java

```
import java.io.*;
import java.util.*;
class Student implements Serializable {
private static final long serialVersionUID = 1L;
String name;
Map<String, Integer> scores = new LinkedHashMap<>();
public Student(String name) {
this.name = name;
}
public double getAverage() {
if (scores.isEmpty()) return 0.0;
int sum = 0;
for (int v : scores.values()) sum += v;
return sum / (double) scores.size();
}
@Override
public String toString() {
return String.format("%s (avg=%.2f)", name, getAverage());
}
}
public class StudentGradeApp {
private static final String DATA_FILE = "students.ser";
private List<Student> students = new ArrayList<>();
private Scanner scanner = new Scanner(System.in);
public static void main(String[] args) {
StudentGradeApp app = new StudentGradeApp();
app.load();
app.run();
}
private void run() {
while (true) {
printMenu();
String choice = scanner.nextLine().trim();
switch (choice) {
case "1": list(); break;
case "2": addStudent(); break;
case "3": addScore(); break;
case "4": ranking(); break;
case "0": save(); System.out.println("Bye."); return;
default: System.out.println("Unknown menu.");
}
}
}
private void printMenu() {
System.out.println("\n==== Student Grade System ====");
System.out.println("1. List students");
System.out.println("2. Add student");
System.out.println("3. Add/Update score");
System.out.println("4. Show ranking");
System.out.println("0. Exit");
System.out.print("Select: ");
}
private void list() {
if (students.isEmpty()) {
System.out.println("No students.");
return;
}
for (Student s : students) {
System.out.println(s);
for (Map.Entry<String, Integer> e : s.scores.entrySet()) {
System.out.println(" " + e.getKey() + ": " + e.getValue());
}
}
}
private Student findStudent(String name) {
for (Student s : students) {
if (s.name.equalsIgnoreCase(name)) return s;
}
return null;
}
private void addStudent() {
System.out.print("Student name: ");
String name = scanner.nextLine().trim();
if (name.isEmpty()) {
System.out.println("Name required.");
return;
}
if (findStudent(name) != null) {
System.out.println("Already exists.");
return;
}
students.add(new Student(name));
System.out.println("Added.");
}
private void addScore() {
System.out.print("Student name: ");
String name = scanner.nextLine().trim();
Student s = findStudent(name);
if (s == null) {
System.out.println("Student not found.");
return;
}
System.out.print("Subject: ");
String subject = scanner.nextLine().trim();
System.out.print("Score (0-100): ");
String v = scanner.nextLine().trim();
try {
int score = Integer.parseInt(v);
s.scores.put(subject, score);
System.out.println("Updated.");
} catch (NumberFormatException e) {
System.out.println("Invalid score.");
}
}
private void ranking() {
if (students.isEmpty()) {
System.out.println("No students.");
return;
}
List<Student> copy = new ArrayList<>(students);
copy.sort(Comparator.comparingDouble(Student::getAverage).reversed());
System.out.println("\n[Ranking]");
int rank = 1;
for (Student s : copy) {
System.out.printf("%d. %s (avg=%.2f)\n", rank++, s.getAverage(), s.getAverage());
// Actually display with name
System.out.printf(" %s\n", s.name);
}
}
@SuppressWarnings("unchecked")
private void load() {
File f = new File(DATA_FILE);
if (!f.exists()) return;
try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(f))) {
students = (List<Student>) ois.readObject();
System.out.println("Loaded " + students.size() + " students.");
} catch (Exception e) {
System.out.println("Failed to load data: " + e.getMessage());
}
}
private void save() {
try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(DATA_FILE))) {
oos.writeObject(students);
System.out.println("Saved " + students.size() + " students.");
} catch (IOException e) {
System.out.println("Failed to save data: " + e.getMessage());
}
}
}
```