Collections

- Scenario: You are a Database Administrator for a university and need to create an application to manage groups of students, teachers and courses.

- (This will be similar with Basic DAO example, but the focus will be on collections and not on decoupling layers using DAO)

- Details:

- The Application will create and store a set of <Groups>

- A <Group> can have one or multiple <Teachers>

- A <Group> can have multiple <Students>

- A <Teacher> can teach one or multiple <Courses>

Steps to follow:

1. Implement a class <Student> with the following private fields

- int id

- String name

- add constructors (default, with all arguments), getters and setters

- override equals and hashCode as <Student> will be part of a Set

2. Implement a class <Course> with the following private fields

- int id

- String name

- add constructors (default, with all arguments), getters and setters

- override equals and hashCode as <Course> will be part of a Set

3. Implement a class <Teacher> with the following private fields

- int id

- String name

- Set <Course> courses (will be a HashSet)

- add constructors (default, with all arguments), getters and setters

- override equals and hashCode as <Course> will be part of a Set

- add following methods:

- Set<Course> getCourses()

- Course getCourse(int id)

- setCourses (Set<Course>)

- addCourser(Course)

- updateCourse(Course, String courseName)

- removeCourse(Course)

4. Implement a class <Group> with the following private fields

- int id

- String name

- Set<Teacher> teachers (will be a HashSet)

- Set<Student> students (will be a HashSet)

- add constructors (default, with all arguments), getters and setters

- override equals and hashCode as <Group> will be part of a Set

- add following methods;

- Set<Teacher> getTeachers()

- Teacher getTeacher(int id)

- setTeachers (Set<Teacher>)

- addTeacher(Teacher)

- updateTeacher(Teacher, String teacherName)

- removeTeacher(Teacher)

- Set<Student> getStudents()

- Student getStudent(int id)

- setStudents (Set<Student>)

- addStudent(Student)

- updateStudent(Student, String studentName)

- removeStudent(Student)

5. Implement a class <Application> with the following private fields

- Set<Group> groups (will be a HashSet)

- add a main method where you should add 2 Groups to the set of groups

- each Group should have 4 Teachers

- each Group should have 3 Students

- each Teacher should teach 2 Courses

- display information for all the groups in a friendly format

Question: why is better to store Teachers for example in a Set and not in a List?

am adaugat si functii de get (int id) pentru Course, Teacher si Student:

- Course getCourse(int id)

- Teacher getTeacher(int id)

- Student getStudent(int id)