Department of Computer Science

420-201-VA Programming 2

Course Project

Title: Human Resources Management Application
Project Due: Please check Lea for each deliverable deadline

Instructions:

This is an individual project that has three deliverables: an analysis and design document, the project code, and a test suite.

Deliverable 1 (UML class diagram and pseudocode)

For this deliverable, your repository should contain the following files:

- UML Class Diagrams for all classes
- Pseudocode for the main methods

Deliverable 2 (Implementation)

For this deliverable, your repository should contain the following files:

- Completed UML Class Diagrams for all classes and interfaces with their relationship
- The implementation of all required features (the source code)
- A snapshot of output demonstrating the functionality of the application whether correct or incorrect, complete, or incomplete.
- Graphic elements to display and input data

Note: It is suggested that you begin working on the code in Week 10, which should give you a full knowledge to complete the project. You will also find that the lessons and lab assignments will prepare you for the Course Project.

Deliverable 3 (Test Suite)

For this deliverable, your repository should contain the following files:

- Unit test to test all functional methods
- Driver class to test all functional methods in the project
- The output should be clear and well-presented containing messages to reflect the code testing.
- A snapshot showing which tests have passed.

Project Objectives

Developing a Graphical User Interface (GUI) in programming is paramount to being successful in the business industry. This project incorporates GUI techniques with other concepts that you have learned in this class. You will:

1. Apply OOP concepts, write subclasses, use aggregation.
2. Create a Graphical User Interface (GUI)
3. Read and write your class data into a .txt file
4. Test your classes
5. Create a documentation for end users

Project Description

Tiny college has asked you to be a part of their team because they need a programmer, analyst, and designer to help them in implementing a model of a human resources management system. In your model, you will have departments objects (representing departments). A department contains lists of teachers (either part-time or full-time teachers) and list of staffs; each of which belonging exclusively to one department. A department has a dean, who should be a teacher of that department.

Implementation Details

1. Implement an abstract class Person that includes at least five shared fields and contains at least one abstract method of your choice (for example to define a person's category (teacher or staff)).
2. Extend class Person with concrete classes, and override the toString() and equals() methods of each class.
3. The class Teacher has two instance variables: speciality and degree and some other fields for its subclasses (part-time or full-time teachers)
4. The class Staff has also two instance variables: duty and workload
5. Implement an interface PayRoll that contains ComputePayRoll() method to be implemented as follows:
   a. For each full-time teacher, the salary is computed as (32 * degreeRate * 2) * 0.85, where degreeRate is 112, 82, 42 for PhD, Master, and Bachelor, respectively.
   b. For each part-time teacher, the salary is computed as (hoursWorked * degreeRate *2) * 0.76, where degreeRate is 112, 82, 42 for PhD, Master, and Bachelor, respectively.
   c. For each staff, the salary is computed as (workload * 32 * 2) * 0.75, where workload is the weekly working hours. The working hours cannot exceed 40.
   d. Implement the method ComputePayRoll() inside Teacher and Staff classes.
6. Teachers and staffs are added/assigned to a department, first by loading/reading from a text file during the first execution of the application. Moreover, new teachers and staff must be added/assigned through the GUI application.

7. Trying to add a teacher or a staff member to an inexistent department (based on department id) should rise/throw an exception.
8. Trying to add a teacher or a staff member that already exists/ added (based on the id) to department should throw an exception.
9. A department class has a list of teachers and staff. A department class also has a dean, who should be a teacher of that department, otherwise an exception must be thrown.
10. Each new teacher and staff added to a department should be stored to the text file before exiting the application, generating a new version of the text file. So, if the text file does exist (which is the case), new teachers or staffs should be appended. There are files of teachers and staffs.
11. The customized exceptions handling for the cases above must be implemented.
12. Test sets should be generated for testing the effectiveness of each method in your model.

Design Requirements

1. A clear and precise GUI should be designed for input and output.
2. Create all the classes
3. Include all the supported class Libraries
4. Document your code using JavaDoc
5. Create a Test class using JUnit to test the controller methods. All tests should pass.