Java课程设计报告

学生成绩管理系统的设计与实现

19308024崔子潇 19335040丁维力 15331045陈振宇

1. 项目背景介绍

使用面向对象思想，以一个Java可运行程序为主体，设计和实现一个给教师和学生使用的学生成绩管理系统。

教师可以使用这个系统查看、管理自己负责的课程和课程中的学生成绩。

学生可以使用这个系统查询自己的成绩。

1. 系统功能介绍
2. 教师账号/学生账号的注册功能。



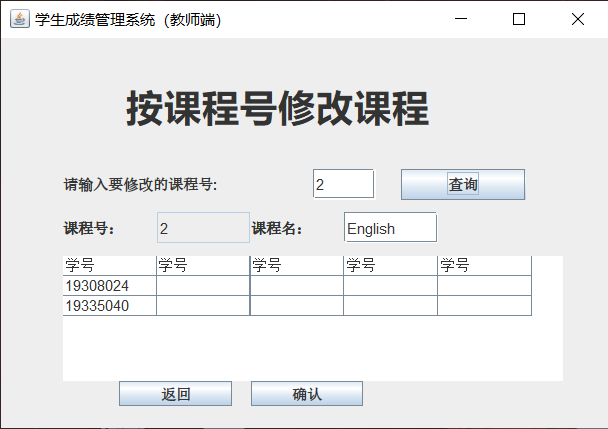
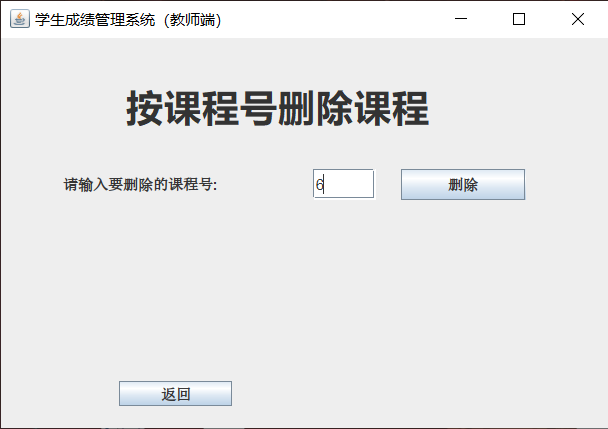
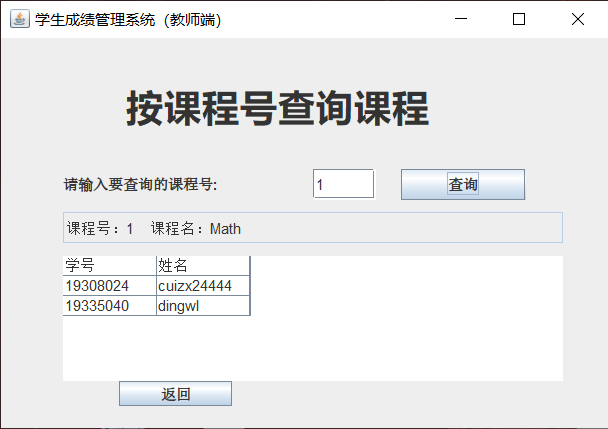
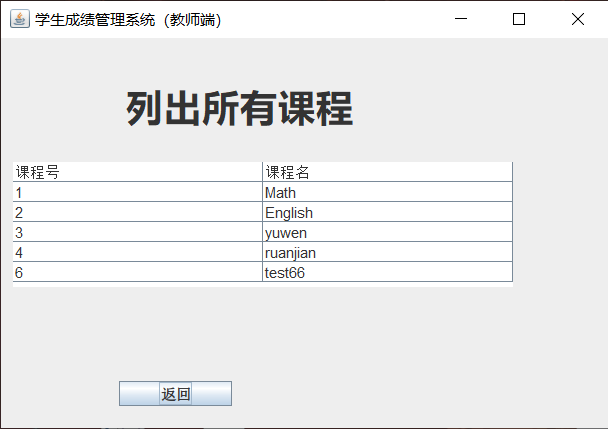
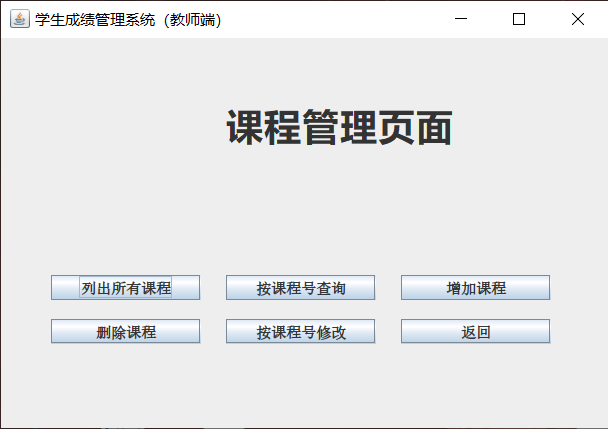
1. 教师账号/学生账号的登录功能。



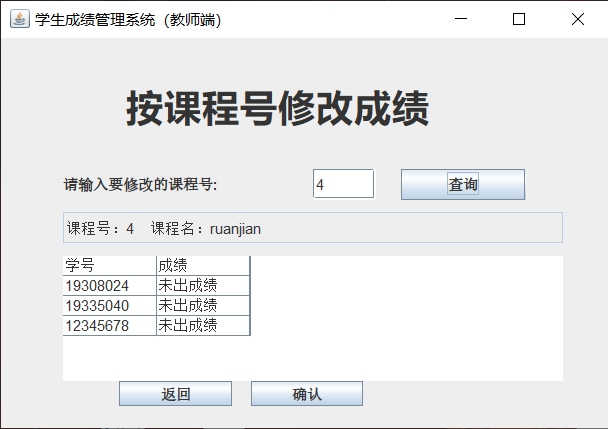
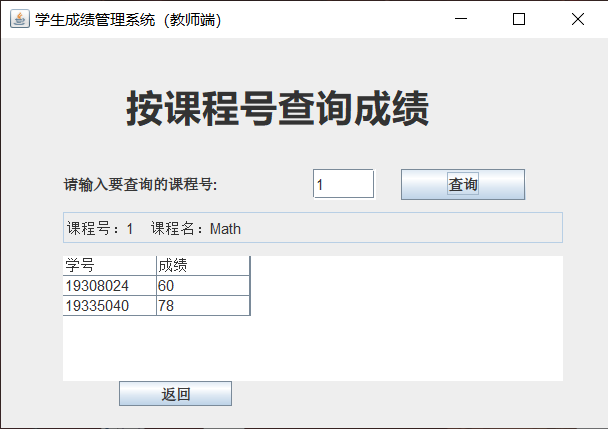
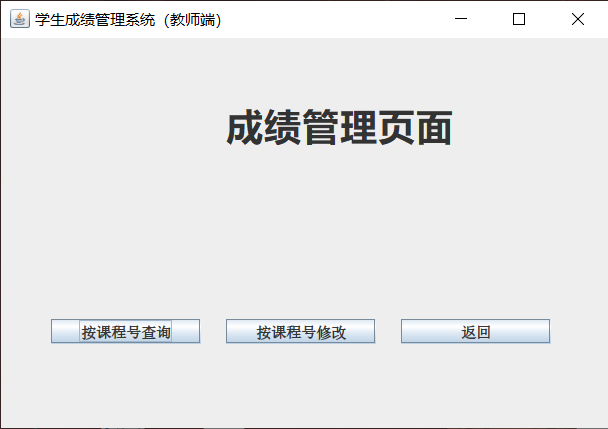
1. 教师账号，可使用修改个人信息功能、课程管理功能和成绩管理功能。



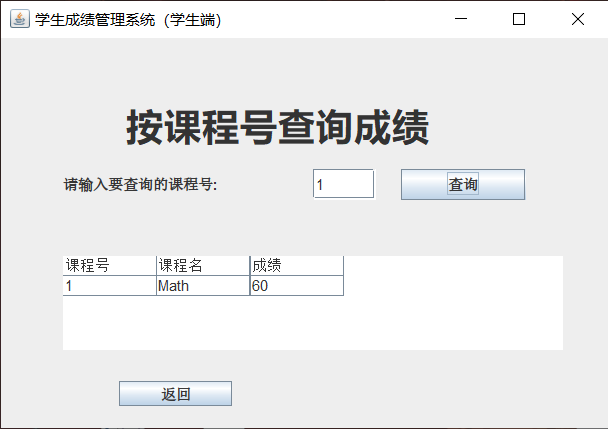
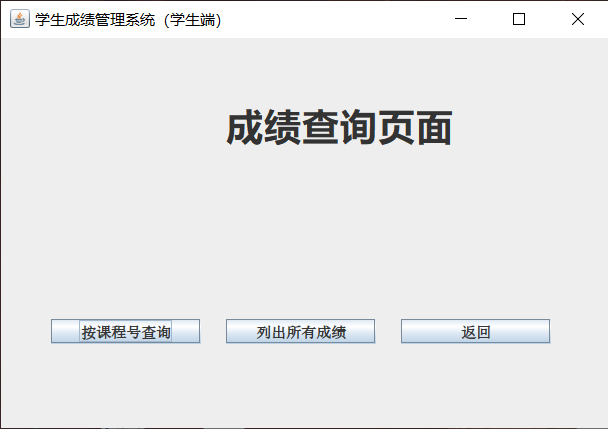
1. 教师账号使用课程管理功能，课程管理功能包括列出所有课程、按课程号查询课程、增加课程、删除课程、按课程号修改课程功能。



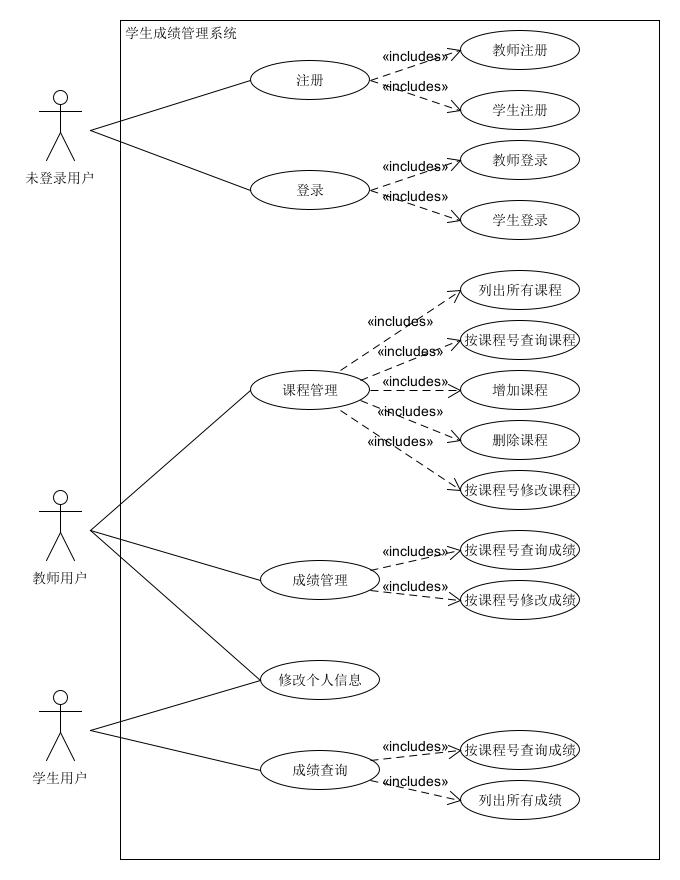
1. 教师账号使用成绩管理功能，其中包括按课程号查询成绩、按课程号修改成绩功能。



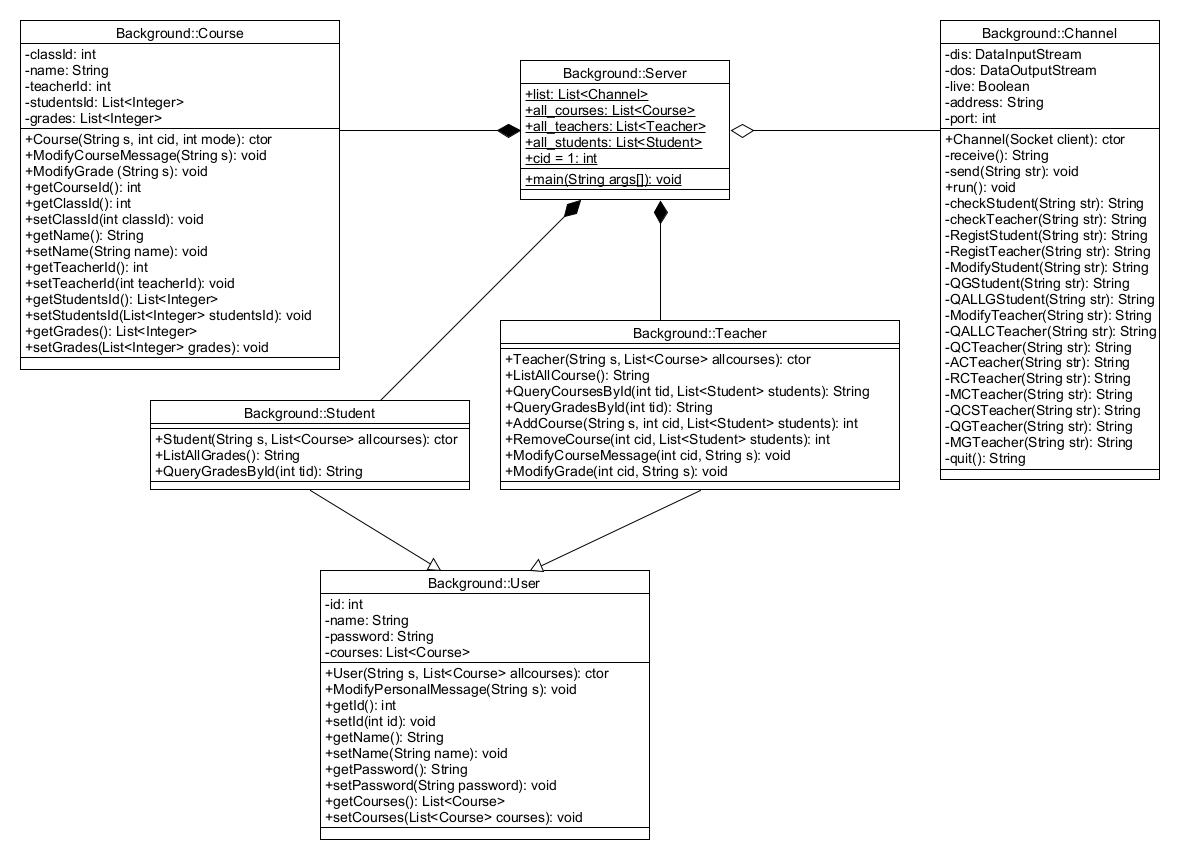
1. 学生账号，可使用修改个人信息功能和成绩查询功能，成绩查询功能包括按课程号查询成绩、列出所有成绩功能。



1. 用例图



1. 系统类图
2. Server端类图



Server类：代表服务器端，包含连接线程列表、学生账号列表、教师账号列表、课程列表。初始化时会从硬盘中的txt文件加载上述数据

Channel类：代表用于与客户端连接的线程，一个Channel负责和一个客户端的连接（双向）。接收客户端传来的String请求，解析并执行请求，并通过String返回执行结果给客户端。

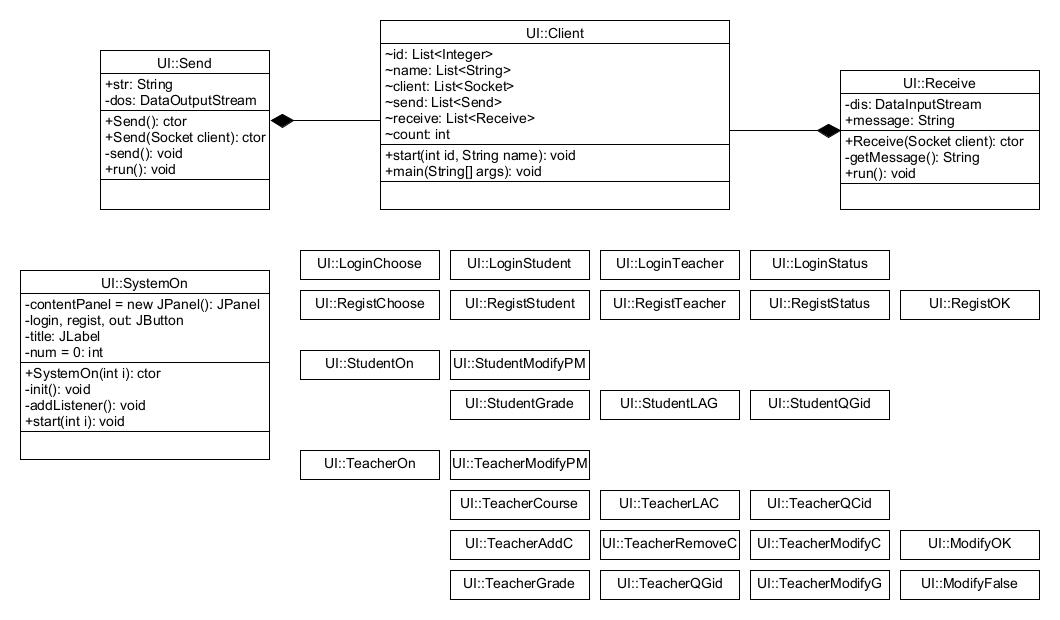
User类：代表用户账号，包含账号id、密码、账号名、参与的课程列表。

Student类：代表学生账号，派生于User类，包含学生按课程号查询成绩、列出所有成绩功能的方法。

Teacher类：代表教师账号，派生于User类，包含教师列出所有课程、按课程号查询课程、增加课程、删除课程、按课程号修改课程功能、按课程号查询成绩、按课程号修改成绩功能的方法。

Course类：代表课程，包含课程id、课程名、任课教师id、上课学生id列表和对应学生成绩列表，包含修改课程信息、修改课程成绩的方法。

1. Client端类图



Client类：代表客户端，包含本地账户id、账户名、发送的连接线程和接受的连接线程。

Send类：代表发送信息给服务器端的连接线程。

Receive类：代表接受来自服务器端信息的连接线程。

SystemOn类和其他类：分别各自代表GUI中的一个界面。比如SystemOn类代表系统主界面。都派生于Jframe类。

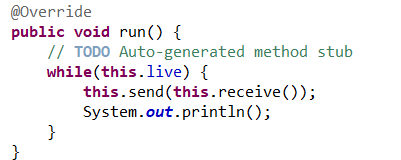
1. 关键模块说明。
2. Server对象即服务器端初始化时，从硬盘中的course.txt等文件中读入课程、账号等数据。



然后创建ServerSocket对象，监听是否有客户端发起连接，若有则建立连接，1个客户端的连接由1个连接线程负责。



1. Channel类即服务器端的连接线程持续监听来自客户端的String请求，处理后返回结果给客户端。



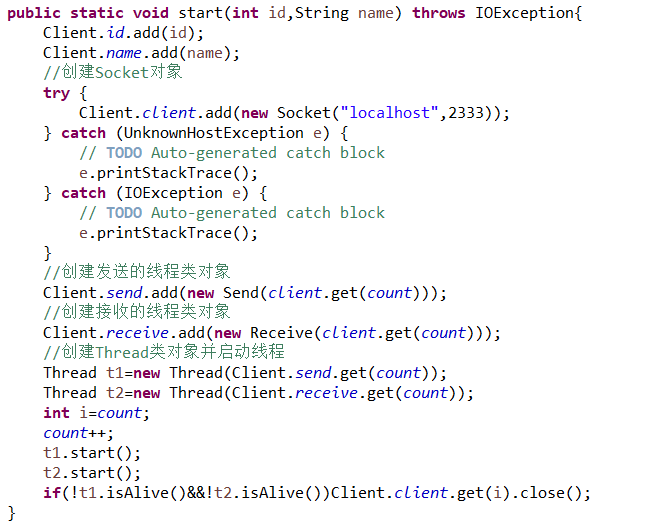
1. Channel类的Receive()方法中，接受到来自客户端的String请求后，通过解析String中的操作码，识别为对应类型的请求，并使用对应的方法执行。



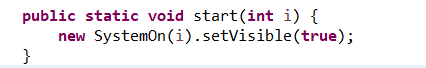
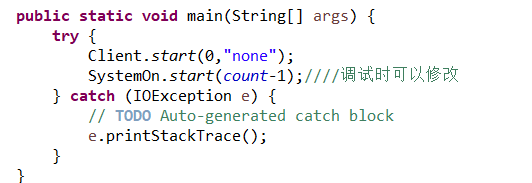
以操作码为1时，执行checkStudent()方法为例，该方法对应查询/登录学生账号。此方法继续解析客户端的String请求中剩余的子字符串，通过”;”号分解为账号id和密码，然后在Server类中的学生账号列表中进行遍历对比。该方法返回的String为执行结果，之后会由Channel的Send()方法发送回客户端。



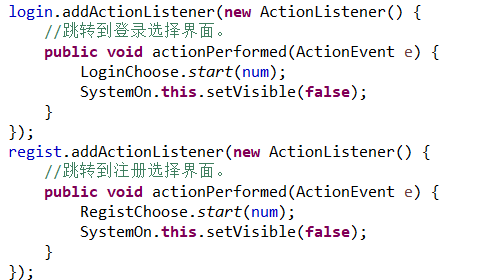
1. Client对象即客户端初始化时，创建1个Send对象和1个Receive对象，分别尝试与服务器端建立发送、接收连接。



此外在客户端的main函数中，也会把systemOn类，即表示客户端GUI主界面的类实例化，从而初始化GUI界面。



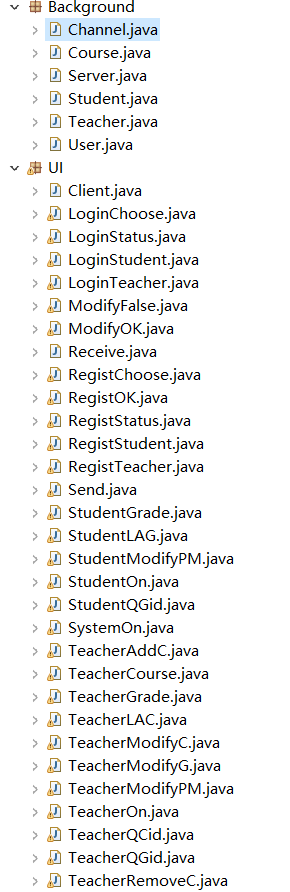
1. 代表GUI界面的类，在监听到用户点击跳转界面按钮事件，需跳转到其他界面时，通过执行代表对应的该界面的类的start()方法，和执行自身的setVisible(false)方法，来实现GUI界面之间的跳转。下面附图是主界面跳转到登录选择界面、注册选择界面的示例。



1. 代表GUI界面的类，在监听到用户点击需发送到服务器请求的按钮事件时，通过了前端的输入校验后，使用特定的格式（如在字符串开头使用数字操作码，以”;”分割信息）组成String，赋值给Send对象的发送缓存，Send对象会将该String发送给服务器端。之后等待Receive对象的接收缓存接受到服务器端返回的String，即服务器端的执行结果，再根据该结果执行对应的操作。下面附图是学生登陆界面中学生登录涉及到的方法的示例。



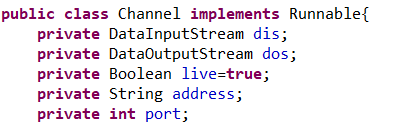
1. 知识点应用说明
2. 类和对象。本课程设计充分使用了面向对象思想，把相关性高的数据或者功能模块抽象、封装成为类。现实事物抽象成为类的例子有User类，Student类，Course类等，功能模块抽象为类的例子有Channel类、Receive类、Send类和各种GUI类等。



1. 超类与继承。比如Student类和Teacher类继承于User类，继承了父类User类的id、name等变量，又有各自独特的子类方法。



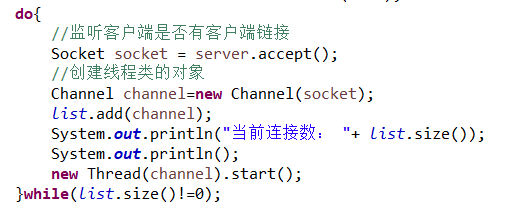
1. 接口及其实现。比如Chanel类、Send类和Receive类都是Runnable的实现。



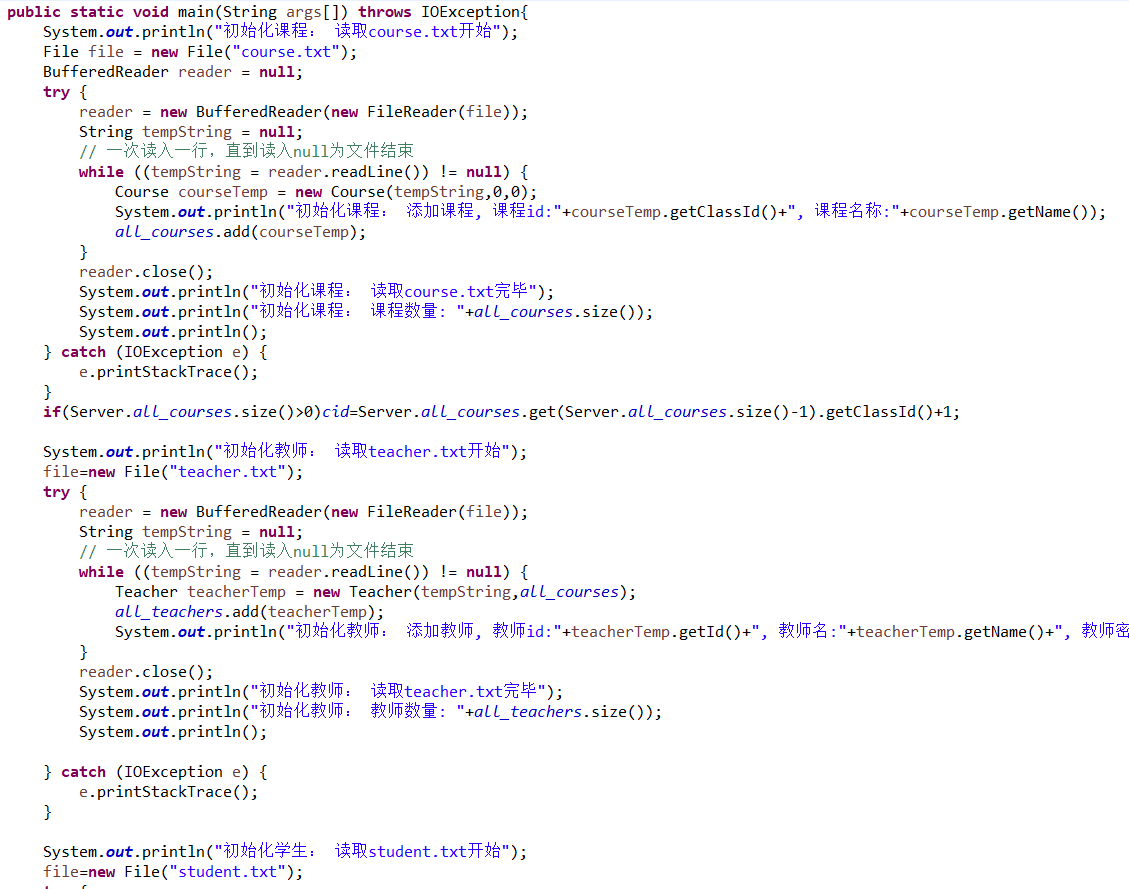
1. 异常处理。比如在Channel类的send()方法中，有对于IOException的异常处理，在出现异常时终止本线程。



1. 多线程。在服务器端，1个Channel线程负责与1个客户端的连接。多个客户端连接服务器端时，服务器端可多线程处理这些客户端的请求。



1. 文件存储。服务器端初始化时从文件中读取数据。

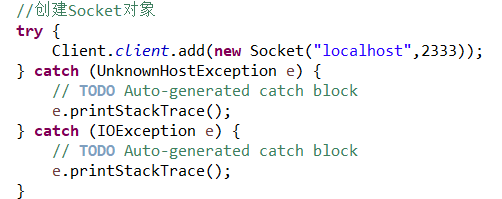


有服务器端的连接线程退出时把数据持久化储存到文件中。



1. 网络编程。服务器端和客户端通过网络通信，通信内容使用String结合特定的格式。

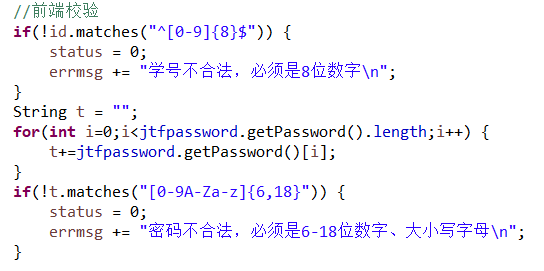




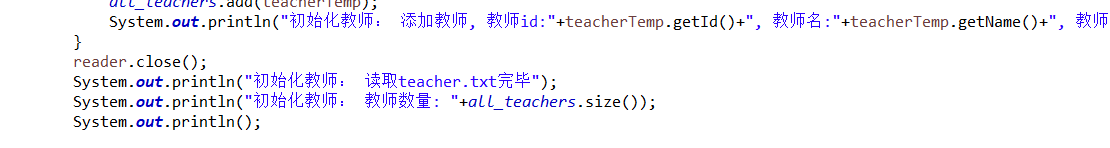
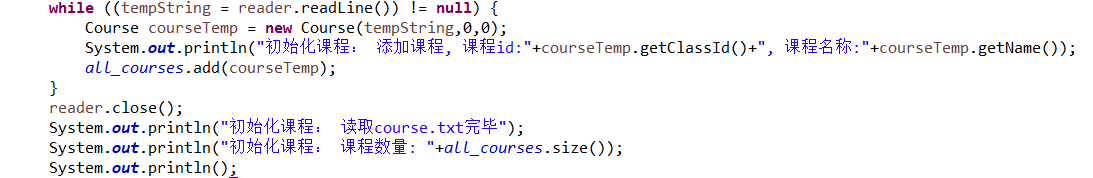
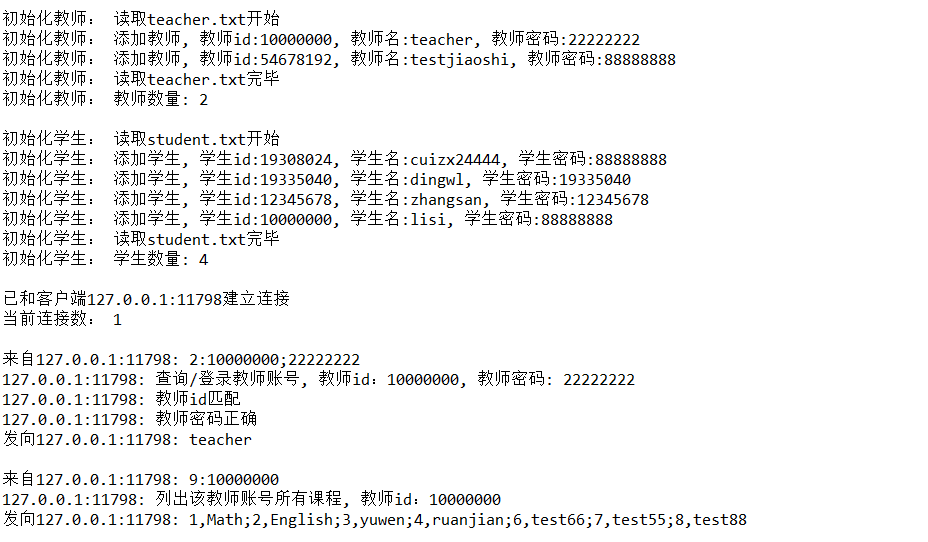
1. Java 图形界面。使用了awt和swing构建图形界面。以主界面为例。



1. 创新点或技术难点说明
2. 在客户端设计了前端输入校验，降低了服务器端的负载。比如在学生登录界面中，设计了对于学生id和密码格式的输入校验。



1. 在服务器端设计了在控制台的日志输出，记录了服务器端的行为，方便服务器端运维人员调试和维护。

1. 未解决的问题或者可以改进的地方。
2. 使用awt和swing实现的图形界面较为简陋，不够美观，可以考虑使用JavaFX令图形界面更美观。
3. 没有使用关系型数据库，可以考虑使用jdbc连接mysql数据库作为持久化储存方案。
4. 功能不多，可以考虑增加更多功能，比如聊天功能、教师对于学生的广播/公告功能。
5. 组员分工情况

19308024崔子潇：需求分析与设计，前端，基本类的实现。

19335040丁维力：后端逻辑设计，网络编程，多线程

15331045陈振宇：后端逻辑设计，测试，课堂展示，实验报告