

Further Programming (COSC2440)

Assignment 1: Build a console app

Lecturer: Minh Vu Thanh



Table of contents:

1. Introduction
2. Classes and Object
3. Functions
4. Diagram
5. Github

1. Introduction:

In general, my application includes 3 main classes: Student, Course and StudentEnrolment. Moreover, there are an interface class and the main file namely StudentEnrolmentManager and EnrollmentSystem. The Student class contains 3 pieces of information about its object including sID, sName and sBd (date of birth) as well as the Course class, it also contains 3 main informations about its object namely cID, cName and cCredits. Besides, StudentEnrolment class contains 2 properties from those classes namely student and course with another property called semester.

Student class:

```
public class Student {
    private String sID;
    private String sName;
    private String sBd;

    public Student(String sID, String sname, String sbd) {
        this.sID = sID;
        this.sName = sname;
        this.sBd = sbd;
    }

    public String getsID() { return sID; }

    public String getsName() { return sName; }

    public String getsBd() { return sBd; }

    @Override
    public String toString() {
        return "Student{" +
            "SID='" + sID + '\'' +
            ", Sname='" + sName + '\'' +
            ", Sbd='" + sBd + '\'' +
            '}';
    }
}
```

Course class:

```
public class Course {
    private String cID;
    private String cName;
    private String cCredits;

    public Course(String cID, String cName, String cCredits) {
        this.cID = cID;
        this.cName = cName;
        this.cCredits = cCredits;
    }

    public String getcID() { return cID; }

    public String getcName() { return cName; }

    public String getcCredits() { return cCredits; }

    @Override
    public String toString() {
        return "Course{" +
            "cID='" + cID + '\'' +
            ", cName='" + cName + '\'' +
            ", cCredits='" + cCredits + '\'' +
            '}';
    }
}
```

StudentEnrolment class:

```
public class StudentEnrolment {
    private Student student;
    private Course course;
    private String semester;

    public StudentEnrolment(Student student, Course course, String semester) {
        this.student = student;
        this.course = course;
        this.semester = semester;
    }

    public Student getStudent() { return student; }

    public Course getCourse() { return course; }

    public String getSemester() { return semester; }

    @Override
    public String toString() {
        return "StudentEnrolment{" +
            "student=" + student +
            ", course=" + course +
            ", semester='" + semester + '\'' +
            '}';
    }
}
```

For more information on the main and interface file, all information has been uploaded to GitHub.

2. Classes and objects:

I created an Array List type for every class in the main file to store their own data:

Student class will have `ArrayList<Student> studentList` that contains all the student data.

Course class will have `ArrayList<Course> courseList` that contains all the course data.

StudentEnrolment will have `ArrayList<StudentEnrolment>`

studentEnrolmentsListthat contains all the enrolment data.

3. Functions:

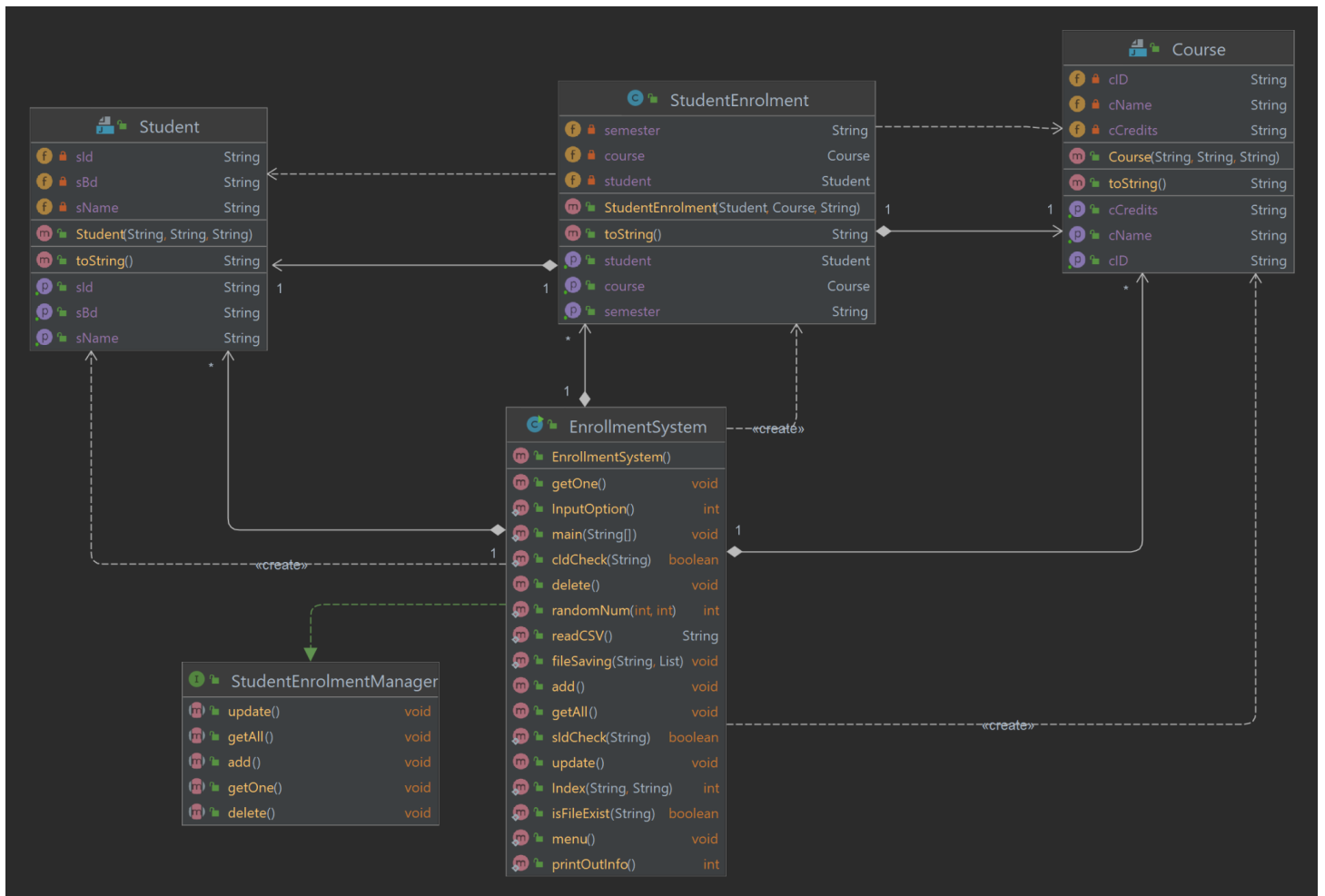
Besides 10 additional functions to support such as fileSaving, readCVS, and so on, basically, my application has 5 main functions including **add()**, **delete()**, **update()**, **getOne()** and **getAll()** which users can see on the interface.

- **add()**: This function will ask users to enter their sID, after some steps of validation like comparing the input with the data in the existing list, this function will add an enrolment record into the list.
- **delete()**: This function is similar to the add function, after validating the input, it will delete the enrolment record in the list.
- **update()**: This function will ask the user to input their sID, after validate the input, the system will print out all of the enrolment information of that student then allow them to use the add() or delete() function.
- **getOne()**: This function will allow the user to select among 3 options.

```
[1] Get a course from a semester.  
[2] Get a student from a semester.  
[3] Get a student from a course.\s""");
```

Then, the system will ask users to input their basic information such as sID, cID and semester in order to compare those information to the data in the existing list. Finally, if all the input information is valid, the system will print out the involving data that match the requirement.

4. Diagram:



5. Github:

Public github link of the application: [anhduc2708/Asm1 \(github.com\)](https://github.com/anhduc2708/Asm1)