Programming 2 – Lab14 – U4 GR2 A

Class `Workplace` represent group of employees. Internally employees (object of `Employee` class) are stored in STL list container (`list<Employee> employees`). Implement missing code in class `Workplace`.

Part 1 (1 point)

Implement method `bool AddEmployee(Employee e)` which adds employee, passed as an argument, to the list `employees`. Use list method `push_back`. Method `AddEmployee` returns true if employee was successfully added. Implement also `friend ostream& operator<<(ostream& out, const Workplace& w)`, which prints info about workplace (about all employers) to the screen. Internally use algorithm `for_each`.

Part 2 (1 point)

Modify implementation of method `AddEmployee`, so that method is checking whether employee we trying to add (with given firs and last name) already exist in the list or no. Internally use algorithm `find_if` and a properly defined lambda expression to verify whether element exist or no. If employee already exist, method return false without adding.

Part 3 (1 point)

Implement method `bool AddEmployee(Employee e)` which gets all information about employee form the user (form keyboard), creates new employee and adds it to the list `employees`.

Part 4 (1 point)

Implement method `void SortByName()` which sorts all employees according to last name and first name respectively. Internally use method sort form list and add necessary comparing functionality (in any way).

Part 5 (1 point)

Implement method `Employee MaxSalary() const` which returns employee with highest salary. Internally use algorithm `max_element` and a properly defined lambda expression.

Part 6 (1 point)

Implement method `void RemoveSalaryAbove(int lim)` which removes all employees with salary above limit (`lim`), passed as an argument. Internally use list method `remove_if`. Additionally, define a functor `SalaryLimit`.

Part 7 (1 point)

Implement method `void meanSalary(string position) const` which prints to the screen employees belonging to the same position and calculated mean salary for that sub-group. Internally copy to the helping list all employees belonging to the same position (use `copy_if` algorithm and a properly defined lambda expression). Print to the screen that helping list (using `for_each`). Calculate

mean salary for that sub-group stored in helping list (use `accumulate` algorithm and a properly defined lambda expression to calculate sum of salary and then divide by number of employees).

Part 8 (1 point)

Implement method `map<string, double> salaryEqualization() const` which returns a map keeping proposition of salary equalization for all possible positions. Inside map for single element a key is a name of a position and a value is a mean salary for that position. Remember to first identify all possible positions (`transform` algorithm can be used). Than for each possible position we need to copy all employees with given position and calculate mean salary (same as in part 7). Finally we add to map element position – mean salary.

Example program output

```
************************ Part 1 () ************************
 | Kazimierz      | Wielki           | programmer         |    6000 |
 | Major          | Major            | sales manager      |    5000 |
 | Piotr          | Kowalski         | senior programmer  |   11000 |
 | Jan            | Kowalski         | junior programmer  |    3000 |
 | Anna           | Smith            | programmer         |    7000 |
 | James          | Bond             | senior programmer  |   13000 |
 | Sherlock       | Holmes           | programmer         |    4321 |
 | Julius         | Caesar           | CEO                | 1200000 |
 | Karolina       | Lewandowska      | project manager    |   15000 |
 | Jan            | Dzban            | programmer         |    5000 |

************************ Part 2 () ************************
Ok - Employeer alredy added!

************************ Part 3 () ************************
First name:   John
Last name:   Been
Position:   CEO
Salary:   1100000
Ok - Employeer added!
 | Kazimierz      | Wielki           | programmer         |    6000 |
 | Major          | Major            | sales manager      |    5000 |
 | Piotr          | Kowalski         | senior programmer  |   11000 |
 | Jan            | Kowalski         | junior programmer  |    3000 |
 | Anna           | Smith            | programmer         |    7000 |
 | James          | Bond             | senior programmer  |   13000 |
 | Sherlock       | Holmes           | programmer         |    4321 |
 | Julius         | Caesar           | CEO                | 1200000 |
 | Karolina       | Lewandowska      | project manager    |   15000 |
 | Jan            | Dzban            | programmer         |    5000 |
 | John           | Been             | CEO                | 1100000 |


************************ Part 4 () ************************
 | John           | Been             | CEO                | 1100000 |
```

James	Bond	senior programmer	13000
Julius	Caesar	CEO	1200000
Jan	Dzban	programmer	5000
Sherlock	Holmes	programmer	4321
Jan	Kowalski	junior programmer	3000
Piotr	Kowalski	senior programmer	11000
Karolina	Lewandowska	project manager	15000
Major	Major	sales manager	5000
Anna	Smith	programmer	7000
Kazimierz	Wielki	programmer	6000

********** Part 5 () **********
| Julius | Caesar | CEO | 1200000 |

********** Part 6 () **********
James	Bond	senior programmer	13000
Jan	Dzban	programmer	5000
Sherlock	Holmes	programmer	4321
Jan	Kowalski	junior programmer	3000
Piotr	Kowalski	senior programmer	11000
Karolina	Lewandowska	project manager	15000
Major	Major	sales manager	5000
Anna	Smith	programmer	7000
Kazimierz	Wielki	programmer	6000

********** Part 7 () **********
Jan	Dzban	programmer	5000
Sherlock	Holmes	programmer	4321
Anna	Smith	programmer	7000
Kazimierz	Wielki	programmer	6000
Mean programmer salary: 5580.25

********** Part 8 () **********
Proposition of salary equalization
junior programmer - 3000
Proposition of salary equalization
programmer - 5580.25
Proposition of salary equalization
project manager - 15000
Proposition of salary equalization
sales manager - 5000
Proposition of salary equalization
senior programmer - 12000