****

School of Engineering Business and Technology

**PROJECT**

**OBJECTIVES:**

The project will give you the opportunity to review and practice on the following:

-Types of Data structure: Array Based Structure, Linked List structures, Hashed Data Structure or Binary Search Tree Data structure

-Condition structure, loop structure

-read from keyboard

-working with the file: -file: open, check file existing, read file, write to file

-class, object: constructor, mutator methods, accessor methods, toString and others, and how to access the class members from client site

-display information on screen with requested format on numbers

-Handling input errors

- Write comments

**REQUIREMENT:**

Write an application for the Employee system named EmployeeSystem\_yourLastname that allows the company can do the following tasks with the information of employees:

1. Add One new employee/employeeWithTitle
2. Remove One employee by employeeID
3. Update information of one employee (department, salary, title, contact information (phone, address)
4. Search One Employee by employee ID
5. List Employees with the same Last name
6. List Employees with the same title
7. List Employees in one Department

The information of all employees who are working at the company is stored in a text file named employeeInfo\_yourLastName.txt where the information of one employee is written on one line in the file separating by comma:

Id, lastname, firstname, hired year, salary, phone, address, department (if he/she is an employee)

Id, lastname, firstname, hired year, salary, phone, address, department, title, bonus (if he/she is an employee with a title)

For example:

1234567, William, Thomas, 2012, 55000.0, 2141234567, 111 Walnut Rd Dallas TX 75243, IT, Manager, 2000.0

1122334, Garcia, Brian, 2011, 35000.0, 9721112323, 222 Plano Rd Dallas TX 75081, Warehouse

TASK1: Add One New Employee/employeeWithTitle

This task helps to insert one new employee/employeeWithTitle

-Display messages to ask and read information from the keyboard of one employee. If the employee is an employee with a title ask for the title and bonus then insert it to the data structure

-Show all the structure to ensure the new employee is added

TASK2: Remove One employee with employee ID as key search

This task asks for the employ id as a key, then search in the structure. If it is found, display the information on the screen; if it is not found, display the message “the ID cannot be found”

-Show all the structure to see the miss

TASK3: Update information of one employee (department, salary, title, contact information (phone, address)

This task is to update the information of one employee.

-Display the question to ask for the employee id who needs to be updated the information

-Use the employee id to fetch the employee

-Display the message to ask what is new information (department, salary, title or contact information) ?

-Read the new information

-Update the information to the node

-Update the node with the key by the new updated node

-Show all the structure to see the change

TASK4: Search One Employee by employee ID

This task use the fetch operation to search the employee by employee id

-Ask for the employ id, then fetch the node from the data structure and display the infor of the node

TASK5: List Employees with the same Last name

This task searches and displays all the list of employees who have the same last name

-Ask for the Last name as search key to search

-search all the employees who have the last name as the search key and display all of them on the screen

Note: You should provide a new method in the data structure class to search nodes by the last name and display all of them on the screen

TASK6: List Employees with the same title

This task searches and displays all the list of employees who have the same title

-Ask for the title to seach

-Searh and display on the screen all the employees who has the same tittle

Note: You should provide a new method in the data structure class to search nodes by the title and display all of them on the screen

TASK7: List Employees in one Department

This task searches and displays all the list of employees who are working at the same department

-Ask for the name of the department

-search and display on the screen all employees who are in the department

Note: You should provide a new method in the data structure class to search nodes by the department and display all of them on the screen

Output requested for one employee:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID: xxxxxxx

Employee name: LASTNAME, Firstname

Hired year: xxxx

Salary: xxxxx.x

Phone: xxxxxxxxxx

Address: xxx xxxxxxxxxxxxx

Department: xxxxxxxx

Tittle: xxxxxxx

Bonus: xxxxx.x

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID: 1234567

Employee name: WILLIAM, Thomas

Hired year: 2012

Salary: 55000.0

Phone: 2141234567

Address: 111 Walnut Rd Dallas TX 75243

Department: IT Department

Tittle: Manager

Bonus: 2000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OR

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID: 1122334

Employee name: GARCIA, Brian

Hired year: 2011

Salary: 35000.0

Phone: 9721112323

Address: 222 Plano Rd Dallas TX 75081

Department: Warehouse Department

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HOW TO DO THE PROJECT

-Read the requirement to understand what the requirement asks for

-Determine how many data type classes we need for this project: class Employee and class EmployeeWithTitle can be reused

-You decide what type of data structure you use in this project

Notes:

-Create one object of the data structure

-When the application starts, you should read all the information of employees in the input file named employeeInfo\_yourLastName.txt.

-The information of each employee is saved in one line in the file

-For each employee read from the input file, create a node then insert to the data structure

-After finishing reading, display the menu to allow users to select the task

-During the process of the application EmployeeSystem\_yourLastName, the tasks on the employees are only happened on the data structure

-When the users select the Exit task, before terminating the program, open the input file employeeInfo\_yourLastName.txt and write all the information of employees in the data structure back to the file to save for the future used

HOW TO TURN IN

-All the source code .java files

-File employeeInfo\_yourLastName.txt.

-Pseudo code of your project in word document

HOW TO GRADE THE PROJECT

|  |  |
| --- | --- |
| Class Employee\_yourLastName.java data members, constructor, toString, and other methods | 2 |
| Class EmployeeWithTitle\_yourLastName.java, data members constructure, toString, and other methods | 3 |
| class Data structure (you can choose any data structure type we learned) |  |
| Data structure: Method insert new employee | 2 |
| Data structure: Method search by employee id | 2 |
| Data structure: Method search by lastname | 3 |
| Data structure: Method search by title | 3 |
| Data structure: Method search by department | 3 |
| Data structure: Method update | 2 |
| Data structure: Method show all to print out all employees | 2 |
| Controlling class EmployeeSystem\_yourLastName. |  |
| Pseudo-code of main() | 5 |
| Menu – handle menu to terminate when user select exit | 2 |
| Initial the data structure | 2 |
| Read file employeeInfo\_yourLastName.txt, split information, create the node and add to the data structure- Open and close file | 5 |
| Task Add one new employee – read info and insert | 2 |
| Task Remove one employee by employee ID | 2 |
| Task: Update information of one employee | 3 |
| Task: Search one employee by employee id | 2 |
| Task: List employees with the same last name | 2 |
| Task: Liet employees with the same title | 2 |
| Task: List employees with the same department | 2 |
| Task: Exit – open/close output file – Write all nodes back to the file employeeInfo\_yourLastName.txt | 4 |
| Compile success – qualify the requirement | 10 |
| Write the comment | 5 |
| Turn in on time | 10 |
| Project scores | 80 |

1. Add One new employee/employeeWithTitle
2. Remove One employee by employeeID
3. Update information of one employee (department, salary, title, contact information (phone, address)
4. Search One Employee by employee ID
5. List Employee with the same Last name
6. List Employees with the same title

List Employees in one Department