

Task#08 – Collection Classes

Task 1:

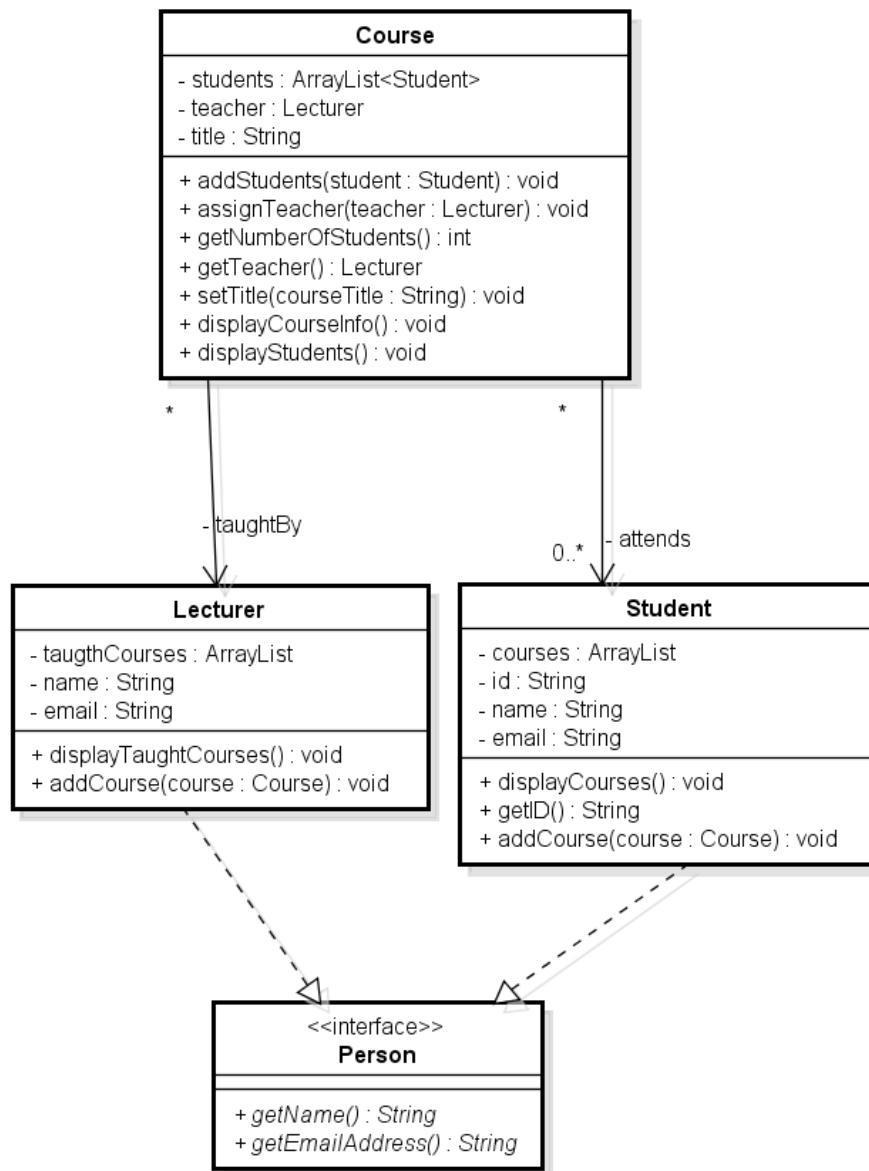
LinkedList
+addFirst(element : Object) : void +addLast(element : Object) : void +getFirst() : Object +getLast() : Object +removeFirst() : Object +removeLast() : Object

By using these new methods, you can easily treat the *LinkedList* as a stack, queue, or other end-oriented data structure. A queue is a *LinkedList* in which elements are added and retrieved in first-in-first-out order. Similarly a stack is a *LinkedList* in which elements are added and retrieved in first-in-last-out order. Implement two classes *Queue* and *Stack* which using the class *LinkedList* and test the implemented classes in a test program.

Task 2:

Students and Courses

This programming test deals with an imaginary application that models university courses. (40 Marks)



Course

- Each Course object maintains a list of the students on that course and the lecturer who has been assigned to teach that course.
- The Course object has behavior that allows assigning title to the course, adding students to the course, and assigning a teacher.
- *getNumberOfStudent()* returns the total number of students enrolled in the course.
- *displayCourseInfo()* displays the name of course, total number of students in the course, and the name of lecturer teaching the course.
- *displayStudents()* displays the name and ids of the students enrolled in the course.

Person

- The Person interface enforces implementation of the getter methods for name and email address.

Lecturer

- Teachers are modeled as Lecturer objects and implement the Person interface. As a lecturer may teach more than one course, there is an association between Course and Lecturer. The “taughtBy” relationship shows that a Course only has a single teacher, but that a lecturer may teach several Courses.
- Each Lecturer object maintains a list of the Courses that he teaches.
- *displayTaughtCourses()* calls the *displayCourseInfo()* method for each of the course taught by the teacher
- Two getter methods return the name and email address of the teacher.

Student

- There is a similar relationship between Course and Student. A course is attended by zero or more Students, and a Student may attend multiple courses.
- Student implements the Person interface.
- For each student, there is a list of courses he is enrolled in. A course can be added for a student by *addCourse()* method.
- *displayCourses()* method calls the *displayCourseInfo()* method for each courses the student is enrolled in.

TestApplication

Write a main class that creates two courses, two lecturers, and four students. Assign a teacher and some students to each course. Display information about each course i.e. its title, total number of students in the course, the name of lecturer teaching the course, and name and id of each student enrolled in the course.

Make sure that you satisfy both ends of a relation i.e. if you are adding a student to the list of enrolled students in a course, make sure that you add the course to the list of courses in the student object as well.

For each student, display the list of courses he is enrolled in.

Help Code:

```
List<String> arrayList = new ArrayList<String>();
Iterator<String> itr = arrayList.iterator();
    while (itr.hasNext()){
        System.out.println(itr.next());
    }
```