Java Final Project

(version 3.2)

You are to write a program that manages teachers and students as described below:

Teacher and Student are classes that share a common information, which needs to be implemented as its own class called Person. Teacher and Student will both inherit from Person.

The Person class will contain the following informaiton:

- id - the Person's id number
- firstName - the Person's first name
- lastName - the Person's last name
- status - the Person's marital status (SINGLE/MARRIED/DIVORCED) - for this field, create a new enum type called MaritalStatus.

The Person class will contain the following mehtods:

- input() - will read from the user (using a Scanner for example) data to insert into the Person's fields.
- toString() - will print in a single line only the Person's information which will include their ID, their first name, last name and their marital status (in that order).

The Teacher class will have an additional field:

- wage - will represent the monthly wage of that teacher.

The Teacher class will override the input and toString method it inherited to include the information of the wage.

The Student class will have an additional field:

- scores - will represent a list of Score objects (explained later in this document) (You may use ArrayList or LinkedList to hold and manage the scores)

Each student will have an additional method:

- getAverageScore - will calculate and return the average score of the student.

There is no need to override the input function for Student.
However, the toString method will be overriden to include the average score of that student.

the Score class will contain the following fields:

- course - a string representing the name of the course in which the score was given.
- value - the score given for that course
- (there is no need to avoid duplicates in the students "scores" list. there can be more than one score with the same course name)

In addition to the Above, you need to create a class to represent the program (called Program). In the start of the program a list of Person will be created, which is meant to hold Student and Teacher objects. (the variable name of that list wil be called "**people**")

You may use LinkedList or ArrayList to represent **people**.

Afterewards, the program will start a loop which will print a menu, ask the user for the option to perform, and then perform it (and then the program will repeat in a loop).

The menu options are:

1. **Add Teacher** - will read the user's input to create a new **Teacher** and will add it to the list.

2. **Add Student** - will read the user's input to create a new **Student** and will add it to the list.

3. **Show All People** - will print all all the Person objects inside **people** (using toString)

4. **Show All Teachers** - will print all the Teacher objects

5. **Show All Students** - will print all the Student objects

6. **Show Teacher by ID** - will read an ID from the user and will print the Teacher in the list with that ID (if no Person with that ID exists in the list print "**Doesn't exists**" instead. If a Person exists with this ID, but it's not a Teacher, print "**Not a teacher**" instead.)

7. **Show Student by ID** - will read an ID from the user and will print the Student in the list with that ID (if no Person with that ID exists in the list print "**Doesn't exists**" instead. If a Person exists with this ID, but it's not a Student, print "**Not a student**" instead.)

   In addition, if a student with that ID is found, a sub-menu will show to allow the user to modify that Student's scores list. The sub-menu's options are as follows:

   a. **Show All Scores** - will print all the scores of the students, line by line. Each line will contain the score index, the course name and the value of the score. (The score index is the position of the score in the **scores** list. i.e. the first score is index 0, the second score is index 1 and so forth.)

   b. **Insert a New Score** - will input the user for a course name and value and will insert a new Score with that data.

   c. **Show Average Score** - will print the average score

   d. **Return to the Main Menu** - will break out of the sub-menu loop and return to the main manu loop.

8. Show Students by Score Range - will input the user for minimum and maximum score and will show all the students with the average score in that range (including those values)
9. Quit - will quit the loop, and thus, the program itself. (This option can be numbered as option 0 instead of 9)

Good Luck!