Spring 2021 – CS203 Object Oriented Programming
Homework 3

Deadline: 03/21/2021 Sunday 11:59 pm

UAB Hospital System

Objectives:

- To practice types, objects, classes, inheritance and arrays in JAVA

UAB Hospital System

The goal of this assignment is to create a hospital employee tracking system for the UAB hospitals. You will create a Java project to automate the adding, deleting and displaying the list of UAB hospital employees. The set of classes define these employees of a hospital; *hospital employee, doctor, nurse, administrator, surgeon, receptionist,* and *janitor.*
Your code initially will read a list from txt file (uabEmployee.txt) create the corresponding objects. You will store the objects in java arrays and when you terminate the program you will save the updated list of the employees into same txt file.

Homework Instructions

- You must use inheritance in creating those classes (Penalty = 20 points)
- You can use arrays to store objects of the same Class, you can assume maximum size = 10 or one array to store all objects of all the Classes, you can assume maximum size = 50.
- When you start the program, you will read the uabEmployee.txt file line by line. Each line will be containing information for one specific object. You will create these objects and store them in your object array (program memory)
- You need to display a menu with the following options; `Display the employee list, add employee, update the database`, and an optional `delete employee` feature.
- According to the user's selection from the menu, your code should be able to
    - add a hospital employee, doctor, nurse, administrator, surgeon, receptionist, and janitor
    - remove a hospital employee, doctor, nurse, administrator, surgeon, receptionist, and janitor *given the role and the blazerId* (Bonus question → 20 points)
    - display the hospital employees in the format given below
    - update the database (save the updated object list into the same txt file)
- Use the Constructors to automatically initialize the instance variables (they should be all declared as private, Penalty = 20 points)
- Create the appropriate accessor and mutator methods for each private instance variables (setters/getters)
- Add (overwrite) the methods to `display`
    - Use `toString` method and `super` for printing the employee list

- You will create the necessary attributes and methods for each class (check the sample input and output to decide the methods and attributes)
- **A sample uabEmployee.txt file;**

E Lucy 1995
D John 1187 Gastro
N Julia 1254 4
A Jacob 4123 Business
A Catherine 4125 Management
R Alice 4101 Talking Y
J David 4789 Maintenance N
D Samuel 4891 Radiology
S Steve 4345 Brain Y

- **An expected output is;**

****
Welcome to the UAB Employee System
****
The UAB Hospital System has the following employees:

Total Number of employees = 9

Hospital Employees: 1
Name: Lucy BlazerId: 1995

Doctors: 2
Name: John BlazerId: 1187 Specialty: Gastro
Name: Samuel BlazerId: 4891 Specialty: Radiology

Surgeons: 1
Name: Steve BlazerId: 4345 Specialty: Brain Operating: Y

Nurses: 1
Name: Julia BlazerId:1254 Number of Patients: 4

Administrators: 2
Name: Jacob BlazerId: 4123 Department: Business
Name: Catherine BlazerId: 4125 Department: Management

Receptionist: 1
Name: Alice BlazerId: 4101 Department: Talking Answering: Y

Janitors: 1
Name: David BlazerId 4789 Department: Maintenance Sweeping: N

****

- The explanation of the input file;
  - **E** stands for Hospital Employee, **Lucy** for name and **1995** for the blazerId
  - **D** stands for Doctor, **John** for name, **1187** for blazerId, and **Gastro** for the specialty
  - **N** stands for Nurse, **Julia** for name, **1254** for blazerId, and **4** for number of patients
  - **A** stands for Administrator, **Jacob** for name, **4123** for blazerId, and **Business** for department
  - **A** stands for Administrator, **Catherine** for name, **4125** for blazerId, and **Management** for department
  - **R** stands for Receptionist, **Alice** for name, **4101** for blazerId, **Talking** stands for department, and **Y** stands for answering
  - **J** stands for Janitor, **David** for name, **4789** for blazerId, **Maintenance** for department, **N** for sweeping
  - **D** stands for Doctor, **Samuel** for name, **4891** for blazerId, **Radiology** for department
  - **S** stands for Surgeon, **Steve** for name, **4345** for blazerId, **Brain** for department, **Y** for operating.

- You need to create an efficient user interface (not a Graphical User Interface, just use `system.out` class to print and get inputs). You should ask the necessary questions to add/remove employees
  - For example; if the user wants to add a new Surgeon;
    - You need to ask, the name, blazerId, department and whether operating or not
  - If the user wants to remove any employee
    - Ask the role and the blazerId (assume we have unique blazerId in the database)

- ***Deliverables:***
  a. A well written project report (.pdf)
     i. Explain your classes, attributes and methods. Explain how you implemented the Inheritance, and benefits of this implementation
     ii. Put the screenshots of your outputs (the working version of your code)
  b. Project.zip file
     i. Submit the working version of the whole project file
     ii. Do not submit only the classes
  c. Independence_completion_form

If any of the problem statements is unclear, use the Canvas discussion board to ask for clarifications.