D0017D VT24 Re-Examination: LTU Lab Assistant Manager

August 24, 2024

1 Important

Your exam will not be eligible for grading if:

- any global variables (only the global Scanner object and the constants are allowed) are used.
- any built-in functionality for storing data (such as ArrayLists etc.) are used. In other words, the same rules as your assignments: only arrays are allowed for storing data. In other words, only import statement you will have is `import java.util.Scanner;`.
- inbuilt sorting algorithms is used, Please implement your own, for example, use the bubble sort algorithm that you have learned in the course.
- object-oriented programming is used, meaning Main is the only class you will write and no other classes.

2 Assignment description

Lab assistants are essential to help teachers with the lab assignments for different courses at LTU. Your task is to implement a system that can register the working hours of lab assistants and generate pay slips.

Menu At the start, the program should print a menu of options the user can select. Exiting the program by pressing ***q*** should be possible.

```
----------------------------------
# LTU Lab Assistant Manager
----------------------------------
1. Add lab assistant
2. Remove lab assistant
3. Register working hours
4. Print pay slip
5. Print assistant summary
q. End program
> Enter your option:
```

Add lab assistant A lab assistant should be registered in the system by providing following information:

- Name of the assistant
- Number of education credits (högskolepoäng/hp) between 0 and 400.

Once all above mentioned user input is provided, the program should generate a unique ID number between 1000 and 9999: Make sure to implement necessary checks in order to prevent two lab assistants from being assigned the same id number! Lab assistants will be referenced by this id number in other menu options.

```
> Enter your option: 1
> Enter lab assistant's name: Andreas Karlsson
> Enter number of education credits: 125

Lab assistant Andreas Karlsson was assigned ID 3988 and added to the system.
```

Remove a lab assistant A lab assistant should be able to be removed from the system by providing its ID number. Make sure to implement logic that checks whether there already is a lab assistant with provided ID number. If that is the case, an error message should be presented.

```
> Enter your option: 2
> Enter assistant's ID number: 3133

There is no lab assistant registered with ID 3133.

> Enter your option: 2
> Enter assistant's ID number: 3988

Lab assistant Andreas Karlsson was removed from the system.
```

Register working hours The program should enable the user to register a lab assistant's working hours. The lab assistant is referenced by its ID number. The user should enter the start time, end time, and date of the lab session. Make sure to implement necessary checks for date (format **YYYY-MM-DD**) and time (format **HH:MM**). In order to simplify validation of dates, assume that there is 31 days in each month (for example, there is no need to say 2024-02-31 is an invalid date, it is a acceptable date for this program). Sessions start and end on same date, So no need to ask for end date and start date. There are many other requirements here that you might think off, but none of them need to implemented. If there is no direct requirement mentioned, you don't need to implement them. For example, you don't need to check if there is already a session registered for the entered combination or if there are overlapping sessions, etc.

This information should be used to calculate assistant's salary for the session by using following rules:

- Assistant is paid for each started 30 minute time period, for example: if assistant was working between 9:45 and 10:50 (1 hour and 5 minutes), the assistant gets paid for 1,5 hours.
- If a session is shorter than one hour, the assistant is paid for one hour, for example: if assistant was working between 13:00 and 13:25 (25 minutes), the assistant is paid for one hour.
- Hourly salary depends on the amount of education credits that the assistant has
  - 120 kr per hour for assistants that have between 0 and 99 education credits.
  - 140 kr per hour for assistants that have between 100 and 249 education credits
  - 160 kr per hour for assistants that have between 250 and 400 education credits

Session salary should be presented to the user once all necessary information is provided.

```
> Enter your option: 3
> Enter lab assistant's ID number: 3988
> Enter start time of the session: 14:30
> Enter end time of the session: 15:45
> Enter date of the session: 2023-12-14

Lab assistant: Andreas Karlsson
Session time: 1 hour 45 minutes
Salary: 280 kr
```

Print pay slip It should be possible to print pay slips for a specific lab assistant (referenced by its unique ID). This functionality should present the following information:

- Assistant's name, ID number, and number of education credits.
- Summary of sessions that include:
  - Shift dates and times.
  - Shift salaries.
- Total number of sessions.
- Total salary.

The output should look similar to the example run below.

```
> Enter your option: 4
> Enter lab assistant's ID number: 3988

Pay slip LTU

Name: Andreas Karlsson (3988)
Number of education credits: 125

Sessions:

Date          Start     End       Salary
2023-10-29    10:00     12:00     280 kr
2023-11-23    08:45     09:30     140 kr
2023-11-24    14:00     16:15     350 kr

Total number of sessions: 3
Total salary: 770 kr
```

NOTE! No sorting is required for this menu option.

Print assistant summary It should be possible to print the assistant summary. The summary should consist of the following information:

- Table of all assistants sorted by their names (see example). NOTE! If you need more time, it is better to skip the sorting part. Just print unsorted data to get some points for this task.
  - Name.
  - ID number.
  - Number of credits.
  - Total salary.
- Total number of lab sessions.
- Total salary to all assistants.

```
> Enter your option: 5
LTU Lab Assistant Manager summary:

Lab assistants:
Name                ID           Credits    Salary
Andreas Karlsson    3988         125        770 kr
Maxim Khamrakulov   1701         315        320 kr
Sandeep Patil       9844         350        960 kr

Total number of sessions: 8
Total salary: 2050 kr
```

3 Instructions

The program should have necessary error handling (ways of validating the input, etc.). The program should under no circumstances crash when receiving incorrect input. It can store additional information other than provided in the assignment description. You should create appropriate data structures to store the information. All eventual floating point numbers (if any are used) should be rounded to two (2) decimals. The programs that compile with the compilation error will not be graded (it should be possible to run the program).

4 Tips

- Do one functionality/method/option at a time. It is possible to pass the assignment by not finalizing all the parts! Leave the hardest parts till the end. For example, first, get the program working for correct/valid inputs. Then, update the program to handle error conditions. Refer to the grading matrix to plan your work.
- Write down requirements for each function as comments on top of the function; this will help you not forget a requirement.

5 Assumptions

- If you notice that some information is missing from the assignment description, you are allowed to make your assumptions as long as it does not change the condition and basis of the assignment.
- It is okay to assume a fixed number for your data structures; for example, assuming that there can be a maximum of 100 lab assistants is fine. This is just an example. This is so you don't have to expand the array to add more data.

6 Requirements

- ***Please do not forget to include your name and LTU username on the top of the file as an author***
- You are free to use either replit or any IDE of your choice.
- Do not use packages (there should not be any code beginning with `package`).
- Follow the course's coding and formatting conventions (usage of constants, indentation, etc.).
- Use methods in your solution.

The assignment is to be solved independently! It is allowed to get inspiration from the Internet, but you are to solve the task independently! Always provide a source for the information that you get from external sources. **NOTE! It is not allowed to copy entire code sections!** You are, however, allowed to copy the code that you have written yourself.

7 Submission

You must submit a single Java file to Canvas.

- Make sure your filename is called "'Main.java"' and your class name is called "'Main"'. Upload this single file Main.java to canvas.
- You can upload multiple times to Canvas.