**Exercise 10: Implementing the MVC Pattern**

public class MVCPatternExample {

// Step 2: Model Class

static class Student {

private String name;

private String id;

private String grade;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getGrade() {

return grade;

}

public void setGrade(String grade) {

this.grade = grade;

}

}

// Step 3: View Class

static class StudentView {

public void displayStudentDetails(String name, String id, String grade) {

System.out.println("Student Details:");

System.out.println("Name: " + name);

System.out.println("ID: " + id);

System.out.println("Grade: " + grade);

}

}

// Step 4: Controller Class

static class StudentController {

private Student model;

private StudentView view;

public StudentController(Student model, StudentView view) {

this.model = model;

this.view = view;

}

public void setStudentName(String name) {

model.setName(name);

}

public String getStudentName() {

return model.getName();

}

public void setStudentId(String id) {

model.setId(id);

}

public String getStudentId() {

return model.getId();

}

public void setStudentGrade(String grade) {

model.setGrade(grade);

}

public String getStudentGrade() {

return model.getGrade();

}

public void updateView() {

view.displayStudentDetails(model.getName(), model.getId(), model.getGrade());

}

}

// Step 5: Test in main()

public static void main(String[] args) {

Student model = new Student();

model.setName("Kalai");

model.setId("S101");

model.setGrade("A");

StudentView view = new StudentView();

StudentController controller = new StudentController(model, view);

controller.updateView();

System.out.println("\nUpdating student grade to A+...\n");

controller.setStudentGrade("A+");

controller.updateView();

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.