

Technical Documentation: Course Enrollment and Grade Management System

Overview

This Java-based system manages the enrollment of students in courses, tracks their grades, and provides detailed reports on student performance. It consists of four main classes, each with specific responsibilities and functionalities.

Class Descriptions

1. Student Class

Purpose: Represents individual students, storing their personal details and enrolled courses.

Attributes:

- String name: The student's name.
- String ID: A unique identifier for the student.
- HashMap<Course, String> enrolledCourses: Maps Course objects to grades.

Key Methods:

- enrollInCourse(Course course): Enrolls the student in a specified course.
- assignGrade(Course course, String grade): Assigns a grade to the student for a specified course.
- getEnrolledCourses(): Returns the map of enrolled courses and grades.
- Getters and setters for name and ID.

2. Course Class

Purpose: Represents academic courses with specific attributes.

Attributes:

- String courseCode: A unique code identifying the course.
- String name: The name of the course.
- int maxCapacity: The maximum number of students that can enroll.

Key Methods:

- Getters for courseCode, name, and maxCapacity.

3. CourseManagement Class

Purpose: Manages the main functionalities such as enrollments, grade assignments, and student data retrieval.

Attributes:

- ArrayList<Student> students: Stores all Student objects.
- ArrayList<Course> courses: Stores all Course objects.
- HashMap<Student, HashMap<Course, String>> courseGrades: Maps each Student to their courses and grades.

Key Methods:

- `findStudentById(String id)`: Retrieves a Student object by its ID.
- `findCourseByCode(String code)`: Retrieves a Course object by its code.
- `enrollStudent(Student student, Course course)`: Enrolls a student in a specified course.
- `assignGrade(Student student, Course course, String grade)`: Assigns a grade to a student for a specified course.
- `calculateOverallGrade(Student student)`: Calculates the overall grade for a student.
- `getStudentDetails(String studentId)`: Retrieves detailed information about a student, including enrolled courses and grades.
- `listAllCourses()`: Returns a list of all available courses.

4. CourseManagementCLI Class

Purpose: Provides a command-line interface for administrators to interact with the system.

Key Methods:

- `main(String[] args)`: The entry point of the program. Displays a menu and processes user input.
- `displayMenu()`: Displays the available menu options.
- `addCourse()`: Interface to add a new course.
- `enrollStudent()`: Interface to enroll a student in a course.
- `assignGrade()`: Interface to assign a grade to a student in a course.
- `calculateOverallGrade()`: Interface to calculate and display a student's overall grade.
- `viewStudentDetails()`: Interface to display detailed information about a student.
- `listAllCourses()`: Interface to list all available courses.

System Interaction and Workflow

- **Adding Courses and Students**: The system allows adding new courses and enrolling students in these courses.
- **Assigning Grades**: Administrators can assign grades to students for their courses.
- **Viewing Student Details**: The system can display detailed information about a student, including their enrolled courses, grades for each course, and overall grade.
- **Listing Courses**: All available courses can be listed through the CLI.
- **Calculating Overall Grades**: The system calculates the overall grade of students based on their course grades.

Notes

- Data Persistence: Currently, the system does not persist data between runs. Implementing data storage (e.g., using a database) would be necessary for long-term data retention.
- The system initializes with sample students and courses for demonstration and testing purposes.