**Exercise 10: Implementing the MVC Pattern**

**SCENARIO:**

You are developing a simple web application for managing student records using the MVC pattern.

**STEPS:**

1. **Create a New Java Project:**
   * Create a new Java project named **MVCPatternExample**.
2. **Define Model Class:**
   * Create a class **Student** with attributes like **name, id, and grade**.
3. **Define View Class:**
   * Create a class **StudentView** with a method **displayStudentDetails()**.
4. **Define Controller Class:**
   * Create a class **StudentController** that handles the communication between the model and the view.
5. **Test the MVC Implementation:**
   * Create a main class to demonstrate creating a **Student**, updating its details using **StudentController**, and displaying them using **StudentView**.

**SOLUTION:**

**Step 1:** Create a New Java Project

**Step 2:** Define Model Class:

class Student{

String name;

String id;

String grade;

public Student(String name,String id,String grade)

{

this.name=name;

this.id=id;

this.grade=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**: Define View Class:

class StudentView{

//private StudentController stdctrl;

//private Student student;

public void displayStudentDetails(Student student)

{

System.out.println("NAME: "+student.getName());

System.out.println("ID: "+student.getId());

System.out.println("GRADE: "+student.getGrade());

System.out.println();

}

}

**Step 4:** Define Controller Class:

class StudentController

{

private Student student;

private StudentView view;

private List<Student> students=new ArrayList<>();

public StudentController(StudentView view)

{

this.view=view;

}

public void addStudent(Student student)

{

students.add(student);

}

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

for (Student student : students) {

if (student.getId().equals(id)) {

student.setName(name);

student.setGrade(grade);

}

}

}

public void displayDetails()

{

for(Student student:students)

{

view.displayStudentDetails(student);

}

}

}

**Step 5:** Test the MVC Implementation:

public class MVCPattern {

public static void main(String[] args) {

Student student1 = new Student("ALENA", "101", "A");

Student student2 = new Student("DAPHNE", "102", "B");

Student student3 = new Student("GREGORY", "103", "C");

StudentController stdc = new StudentController(new StudentView());

stdc.addStudent(student1);

stdc.addStudent(student2);

stdc.addStudent(student3);

stdc.displayDetails();

stdc.updateStudent("101","DRAKE","A");

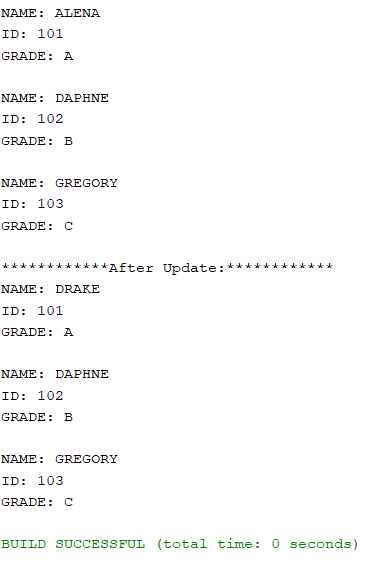
System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*After Update:\*\*\*\*\*\*\*\*\*\*\*\*");

stdc.displayDetails();

}

}

**SAMPLE OUTPUT:**

****