LAB 211 Assignment

Title

Student Management

Background

Write a program to manage information of a student, subject, grading. The program implements the terminology of Object-Oriented Programming (OOP) paradigm. OOP is one of the best choosing ways to design software programs.

In this assignment, you must use an HashMap to store a list of student, subject.

Program Specifications

Build a Student management program. With the following basic functions

1. Add new student
2. Update Student
   2.1 Update Student
   2.2 Delete Student
3. Add new Subject
4. Update Subject
   4.1 Update Subject
   4.2 Delete Subject
5. Enter Grade
6. Student Grade Report
7. Subject Grade Report

Others- Quit

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

Features:

This system contains the following functions:

Display a menu and ask users to select an option.

- **Function 1: Add new student - 50 LOC**
  - ✓ Require to input a student's information: student id, first name, last name, gender, date of birth, email, phone number.

  - ✓ Check the valid data with the following conditions:

    – Student id is not allowed to duplicate in the database.
    – Date of birthday must be a real date.
    – First name, last name, gender, date of birth, email, phone number not allow is null.
    – Phone number is number string which has length from 10 to 12.
    – …..

  - ✓ The new student must be appended to the end of the array list.

  - ✓ Add the student to collection.

✓ Ask to continuous create new student or go back to the main menu.

- **Function 2: Update student**
    - ○ Require enter the student id.

    - ○ If student does not exist, the notification "Student does not exist". Otherwise, user can choose the Update or Delete or back the main menu function:

        - **Function 2.1: Update student information – 50 LOC**
            - – User can edit of the in student's information. If information is blank, then not change old information.

            - – Show the result of the update: success or fail.

            - – Ask to go back to the update menu.

        - **Function 2.2: Delete student– 50 LOC**
            - – User can delete the student.

            - – Must show the confirm message before delete.

            - – Show the result of the delete: success or fail.

            - – Ask to go back to the update menu.

- **Function 3: Add new subject – 50 LOC**
    - ✓ Require to enter the subject information include subject id, subject name, credit.

    - ✓ Check the valid data with the following conditions:

        - – Subject id is not allowed to duplicate in the database.
        - – Credit must be a positive integer number.
        - – Subject name is not empty.
        - – …

    - ✓ Add subject to collection.

    - ✓ Ask to continuous create new subject or go back to the main menu.

- **Function 4: Update subject information**
    - ○ Require enter the subject id
    - ○ If subject does not exist, the notification "Subject does not exist". Otherwise, user can choose the Update or Delete or Back the main menu function
    - **Function 4.1: Update subject information – 50 LOC**
        - ✓ User can edit of the in subject's information. If information is blank, then not change old information.

        - ✓ Show the result of the update: success or fail.

        - ✓ Ask to go back to the Update menu.

- **Function 4.2: Delete Subject– 50 LOC**
  - ✓ User can delete the subject.
  - ✓ Must show the confirm message before delete.
  - ✓ Show the result of the delete: success or fail.
  - ✓ Ask to continuous or go back to the Update menu.

- **Function 5: Entering student 's grade – 100 LOC**
  - ✓ Require enter the student id, student id must be in student list. If student does not exist, the notification "Student does not exist".
  - ✓ Require enter the subject id, subject id must be in subject list. If subject does not exist, the notification "Subject does not exist".
  - ✓ If students have already graded this subject, then ask the user if you want to overwrite it or not?
  - ✓ Require enter the grade item: Labs, Progress tests, Final exam (Validation is required)
  - ✓ Add the grade item of the student to collection.
  - ✓ Show the result of the action: success or fail.
  - ✓ Ask to continuous or go back to the main menu.

Function 6: Student Grade report – 50 LOC
  - ✓ Request enter the student id, student id must be in student list. If student does not exist, the notification "Student does not exist". Otherwise, displaying student's result on the screen. For example:

    - Student ID: SE150946
    - Student name: Nguyen Tran Anh Nguyen
      List of subject sort by Subject Name:

++ No ++	+++++++Subject name ++++++++	++ Average mark ++	++ Status ++
1	C#	5.5	Pass
2	Computer Networking	6.5	Pass
3	Introduction Database	4.0	Not Pass

  - ✓ Average mark: Labs (30%), Progress tests (30%), Final exam (40%
  - ✓ Status is pass if average mark more than four. Otherwise, status will be not pass.
  - ✓ Ask to go back to the main menu.

Function 7: Subject Grade report– 50 LOC
  - ✓ Request enter the subject id, subject id must be in subject list. If subject does not exist, the notification "Subject does not exist". Otherwise, displaying list subject's result on the screen. For example:

- Subject ID: PRO301
- Student name: Java web

List of student sort by Student Name:

++ Student ID ++	++Student name ++	++ Average mark ++	++ Status ++
SE130580	Le Thanh Dat	5.5	Pass
SE150865	Nguyen Hoang Lam	6.5	Pass
SE151316	Tran Vinh An	4.0	Not Pass

✓ Ask to continuous or go back to the main menu.

- The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- The lecturer will explain the requirement only once on the first slot of the assignment.