

# Quản lý điểm

Một trung tâm tin học muốn quản lý các giáo viên, học viên của mình. Thông tin của mỗi giáo viên bao gồm: tên giáo viên, ngày tháng năm sinh, học vị (cử nhân, thạc sĩ, tiến sĩ). Thông tin của mỗi học viên bao gồm: tên học viên, ngày tháng năm sinh, giáo viên chủ nhiệm. Mỗi học viên sẽ đăng ký học một số môn học và phải thi cuối khóa. Thông tin của một môn học bao gồm: tên môn học, số tín chỉ, giáo viên giảng dạy. Sau khi kết thúc khóa học nhà trường sẽ đánh giá loại tốt nghiệp của học viên dựa trên điểm trung bình các điểm thi các môn học của học viên.

## **Yêu cầu 1:** Cho Use case: **In bảng điểm cho sinh viên**

1. Hệ thống hiển thị thông báo:  
Nhập tên học viên:
2. Người dùng nhập mã học viên muốn in bảng điểm
3. Hệ thống **tính điểm trung bình** cho học viên dựa vào kết quả điểm số các môn học của học viên như sau:

$$\text{Điểm TB} = \frac{\sum (\text{Điểm môn } i * \text{Số tín chỉ môn } i)}{\text{Tổng số tín chỉ các môn học}}$$

4. Hệ thống xếp loại tốt nghiệp cho học viên dựa vào điểm trung bình như sau:
  - Điểm TB  $\geq 9$  xếp loại xuất sắc.
  - $9 > \text{Điểm TB} \geq 8$  xếp loại là giỏi.
  - $8 > \text{Điểm TB} \geq 7$  xếp loại là khá.
  - $7 > \text{Điểm TB} \geq 6$  xếp loại là trung bình khá.
  - $6 > \text{Điểm TB} \geq 5$  xếp loại là trung bình.
  - Điểm TB  $< 5$  không được tốt nghiệp.

5. Hệ thống in bảng điểm cho học viên theo mẫu sau:

Name:

Ngày sinh:

STT

Môn

Điểm

...

Điểm TB:

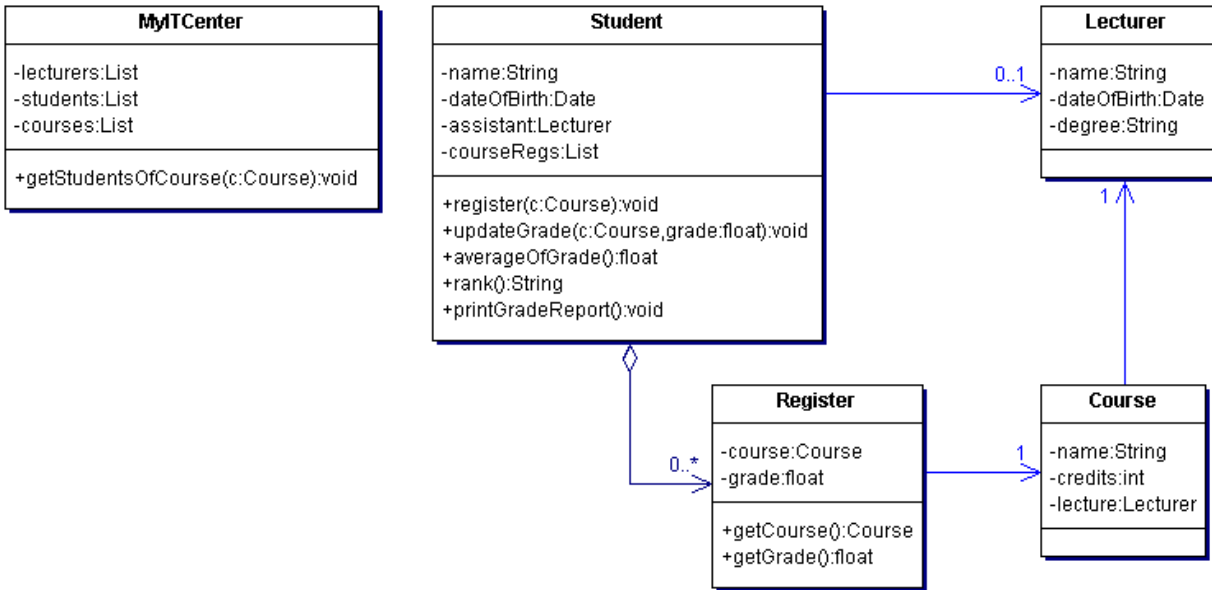
Xếp loại:

## **Hãy hiện thực use case trên:**

1. Nhận diện, thiết kế và cài đặt các lớp cho use case
2. Thiết kế các ứng xử trong use case và gán cho các lớp
  - a. Tính điểm trung bình cho học viên.
  - b. Xếp loại tốt nghiệp cho học viên
  - c. In bảng điểm cho học viên.

## **Yêu cầu 2:** Hãy thiết kế và hiện thực use case

**In danh sách các sinh viên và điểm thi của một môn học nào đó.**



### Course class

```

package gm;
public class Course {
    private String name;
    private int credits;
    private Lecturer lecturer;

    public Course(String name, int credits, Lecturer lecturer) {
        this.name = name;
        this.credits = credits;
        this.lecturer = lecturer;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getCredits() {
        return credits;
    }

    public void setCredits(int credits) {
        this.credits = credits;
    }

    public Lecturer getLecturer() {
        return lecturer;
    }

    public void setLecturer(Lecturer lecturer) {
        this.lecturer = lecturer;
    }
}
  
```

```

    public int hashCode() {
        return name.hashCode();
    }

    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Course other = (Course) obj;
        if (name == null) {
            if (other.name != null)
                return false;
        } else if (!name.equals(other.name))
            return false;
        return true;
    }

    public String toString() {
        return getName() + " " + getCredits() + " " + getLecturer();
    }
}

```

## Lecture class

```

package gm;
import java.util.Date;

public class Lecturer {
    private String name;
    private Date dateOfBirth;
    private String degree;

    public Lecturer(String name, Date dateOfBirth, String degree) {
        this.name = name;
        this.dateOfBirth = dateOfBirth;
        this.degree = degree;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Date getDateOfBirth() {
        return dateOfBirth;
    }

    public void setDateOfBirth(Date dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }
}

```

```

    public String getDegree() {
        return degree;
    }

    public void setDegree(String degree) {
        this.degree = degree;
    }

    public String toString() {
        return getDegree() + " " + getName();
    }
}

```

## Student class

```

package gm;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class Student {
    private String name;
    private Date dateOfBirth;
    private Lecturer assistant;
    private List<Register> courseRegisters = new ArrayList<Register>();

    public Student(String name, Date dateOfBirth) {
        this.name = name;
        this.dateOfBirth = dateOfBirth;
    }

    public String getName() {
        return name;
    }

    public Date getDateOfBirth() {
        return dateOfBirth;
    }

    public Lecturer getAssistant() {
        return assistant;
    }

    public List<Register> getCourseRegisters() {
        return courseRegisters;
    }

    public void register(Course c) {
        courseRegisters.add(new Register(c));
    }

    public void updateGrade(Course c, float grade) {
        for (Register r : courseRegisters) {
            if (r.getCourse().getName().equals(c.getName())) {
                r.setGrade(grade);
                break;
            }
        }
    }
}

```

```

    }
}

public float averageOfGrade() {
    float sum = 0;
    for (Register r : courseRegisters) {
        sum += r.getGrade();
    }
    return sum / courseRegisters.size();
}

public String rank() {
    float average = averageOfGrade();
    if (average >= 9)
        return "XUAT SAC";
    else if (average >= 8)
        return "GIOI";
    else if (average >= 7)
        return "KHA";
    else if (average >= 6)
        return "TRUNG BINH KHA";
    else if (average >= 7)
        return "TRUNG BINH";
    else
        return "ROT";
}

public void printGradeReport() {
    System.out.println("Name: " + getName());
    System.out.println("Date of birth: " + getDateOfBirth());
    System.out.println("STT\tMon\tDiem");
    for (int i = 0; i < courseRegisters.size(); i++) {
        System.out.println(i + "\t"
            + courseRegisters.get(i).getCourse().getName() + "\t"
            + courseRegisters.get(i).getGrade());
    }
    System.out.println("Average of grade: " + averageOfGrade());
    System.out.println("Rank: " + rank());
    System.out.println("-----");
}

public String toString() {
    return getName() + " " + getDateOfBirth();
}
}

```

## Register class

```

package gm;
public class Register {
    private Course course;
    private float grade;
}

```

```

    public Register(Course course) {
        this.course = course;
    }

    public Course getCourse() {
        return course;
    }

    public void setGrade(float grade) {
        this.grade = grade;
    }

    public float getGrade() {
        return grade;
    }
}

```

### MyITCenter class

```

package gm;

import java.util.ArrayList;
import java.util.GregorianCalendar;
import java.util.List;
import java.util.Scanner;

public class MyITCenter {
    private List<Lecturer> lecturers = new ArrayList<Lecturer>();
    private List<Student> students = new ArrayList<Student>();
    private List<Course> courses = new ArrayList<Course>();

    public MyITCenter() {
        init();
    }

    public void init() {
        Lecturer l1 = new Lecturer("Pham Van Tinh", new GregorianCalendar(1970,
            11, 20).getTime(), "PhD");
        Lecturer l2 = new Lecturer("Mai Anh Tho", new GregorianCalendar(1979, 12,
            24).getTime(), "MS");
        Lecturer l3 = new Lecturer("Nguyen Duc Cong Song", new GregorianCalendar(
            1977, 8, 14).getTime(), "MS");
        lecturers.add(l1);
        lecturers.add(l2);
        lecturers.add(l3);

        Course c1 = new Course("Lap trinh co ban", 4, 12);
        Course c2 = new Course("Lap trinh nang cao", 4, 13);
        Course c3 = new Course("Lap trinh mang", 4, 11);
        Course c4 = new Course("Mang may tinh", 4, 11);
        courses.add(c1);
        courses.add(c2);
        courses.add(c3);
        courses.add(c4);

        Student s1 = new Student("Nguyen Van Tuan", new GregorianCalendar(1988,

```

```
1, 23).getTime());
Student s2 = new Student("Tran Van Thiet", new GregorianCalendar(1989,
11, 2).getTime());
Student s3 = new Student("Nguyen Thi Lan", new GregorianCalendar(1988, 6,
20).getTime());
Student s4 = new Student("Nguyen Thi Truc", new GregorianCalendar(1990,
4, 15).getTime());
Student s5 = new Student("Vo Tan Doi",
    new GregorianCalendar(1988, 2, 26).getTime());
Student s6 = new Student("Do The Sang",
    new GregorianCalendar(1989, 3, 12).getTime());
students.add(s1);
students.add(s2);
students.add(s3);
students.add(s4);
students.add(s5);
students.add(s6);

s1.register(c1);
s1.register(c2);
s1.register(c3);

s2.register(c2);
s2.register(c3);
s2.register(c4);

s3.register(c1);
s3.register(c2);
s3.register(c3);
s3.register(c4);

s4.register(c1);
s4.register(c3);
s4.register(c4);

s5.register(c1);
s5.register(c2);
s5.register(c3);
s5.register(c4);

s6.register(c1);
s6.register(c2);

s1.updateGrade(c1, 6);
s1.updateGrade(c2, 7);
s1.updateGrade(c3, 8);

s2.updateGrade(c2, 8.5f);
s2.updateGrade(c3, 5);
s2.updateGrade(c4, 7);

s3.updateGrade(c1, 4);
s3.updateGrade(c2, 7.5f);
s3.updateGrade(c3, 8);
s3.updateGrade(c4, 10);
```

```

        s4.updateGrade(c1, 8);
        s4.updateGrade(c3, 3);
        s4.updateGrade(c4, 6);

        s5.updateGrade(c1, 5);
        s5.updateGrade(c2, 4);
        s5.updateGrade(c3, 6);
        s5.updateGrade(c4, 8);

        s6.updateGrade(c1, 5);
        s6.updateGrade(c2, 8);
    }

    public static void printList(List<?> list) {
        for (Object o : list) {
            System.out.println(o);
        }
        System.out.println("-----");
    }

    public Student findStudentByName(String name) {
        for (Student s : students) {
            if (s.getName().equals(name)) return s;
        }
        return null;
    }

    public Course findCourseByName(String name) {
        for (Course c : courses) {
            if (c.getName().equals(name)) return c;
        }
        return null;
    }

    public List<Student> getStudentsOfCourse(Course c) {
        List<Student> result = new ArrayList<Student>();
        for (Student student : students) {
            for (Register register : student.getCourseRegisters()) {
                if (register.getCourse().getName().equals(c.getName()))
                    result.add(student);
            }
        }
        return result;
    }

    // print report card use case
    public void printStudentReportCardUseCase() {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter student name:");
        String studentName = in.nextLine();
        Student student = findStudentByName(studentName.trim());
        if (student != null) {
            student.printGradeReport();
        }
        else {
            System.out.println("Student name not found!");
        }
    }

```



```
    }  
}  
  
// print student list of course use case  
public void printStudentOfCourseUseCase() {  
    Scanner in = new Scanner(System.in);  
    System.out.println("Enter course name:");  
    String courseName = in.nextLine();  
    Course course = findCourseByName(courseName.trim());  
    if (course != null) {  
        printList(getStudentsOfCourse(course));  
    }  
    else {  
        System.out.println("Course name not found!");  
    }  
}  
  
public static void main(String[] args) {  
    MyITCenter myCenter = new MyITCenter();  
    myCenter.printStudentOfCourseUseCase();  
    myCenter.printStudentOfCourseUseCase();  
}  
}
```