Design

Interfaces(2): AdminInterface, StudentInterface

Classes(5): User, Admin, Student, Course, Login

StudentInterface

AdminInterface

Abstract

Superclass

Interface

User

Admin

Student

Subclass

Subclass

Interface

have objects

Entry Point

Login

Course

Additional Features

1. Program incorporates input validation. In menu, if user enters invalid input, a note prints and programs jumps back to menu. For example, if user enters a wrong course id/section, program will print “Course not found” and jump back to menu. All options will jump back to menu, and user may only exit by calling exit().

2. Program checks for duplicated courses and students. For example, if student tries to register in the same class twice, it will not be allowed and program will jump back to menu.

3. There are some security checks when admin edits a course. For example, if the admin changes max# of students to be smaller than current number of actual students, the action will not be allowed, and program prints error. In addition, although user is allowed to change the current number of students, warnings are designed for different situations (current vs. max, current vs. actual…) since it could be dangerous to the program.

4. Program also serializes Admin to check if an admin account exists. If not, registers.

5. There are also effects such as a login and loading page. In this way, the program offers clear guidance and also obtains a more nostalgic style.

Workflow

Login.java

Automatically reads from .csv or .ser

Entry Point

Asks user to identify as admin/student

If input = admin/Admin

If input = Student/student

Register admin

If account exists

If first time

input username and password,

if correct then login successful

For all inputs:

show student menu. Asks input

show admin menu. Asks input

input 2:

Reports

input 1:

Course management

Inputs

1. View all courses

2. View courses not full

3. Register in a course

4. Withdraw

5. View registered courses

6. Exit

Inputs

1. Create a course

2. Delete a course

3. Edit a course

4. Display info by id

5. Register a student

6. Exit

Inputs

1. View all courses

2. View full

3. Write full to file

4. View students in course

5. View course of students

6. Sort + print

7. Exit

Serialize.

Log out.

If input is exit

(as an int)

If input is not exit

(as an int)

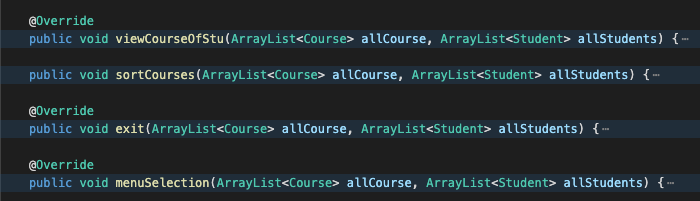
Goes to corresponding functions

OOP Concepts

1. Method overriding

in Student and Admin





2. Abstract class



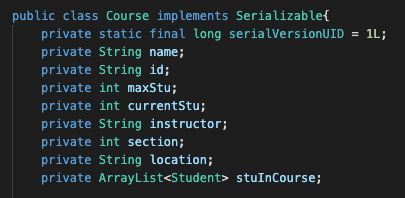
3. Inheritance

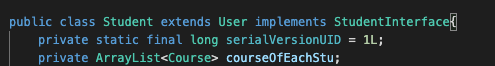


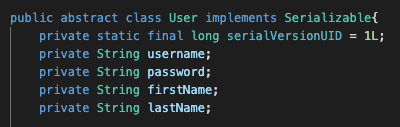


4. Encapsulation

Private data fields in User, Student, Course (Admin has no extra field)







5. ADT

Code uses ArrayList. List specifies the abstract data and ArrayList is its concrete implement

Codes Referenced

https://beginnersbook.com/2013/12/how-to-serialize-arraylist-in-java/

https://www.journaldev.com/2335/read-csv-file-java-scanner

https://stackoverflow.com/questions/18672643/how-to-print-a-table-of-information-in-java

https://www.geeksforgeeks.org/object-serialization-inheritance-java/

https://www.w3schools.com/java/java\_files\_create.asp