/user for Linux.

3.Check a folder in the extensions path and :file:main.js file in the folder.

4.Restart StarUML.

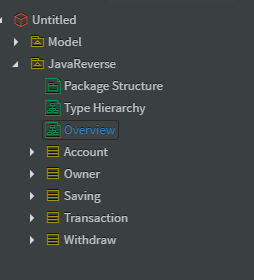
因为系统不让下载java扩展，找了一下官方的官网和文档，去github下载到了本地，重启startuml得到了正确的拓展

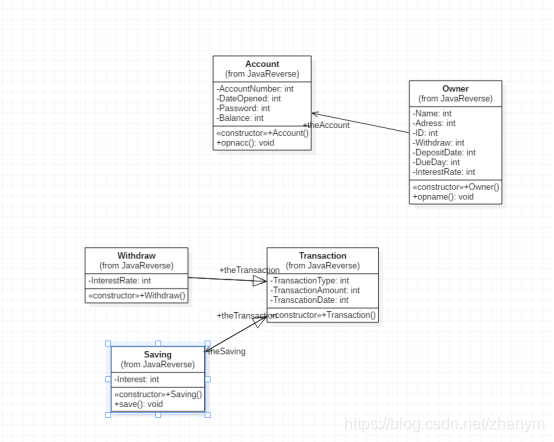
### 逆向工程（因自己创的工程与startuml不匹配，这里引用别人的项目文件）

**步骤一**：通过 StarUml 建立一个新的 Logial View。

**步骤二**：点击上端菜单栏中的 Tools 下拉菜单 Java—Reverse Code并选择代码所在的文件夹。

**步骤三**：左侧树形机构中出现代码中的类





**步骤五**：保存mdj文件

### 正向工程

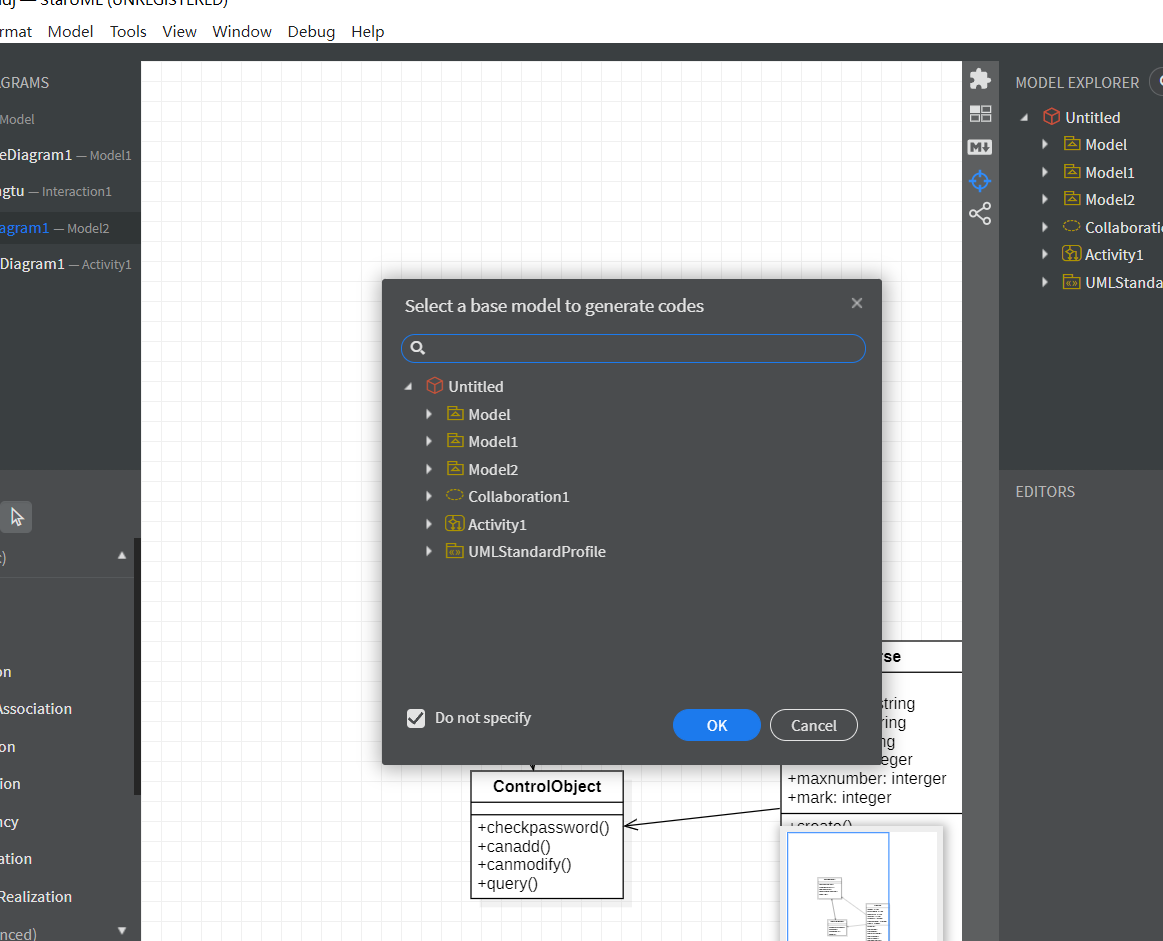


图4.2.1正向工程图

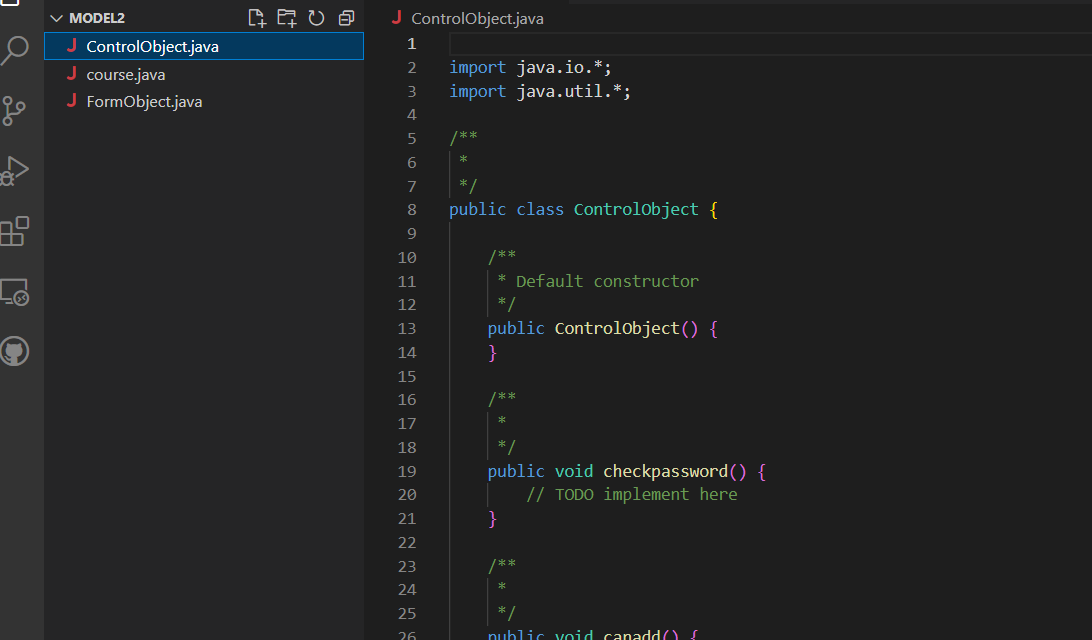


图4.2.2 正向工程图2

得到的结果与类图一致。

## 剩余图形绘制

### 系统活动图

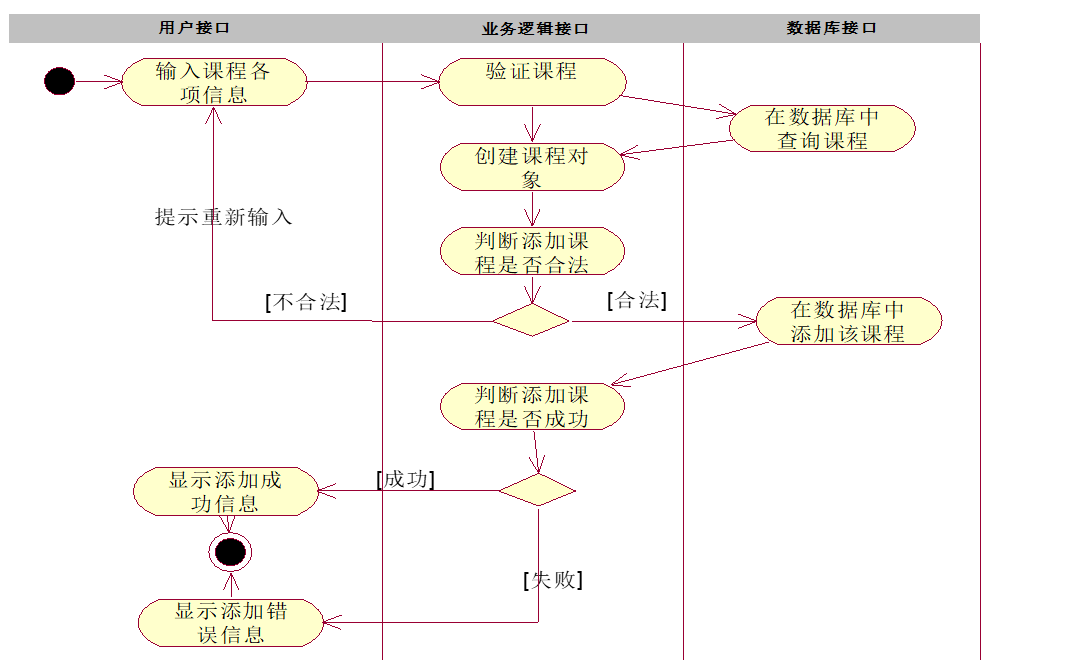


图5.1系统活动图

### 序列图（顺序图）

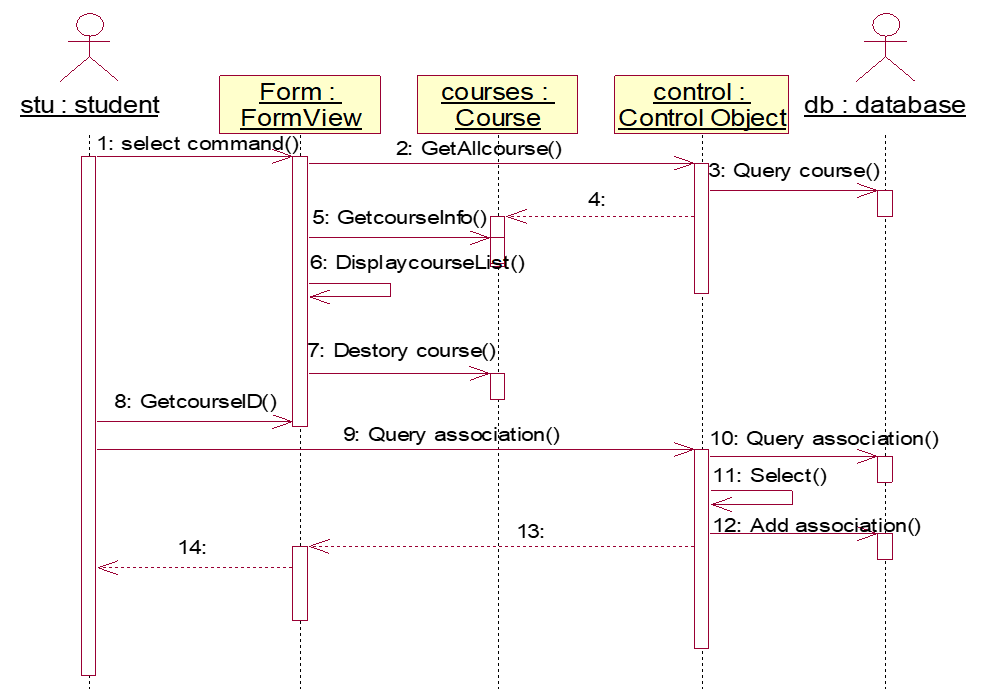


图5.2选择课程顺序图

### 状态图

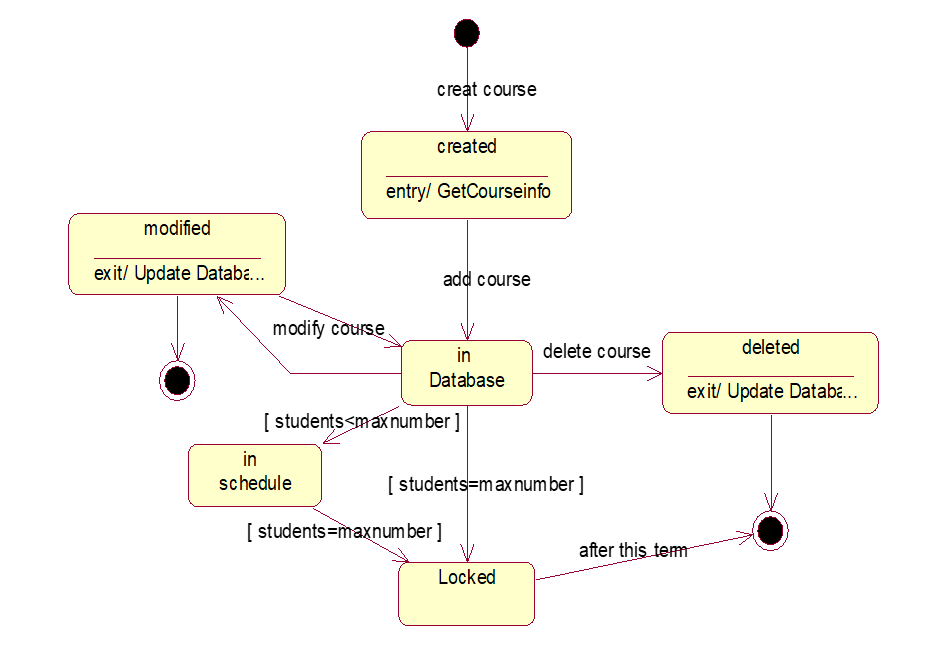
****

图5.3状态图

## 实验体会

经过这次实验，我学习了如何使用Startuml绘制类图，usecase图，活动图等，感受到startuml带来的面向对象开发带来的便捷。通过正向工程能够快速将类图转化为代码框架，通过反向工程能够快速浏览学习这个工程的大致内容。Startuml这一工具对今后的软工学习非常有帮助。