TND002: O-O programming

Lab4: Inheritance 2

Summary: You have to implement code that manages personnel for a company. The company employs two types of employees: directors and workers. Each director leads several workers. Your code should keep track of the employees. The code should be able to sort the employees either alphabetically by the surname, by the salary or by the taxes paid by the employees. The sorted list should be written out to the console. The code consists of a class **Company**, the abstract superclass **Employee** and its concrete subclasses **Director** and **Worker**. The main method is located in the class **Lab4**.

Preparation: This lab covers the topics discussed in lectures 6, 9 and 10. You should also complete Lesson 3 before you come to the lab. You should know about the following issues.

- What is the difference between abstract classes and concrete classes?
- How does the interface Comparable work?
- How do you use JavaDoc to document code?

You can show your completed lab during the sessions for lab 4 and 5.

Part A: Implement the classes

You find a Javadoc description of the code structure and what each method should do (click on index to start with). Only members set *public* are visible. You also find the implementation of **Lab4** for tasks A and B. You can switch in its main method between the three sorting options. Your code should produce the output shown on the next page.

Regarding where you place the *implements Comparable* and where you implement its abstract method: you have to sort all workers and directors together but the taxes are calculated differently for directors and workers. A worker pays 25% taxes of the salary while directors pay 25% taxes of their (salary + bonus). The director's bonus equals 10% of the summed salaries of all of the employees that are managed by him/her.

Employee has the variables: *firstName* (String), *secondName* (String), *employeeNumber* (int), *salary* (double), *sortCriterion* (int) (default value 0) and the constants *BYNAME=0, BYSALARY=1, BYTAXES=2* that are used for *sortCriterion.*

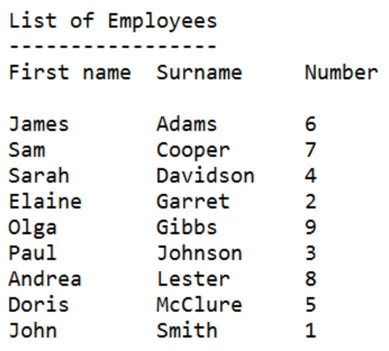
Additional variables that are global to a class: **Company** has one dynamic array that stores all workers and directors once. **Director** has one dynamic array that stores the workers (and only workers) that are assigned to this director.

Part B: Document your code and run Lab4

Document your code such that you get Java documentation that is similar to that you found in the source directory.

Your code works correctly if you get the following console output for the three selection criteria. Your code should format the output properly like the one shown.

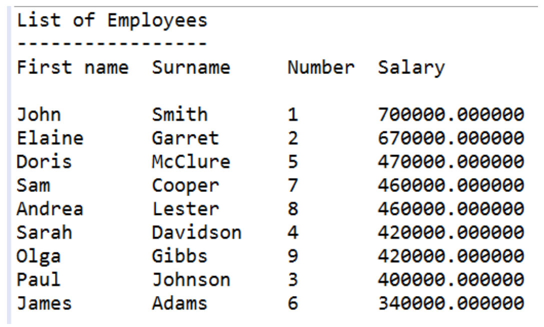
Sorting option BYNAME



```
List of Employees
-----------------
First name  Surname     Number

James       Adams       6
Sam         Cooper      7
Sarah       Davidson    4
Elaine      Garret      2
Olga        Gibbs       9
Paul        Johnson     3
Andrea      Lester      8
Doris       McClure     5
John        Smith       1
```

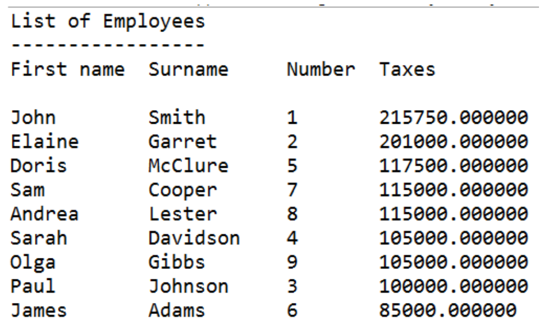
Sorting option BYSALARY



```
List of Employees
-----------------
First name  Surname     Number  Salary

John        Smith       1       700000.000000
Elaine      Garret      2       670000.000000
Doris       McClure     5       470000.000000
Sam         Cooper      7       460000.000000
Andrea      Lester      8       460000.000000
Sarah       Davidson    4       420000.000000
Olga        Gibbs       9       420000.000000
Paul        Johnson     3       400000.000000
James       Adams       6       340000.000000
```

Sorting option BYTAXES

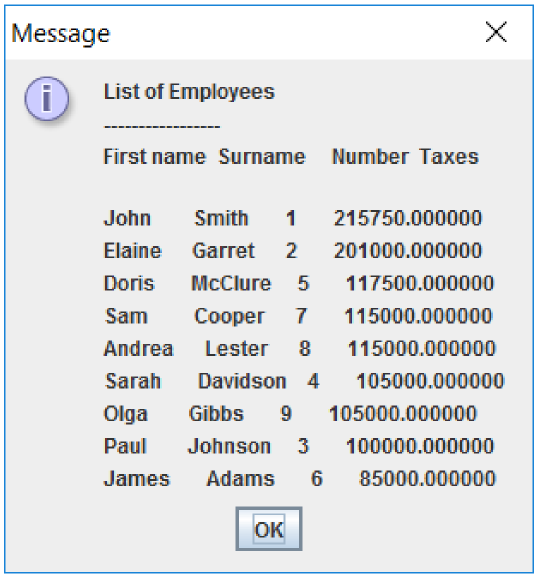


```
List of Employees
-----------------
First name  Surname     Number  Taxes

John        Smith       1       215750.000000
Elaine      Garret      2       201000.000000
Doris       McClure     5       117500.000000
Sam         Cooper      7       115000.000000
Andrea      Lester      8       115000.000000
Sarah       Davidson    4       105000.000000
Olga        Gibbs       9       105000.000000
Paul        Johnson     3       100000.000000
James       Adams       6       85000.000000
```

Part C: Implement pop-up windows

The final task is to use popup windows to switch between the selection criteria and to use a GUI (lecture 10, page 10) to display the sorted list. The popup window should ask you "You want to sort by Surname (1), Salary (2) or paid Taxes (3)?". If you typed in a string that can not be converted into an integer value between 1 and 3, then a second popup window should say "Try again!". This cycle should be repeated until a value 1, 2 or 3 has been typed in. Another popup window should display the sorted list of employees. The text string is no longer formatted properly. You don't have to fix that for the popup window. The result should look like that below.



```
List of Employees
-----------------
First name  Surname    Number  Taxes

John      Smith      1     215750.000000
Elaine    Garret     2     201000.000000
Doris     McClure    5      117500.000000
Sam       Cooper     7      115000.000000
Andrea     Lester    8      115000.000000
Sarah      Davidson   4      105000.000000
Olga      Gibbs      9     105000.000000
Paul      Johnson    3      100000.000000
James      Adams      6      85000.000000
```

Part D: Demonstration

Run the original class **Lab4** and show that your code reproduces the outcome of **Part B** (comment out the code for **Part C**). Show the code you implemented to the lab assistant as well as the JavaDoc documentation you produced. Uncomment the code that launches the pop-up windows and show to the lab assistant that this code works too.