An Organisation ABC wants to store a list of employee records.

An Employee will have following attributes

```
int id;
String name;
int age;
int salary; // per annum
String department;
String city;
```

Company wants to display all the details of employees in the organization.

Company wants to display the names of all the employees in Sorted alphabetical order.

Company wants to get the count of employees from each city.

Company wants to get the monthly salary of each employee.

Use a suitable data structure to implement the above requirements.

Sample Output

List of employees
1 Aman 20 1100000 IT Delhi
2 Bobby 22 500000 Hr Bombay
3 Zoe 20 750000 Admin Delhi
4 Smitha 21 1000000 IT Chennai
5 Smitha 24 1200000 Hr Bengaluru

Names of all employees in the sorted order are
[Aman, Bobby, Smitha, Smitha, Zoe]

Count of Employees from each city
{Bengaluru=1, Bombay=1, Chennai=1, Delhi=2}

Monthly Salary of employee along with their id is
{1=91666.0, 2=41666.0, 3=62500.0, 4=83333.0, 5=100000.0}

Instructions

Follow the following coding standards in the above assignment.

1) Use setters and getters to set or get the values of each attribute.
2) Follow Naming conventions throughout the code.
3) Create a Separate class for each of the data structure implementations.
4) Throw Illegal Argument Exception if
    a) Employee Id < 0
    b) Name is empty or null
    c) Age < 0
    d) Salary < 0
    e) Department is empty or null
    f) City is empty or null.

Stub Code

```
public class Employee {

        int id;
        String name;
        int age;
        int salary; // per annum
        String department;
        String city;
.
.
}
```

```
public class DataStructureA{

        public void sortingNames(ArrayList<Employee> employees) {

 .
.
.
.

}
```

```
public class DataStructureB {

        public void cityNameCount(ArrayList<Employee> employees) {

.
.
.
        public void monthlySalary(ArrayList<Employee> employees) {


.
.
.         try{


            }
        Catch{}
```

```
public static void main(String[] args) {

                ArrayList<Employee> employees = new ArrayList<>();

                Employee e1 = new Employee();
                e1.setId(1);
                .
                .
```