**Implement the following projects which focus on different aspects of Java programming, including:**

Object-Oriented Programming principles

Exception handling

File operations

Data structures

Input validation

User interface design

**3. Student Grade Calculator**

Create a program that calculates average grades and determines pass/fail status for multiple students using object oriented programming in Java.

Key points:

- Use arrays to store grades

- Implement input validation (0-100)

import java.util.\*;

import java.io.\*;

// Custom exceptions

class InvalidGradeException extends Exception {

public InvalidGradeException(String message) {

super(message);

}

}

// Student class to store student information and grades

class Student implements Serializable {

private final String studentId;

private final String name;

private final List<Subject> subjects;

private double averageGrade;

private String status;

public Student(String studentId, String name) {

this.studentId = studentId;

this.name = name;

this.subjects = new ArrayList<>();

}

public void addSubject(Subject subject) {

subjects.add(subject);

calculateAverage();

determineStatus();

}

private void calculateAverage() {

if (subjects.isEmpty()) {

averageGrade = 0.0;

return;

}

double sum = subjects.stream()

.mapToDouble(Subject::getGrade)

.sum();

averageGrade = sum / subjects.size();

}

private void determineStatus() {

status = (averageGrade >= 60) ? "PASS" : "FAIL";

}

// Getters

public String getStudentId() { return studentId; }

public String getName() { return name; }

public List<Subject> getSubjects() { return new ArrayList<>(subjects); }

public double getAverageGrade() { return averageGrade; }

public String getStatus() { return status; }

@Override

public String toString() {

return String.format("Student ID: %s\nName: %s\nAverage Grade: %.2f\nStatus: %s",

studentId, name, averageGrade, status);

}

}

// Subject class to store subject information

class Subject implements Serializable {

private final String name;

private final double grade;

public Subject(String name, double grade) throws InvalidGradeException {

validateGrade(grade);

this.name = name;

this.grade = grade;

}

private void validateGrade(double grade) throws InvalidGradeException {

if (grade < 0 || grade > 100) {

throw new InvalidGradeException("Grade must be between 0 and 100");

}

}

public String getName() { return name; }

public double getGrade() { return grade; }

@Override

public String toString() {

return String.format("%s: %.2f", name, grade);

}

}

// Grade Calculator class to manage all operations

class GradeCalculator {

private final Map<String, Student> students;

private final String dataFile = "students\_data.ser";

public GradeCalculator() {

this.students = new HashMap<>();

loadData();

}

public void addStudent(String studentId, String name) {

students.put(studentId, new Student(studentId, name));

saveData();

}

public void addGrade(String studentId, String subjectName, double grade)

throws InvalidGradeException {

Student student = students.get(studentId);

if (student != null) {

student.addSubject(new Subject(subjectName, grade));

saveData();

}

}

public Student getStudent(String studentId) {

return students.get(studentId);

}

public List<Student> getAllStudents() {

return new ArrayList<>(students.values());

}

// Save data to file

private void saveData() {

try (ObjectOutputStream oos = new ObjectOutputStream(

new FileOutputStream(dataFile))) {

oos.writeObject(students);

} catch (IOException e) {

System.err.println("Error saving data: " + e.getMessage());

}

}

// Load data from file

@SuppressWarnings("unchecked")

private void loadData() {

File file = new File(dataFile);

if (!file.exists()) return;

try (ObjectInputStream ois = new ObjectInputStream(

new FileInputStream(file))) {

Map<String, Student> loadedData = (Map<String, Student>) ois.readObject();

students.putAll(loadedData);

} catch (IOException | ClassNotFoundException e) {

System.err.println("Error loading data: " + e.getMessage());

}

}

}

// Main class with user interface

public class StudentGradeManagementSystem {

private static final Scanner scanner = new Scanner(System.in);

private static final GradeCalculator calculator = new GradeCalculator();

public static void main(String[] args) {

while (true) {

try {

displayMenu();

String choice = scanner.nextLine();

switch (choice) {

case "1":

addNewStudent();

break;

case "2":

addGrades();

break;

case "3":

viewStudentReport();

break;

case "4":

viewAllStudents();

break;

case "5":

System.out.println("Thank you for using the Grade Calculator!");

return;

default:

System.out.println("Invalid choice. Please try again.");

}

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

}

private static void displayMenu() {

System.out.println("\n=== Student Grade Calculator ===");

System.out.println("1. Add New Student");

System.out.println("2. Add Grades");

System.out.println("3. View Student Report");

System.out.println("4. View All Students");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

}

private static void addNewStudent() {

System.out.print("Enter Student ID: ");

String studentId = scanner.nextLine();

System.out.print("Enter Student Name: ");

String name = scanner.nextLine();

calculator.addStudent(studentId, name);

System.out.println("Student added successfully!");

}

private static void addGrades() throws InvalidGradeException {

System.out.print("Enter Student ID: ");

String studentId = scanner.nextLine();

Student student = calculator.getStudent(studentId);

if (student == null) {

System.out.println("Student not found!");

return;

}

System.out.print("Enter Subject Name: ");

String subjectName = scanner.nextLine();

System.out.print("Enter Grade (0-100): ");

try {

double grade = Double.parseDouble(scanner.nextLine());

calculator.addGrade(studentId, subjectName, grade);

System.out.println("Grade added successfully!");

} catch (NumberFormatException e) {

throw new InvalidGradeException("Invalid grade format");

}

}

private static void viewStudentReport() {

System.out.print("Enter Student ID: ");

String studentId = scanner.nextLine();

Student student = calculator.getStudent(studentId);

if (student == null) {

System.out.println("Student not found!");

return;

}

System.out.println("\n=== Student Report ===");

System.out.println(student);

System.out.println("\nSubject Grades:");

for (Subject subject : student.getSubjects()) {

System.out.println(subject);

}

}

private static void viewAllStudents() {

List<Student> students = calculator.getAllStudents();

if (students.isEmpty()) {

System.out.println("No students found!");

return;

}

System.out.println("\n=== All Students ===");

for (Student student : students) {

System.out.println("\n" + student);

}

}

}

**o/p**

**=== Student Grade Calculator ===**

**1. Add New Student**

**2. Add Grades**

**3. View Student Report**

**4. View All Students**

**5. Exit**

**Enter your choice: 1**

**Enter Student ID: 101**

**Enter Student Name: Ram Kuamr**

**Student added successfully!**

**=== Student Grade Calculator ===**

**1. Add New Student**

**2. Add Grades**

**3. View Student Report**

**4. View All Students**

**5. Exit**

**Enter your choice: 2**

**Enter Student ID: 101**

**Enter Subject Name: Ram Kumar**

**Enter Grade (0-100): 79**

**Grade added successfully!**

**=== Student Grade Calculator ===**

**1. Add New Student**

**2. Add Grades**

**3. View Student Report**

**4. View All Students**

**5. Exit**

**Enter your choice: 3**

**Enter Student ID: 101**

**=== Student Report ===**

**Student ID: 101**

**Name: Ram Kumar**

**Average Grade: 79.00**

**Status: PASS**

**Subject Grades:**

**Ram Kumar: 79.00**

**=== Student Grade Calculator ===**

**1. Add New Student**

**2. Add Grades**

**3. View Student Report**

**4. View All Students**

**5. Exit**

**Enter your choice: 1**

**Enter Student ID: 102**

**Enter Student Name: Ravi**

**Student added successfully!**

**=== Student Grade Calculator ===**

**1. Add New Student**

**2. Add Grades**

**3. View Student Report**

**4. View All Students**

**5. Exit**

**Enter your choice: 2**

**Enter Student ID: 102**

**Enter Subject Name: Ravi**

**Enter Grade (0-100): 89**

**Grade added successfully!**

**=== Student Grade Calculator ===**

**1. Add New Student**

**2. Add Grades**

**3. View Student Report**

**4. View All Students**

**5. Exit**

**Enter your choice: 4**

**=== All Students ===**

**Student ID: 101**

**Name: Ram Kumar**

**Average Grade: 79.00**

**Status: PASS**

**Student ID: 102**

**Name: Ravi**

**Average Grade: 89.00**

**Status: PASS**