Project Assignment

Title

Employee management in a computer company

Background

A software company needs to manage employees in the company. In this exercise, students will have to implement some components in the employee management system. The main objects to be managed are Developers, TeamLeader and Testers.

The system is required to support the following queries:

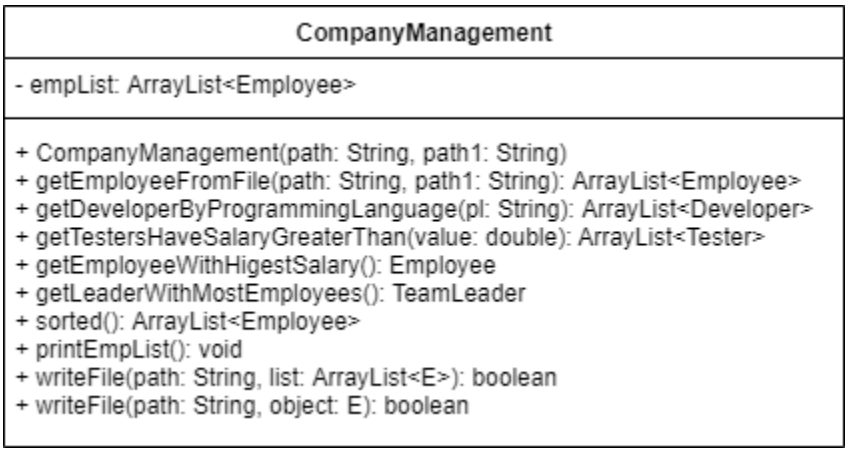
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending salary
7. Write file

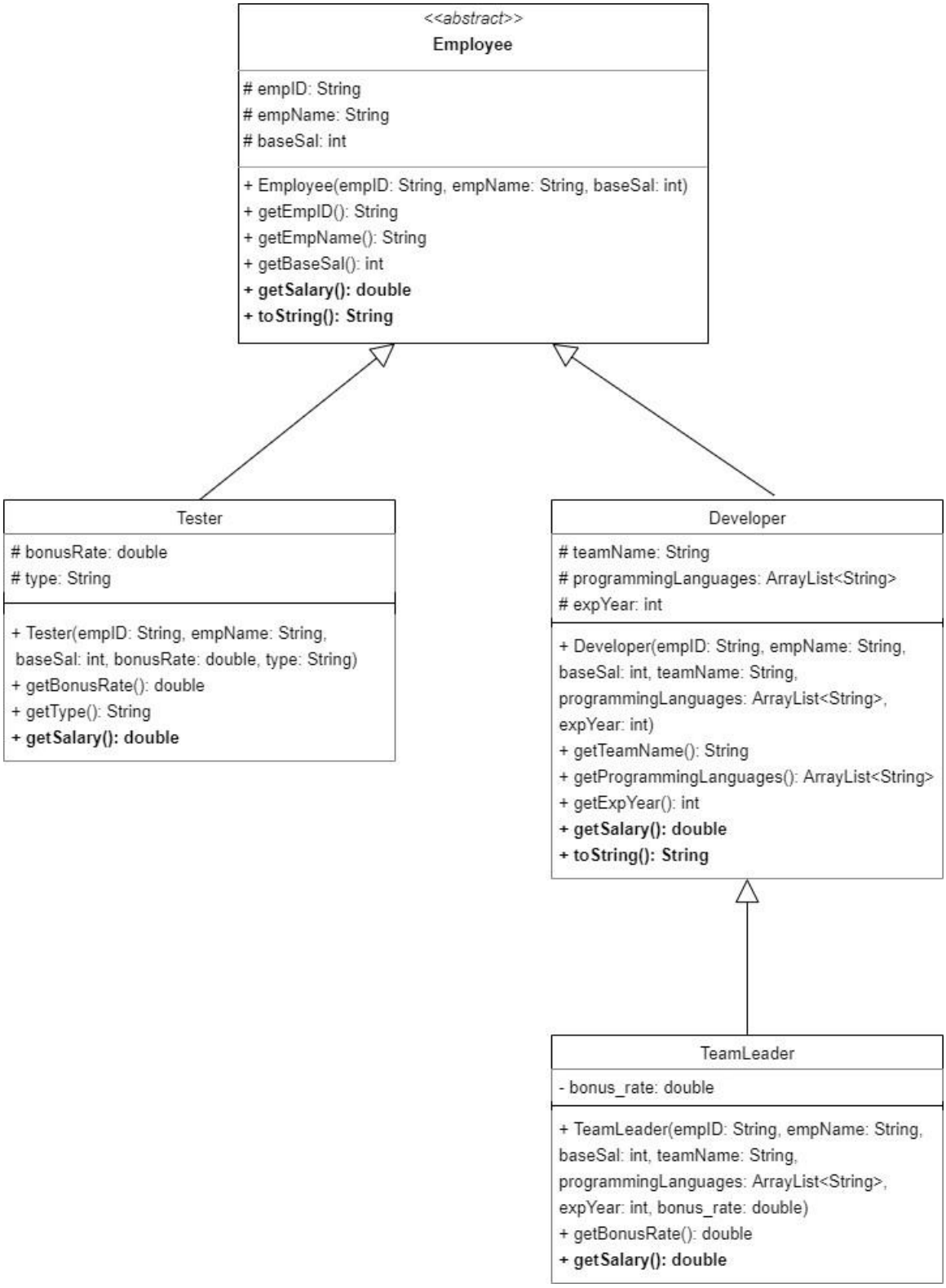
- *Req2.txt*: result list of staff proficient in C++.

- *Req3.txt*: list of employees with salary > 4,700,000.

8. Exit

I. Design





II. Description

- **Developer** and **Tester** inherit from **Employee** class and **TeamLeader** inherit from **Developer** class.

- Explain some properties and methods as follows:

2.1. Employee class

- ***empID***: employee code.

- ***empName***: employee name.

- ***baseSal***: base salary of the employee.

- The abstract method ***getSalary().***

- ***toString()***: returns the string in the format:

empID_empName_baseSal

2.2. CompanyManagement class

- ***empList***: list of employees

- ***CompanyManagement(String path, String path1)***: contructor method, initialize empList by calling file read method.

- ***getEmployeeFromFile(String path, String path1):*** reads from the file into the empList list with path being the path to the *ListOfEmployees.txt* file containing the list of employees and path1 being the path to the *PLInfo.txt* file containing the list of programming languages that every employee is proficient.

- ***getDeveloperByProgrammingLanguage(String pl)***: returns a list of programmers who are proficient in the input pl language.

- ***getTestersHaveSalaryGreaterThan(double value):*** returns a list of testers whose total salary is greater than the value of the parameter passed.

- ***getEmployeeWithHigestSalary():*** returns the employee with the highest salary in the list of employees.

- ***getLeaderWithMostEmployees():*** returns the team leader of the group with the most programmers.

- ***sorted():*** returns the sorted list of employees empList. The method returns a list of Employees sorted by salary descending. If the salary is equal, then sort by the first letter (in alphabetical order) of the Employee's name (name is the last word in the last name). The sorting is done on a new list copied from the *empList*

- ***printEmpList():*** print *empList* to the command prompt screen.

- ***printEmpList(ArrayList<Employee> list):*** print *list* to the command prompt screen.

- ***writeFile(String path, ArrayList<E> list):*** write file of employee list

- ***writeFile(String path, E object):*** write the required employee file.

2.3. Developer Class

- ***teamName***: the team name of the programmer. Each programmer belongs to only one group.

- ***programmingLanguages***: a list of programming languages that programmers are proficient in.

- ***expYear***: the number of years of experience of the programmer.

- ***getSalary***(): returns salary, as described below.

- ***toString()***: returns the string in the format:

+ empID_empName_baseSal_teamName_programmingLanguages_expYear

+ programmingLanguages in the format*:*

[programmingLanguages1, programmingLanguages2,…]

2.4. Tester class

- ***bonusRate***: rate of additional income.

- ***type***: type of tester (Automation Test - AM/Manual Test - MT).

- ***getSalary()***: returns salary, as described below.

2.5. TeamLeader class: TeamLeader is also a programmer, each team has only 01 TeamLeader.

- ***bonus_rate***: rate of additional income.

- ***getSalary():*** returns salary, as described below.

2.6. Description of getSalary() method:

- Developer

+ If number of years of experience >= 5:

*Salary = basic salary + years of experience * 2,000,000 (2 million).*

+ 5 > Years of experience >= 3

*Salary = base salary + years of experience * 1,000,000 (1 million).*

+ The remaining case:

Salary = base salary.

- TeamLeader

If **Developer** is **TeamLeader**, there will be additional income.

*Salary = developer salary + bonusRate * developer salary*

- Tester

*Salary = base salary + bonusRate * base salary*

III. Description of Input file

- The input file is named ***ListOfEmployees.txt*** contains a list of employees with each line corresponding to an employee's attribute separated by a commus (*",")* in the format:

- Developer

Ordinal number, Employee ID, Employee name, Team name, Years of experience, Basic salary

```
1,D01,Nguyen Dinh Minh Khoi,Run,1,5000000
2,D02,Pham Le Anh Khoa,Fly,3,6000000
```

- TeamLeader

Ordinal number, Employee ID, Employee name, Team name, Years of experience, The letter L indicates this is TeamLeader, Bonus rate, Basic salary

```
6,D04,To Quoc Bao,Walk,3,L,0.3,10000000
```

- Tester

Ordinal number, Employee ID, Employee name, Bonus rate, Type of tester, Base salary

```
3,T01,Vu Bao Dang Khoa,0.2,AT,3000000
4,T02,Truong Pham Thao Mi,0,MT,2000000
```

IV. Description of output file and project

1. Option 1

```
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending  salary
7. Write file
8. Exit
Your options from 1 - 8:1
D01_Nguyen Dinh Minh Khoi_5000000_Run_[C, C#, C++]_1
D02_Pham Le Anh Khoa_6000000_Fly_[C, Java]_3
T01_Vu Bao Dang Khoa_3000000
T02_Truong Pham Thao Mi_2000000
D03_Tran Tuan Kiet_12000000_Run_[Java, Java Script]_7
D04_To Quoc Bao_10000000_Walk_[C, C++]_3
D05_Tran Bao Tin_10000000_Jump_[Java, C#]_3
D06_Le Van Nam_8000000_Jump_[PHP, Ruby]_1
T03_Pham Thi Nhu_6000000
T04_Nguyen Duy An_5000000
D07_Nguyen Quoc Tin_15000000_Fly_[C, Golang]_5
D08_Dung Cam Quang_13000000_Run_[C++, Java]_3
D09_Nguyen Quang Duy_10000000_Walk_[Java, PHP, Java Script]_5
D10_Dang Thanh Tu_9000000_Run_[Java]_2
D11_Huynh Tan Hung_12000000_Fly_[Golang, Swift]_6
T05_Vu Thi Hanh_4000000
T06_Truong My Lien_7000000
T07_Do Thi Nhan_8000000
D12_Tran Duc Minh_9000000_Walk_[C, Java, PHP]_3
```

2. Option 2

```
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending  salary
7. Write file
8. Exit
Your options from 1 - 8:2
Input Programming Language: Java
D02_Pham Le Anh Khoa_6000000_Fly_[C, Java]_3
D03_Tran Tuan Kiet_12000000_Run_[Java, Java Script]_7
D05_Tran Bao Tin_10000000_Jump_[Java, C#]_3
D08_Dung Cam Quang_13000000_Run_[C++, Java]_3
D09_Nguyen Quang Duy_10000000_Walk_[Java, PHP, Java Script]_5
D10_Dang Thanh Tu_9000000_Run_[Java]_2
D12_Tran Duc Minh_9000000_Walk_[C, Java, PHP]_3
D13_Nguyen Thanh Quan_10000000_Walk_[C++, Java, PHP]_5
Input Salary:
```

3. Option 3

```
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending  salary
7. Write file
8. Exit
Your options from 1 - 8:3
Input Salary: 500000
T01_Vu Bao Dang Khoa_3000000
T02_Truong Pham Thao Mi_2000000
T03_Pham Thi Nhu_6000000
T04_Nguyen Duy An_5000000
T05_Vu Thi Hanh_4000000
T06_Truong My Lien_7000000
T07_Do Thi Nhan_8000000
```

4. Option 4 and option 5

```
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending  salary
7. Write file
8. Exit
Your options from 1 - 8:4
D07_Nguyen Quoc Tin_15000000_Fly_[C, Golang]_5
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending  salary
7. Write file
8. Exit
Your options from 1 - 8:5
D04_To Quoc Bao_10000000_Walk_[C, C++]_3
```

5. Option 6

```
1. Read all Employees and print to screen
2. Show staff proficient in a Programming Language
3. Show Tester has a height salary
4. Show Employee's higest salary
5. Show Leader of the Team has most Employees
6. Sort Employees as descending  salary
7. Write file
8. Exit
Your options from 1 - 8:6
D07_Nguyen Quoc Tin_15000000_Fly_[C, Golang]_5
D03_Tran Tuan Kiet_12000000_Run_[Java, Java Script]_7
D11_Huynh Tan Hung_12000000_Fly_[Golang, Swift]_6
D09_Nguyen Quang Duy_10000000_Walk_[Java, PHP, Java Script]_5
D13_Nguyen Thanh Quan_10000000_Walk_[C++, Java, PHP]_5
D08_Dung Cam Quang_13000000_Run_[C++, Java]_3
D04_To Quoc Bao_10000000_Walk_[C, C++]_3
D05_Tran Bao Tin_10000000_Jump_[Java, C#]_3
D12_Tran Duc Minh_9000000_Walk_[C, Java, PHP]_3
T07_Do Thi Nhan_8000000
D02_Pham Le Anh Khoa_6000000_Fly_[C, Java]_3
D10_Dang Thanh Tu_9000000_Run_[Java]_2
D06_Le Van Nam_8000000_Jump_[PHP, Ruby]_1
T03_Pham Thi Nhu_6000000
```

6. Option 7


Source Packages
<default package>
CompanyManagement.java
Developer.java
Employee.java
Req2.txt
Req3.txt
TeamLeader.java
TestCM.java
Tester.java
input
ListOfEmployees.txt
PLInfo.txt