JAVA PROJECTS5

5.Student Grade Management System

Description: Develop a student grade management system to store and manage student records, course grades, GPA calculation, and generate academic reports**.**

1.Student class

```
import java.util.ArrayList;
import java.util.List;

public class Student {
    private int id;
    private String name;
    private List<Grade> grades;

    public Student(int id, String name) {
        this.id = id;
        this.name = name;
        this.grades = new ArrayList<>();
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public void addGrade(Grade grade) {
        grades.add(grade);
```

```
    }

    public double calculateGPA() {
        double totalPoints = 0;
        int totalCredits = 0;

        for (Grade grade : grades) {
            totalPoints += grade.getGradePoints();
            totalCredits += grade.getCourse().getCredits();
        }

        return totalCredits == 0 ? 0 : totalPoints / totalCredits;
    }

    public List<Grade> getGrades() {
        return grades;
    }

    @Override
    public String toString() {
        return "Student ID: " + id + ", Name: " + name;
    }
}
```

2.Course class

```
public class Course {
    private String code;
    private String title;
    private int credits;
```

```
    public Course(String code, String title, int credits) {
        this.code = code;
        this.title = title;
        this.credits = credits;
    }

    public String getCode() {
        return code;
    }

    public String getTitle() {
        return title;
    }

    public int getCredits() {
        return credits;
    }

    @Override
    public String toString() {
        return "Course Code: " + code + ", Title: " + title + ", Credits: " + credits;
    }
}
```

3.Grade Class

```
public class Grade {
    private Course course;
    private String grade;

    public Grade(Course course, String grade) {
```

```
    this.course = course;
    this.grade = grade;
}

public Course getCourse() {
    return course;
}

public String getGrade() {
    return grade;
}

public double getGradePoints() {
    switch (grade) {
        case "A":
            return 4.0 * course.getCredits();
        case "B":
            return 3.0 * course.getCredits();
        case "C":
            return 2.0 * course.getCredits();
        case "D":
            return 1.0 * course.getCredits();
        case "F":
            return 0.0;
        default:
            return 0.0;
    }
}
```

```
    @Override
    public String toString() {
        return "Course: " + course + ", Grade: " + grade;
    }
}
```

4.School class

```
import java.util.ArrayList;
import java.util.List;

public class School {
    private List<Student> students;
    private List<Course> courses;

    public School() {
        students = new ArrayList<>();
        courses = new ArrayList<>();
    }

    public void addStudent(Student student) {
        students.add(student);
    }

    public void addCourse(Course course) {
        courses.add(course);
    }

    public Student findStudentById(int id) {
        for (Student student : students) {
            if (student.getId() == id) {
```

```
            return student;
        }
    }
    return null;
}

public Course findCourseByCode(String code) {
    for (Course course : courses) {
        if (course.getCode().equals(code)) {
            return course;
        }
    }
    return null;
}

public void listStudents() {
    for (Student student : students) {
        System.out.println(student);
    }
}

public void listCourses() {
    for (Course course : courses) {
        System.out.println(course);
    }
}
}
```

5.Main class

```
public class Main {
```

```
public static void main(String[] args) {
    // Create a school
    School school = new School();

    // Add courses to the school
    Course course1 = new Course("CS101", "Introduction to Computer Science", 3);
    Course course2 = new Course("MATH101", "Calculus I", 4);
    school.addCourse(course1);
    school.addCourse(course2);

    // Add students to the school
    Student student1 = new Student(1, "Alice");
    Student student2 = new Student(2, "Bob");
    school.addStudent(student1);
    school.addStudent(student2);

    // Add grades for students
    student1.addGrade(new Grade(course1, "A"));
    student1.addGrade(new Grade(course2, "B"));
    student2.addGrade(new Grade(course1, "C"));
    student2.addGrade(new Grade(course2, "A"));

    // List students
    System.out.println("Students in the school:");
    school.listStudents();

    // List courses
    System.out.println("\nCourses offered by the school:");
    school.listCourses();
```

```
        // Generate academic reports
        System.out.println("\nAcademic report for Alice:");
        System.out.println("GPA: " + student1.calculateGPA());
        for (Grade grade : student1.getGrades()) {
            System.out.println(grade);
        }

        System.out.println("\nAcademic report for Bob:");
        System.out.println("GPA: " + student2.calculateGPA());
        for (Grade grade : student2.getGrades()) {
            System.out.println(grade);
        }
    }
}
```

OUTPUT:

```
Students in the school:
Student ID: 1, Name: Alice
Student ID: 2, Name: Bob

Courses offered by the school:
Course Code: CS101, Title: Introduction to Computer Science, Credits: 3
Course Code: MATH101, Title: Calculus I, Credits: 4

Academic report for Alice:
GPA: 3.2857142857142856
Course: Course Code: CS101, Title: Introduction to Computer Science, Credits: 3, Grade: A
Course: Course Code: MATH101, Title: Calculus I, Credits: 4, Grade: B
```

Academic report for Bob:

GPA: 2.857142857142857

Course: Course Code: CS101, Title: Introduction to Computer Science, Credits: 3, Grade: C

Course: Course Code: MATH101, Title: Calculus I, Credits: 4, Grade: A