import java.util.\*;  
  
class Classroom {  
 final String name;  
 final Map<String, Student> students = new HashMap<>();  
 final Map<String, Assignment> assignments = new HashMap<>();  
 Classroom(String name){  
 [this.name](http://this.name/) = name;  
 }  
}  
  
class Student {  
 final String id;  
 final Set<String> submitted = new HashSet<>();  
 Student(String id){  
 [this.id](http://this.id/) = id;  
 }  
}  
  
class Assignment {  
 final String id;  
 final String details;  
 Assignment(String id, String details){ [this.id](http://this.id/)=id; this.details=details; }  
}  
  
public class VirtualClassroomManager {  
 private final Map<String, Classroom> classrooms = new HashMap<>();  
  
 public void addClassroom(String name) {  
 if (classrooms.containsKey(name)) {  
 System.out.println("Classroom exists.");  
 return;  
 }  
 classrooms.put(name, new Classroom(name));  
 System.out.println("Classroom " + name + " has been created.");  
 }  
 public void removeClassroom(String name) {  
 classrooms.remove(name);  
 System.out.println("Classroom " + name + " removed.");  
 }  
 public void addStudent(String studentId, String className) {  
 Classroom c = classrooms.get(className);  
 if (c==null) {  
 System.out.println("Classroom not found.");  
 return;  
 }  
 if (c.students.containsKey(studentId))  
 System.out.println("Student already enrolled.");  
 else {  
 c.students.put(studentId, new Student(studentId));  
 System.out.println("Student " + studentId + " enrolled in " + className);  
 }  
 }  
 public void scheduleAssignment(String className, String assignId, String details) {  
 Classroom c = classrooms.get(className);  
 if (c==null) {  
 System.out.println("Classroom not found.");  
 return;  
 }  
 c.assignments.put(assignId, new Assignment(assignId, details));  
 System.out.println("Assignment scheduled for " + className);  
 }  
 public void submitAssignment(String studentId, String className, String assignId) {  
 Classroom c = classrooms.get(className);  
 if (c==null) {  
 System.out.println("Classroom not found.");  
 return;  
 }  
 Student s = c.students.get(studentId);  
 if (s==null) {  
 System.out.println("Student not enrolled.");  
 return;  
 }  
 Assignment a = c.assignments.get(assignId);  
 if (a==null) {  
 System.out.println("Assignment not found.");  
 return;  
 }  
 s.submitted.add(assignId);  
 System.out.println("Assignment " + assignId + " submitted by Student " + studentId + " in " + className);  
 }  
 public void listClassrooms() {  
 System.out.println("Classrooms: " + classrooms.keySet());  
 }  
 public void listStudents(String className) {  
 Classroom c = classrooms.get(className);  
 if (c==null) { System.out.println("Classroom not found."); return; }  
 System.out.println("Students in " + className + ": " + c.students.keySet());  
 }  
 public void listAssignments(String className) {  
 Classroom c = classrooms.get(className);  
 if (c==null) { System.out.println("Classroom not found."); return; }  
 System.out.println("Assignments in " + className + ": " + c.assignments.keySet());  
 }  
  
 public static void main(String[] args) {  
 VirtualClassroomManager manager = new VirtualClassroomManager();  
 Scanner sc = new Scanner(System.in);  
 System.out.println("=== Virtual Classroom Manager ===");  
 while (true) {  
 System.out.println("\nCommand: add\_classroom <name>") ;  
 System.out.println("\nCommand: add\_student <id> <class>");  
 System.out.println("\nCommand: schedule\_assignment <class> <aid> <details>");  
 System.out.println("\nCommand: submit\_assignment <studentId> <class> <aid>");  
 System.out.println("\nCommand:list\_classes");  
 System.out.println("\nCommand:list\_students <class> ");  
 System.out.println("\nCommand:list\_assignments <class>");  
 System.out.println("\nCommand:exit");  
 String line = sc.nextLine().trim();  
 if (line.equalsIgnoreCase("exit")) break;  
 String[] p = line.split("\\s+", 4);  
 try {  
 switch (p[0]) {  
 case "add\_classroom": manager.addClassroom(p[1]); break;  
 case "add\_student": manager.addStudent(p[1], p[2]); break;  
 case "schedule\_assignment": manager.scheduleAssignment(p[1], p[2], p.length>3?p[3]:""); break;  
 case "submit\_assignment": manager.submitAssignment(p[1], p[2], p[3]); break;  
 case "list\_classes": manager.listClassrooms(); break;  
 case "list\_students": manager.listStudents(p[1]); break;  
 case "list\_assignments": manager.listAssignments(p[1]); break;  
 default: System.out.println("Unknown command");  
 }  
 } catch (Exception e) { System.out.println("Error: " + e.getMessage()); }  
 }  
 System.out.println("Exiting Virtual Classroom Manager.");  
 }  
}