2. C#: Employees Management

Given a list of data, implement the following 3 methods using LINQ in a class that allows for managing employees:

1. *AverageAgeForEachCompany* - calculates the average age of employees for each unique company and returns the results as a **Dictionary<string, int>** sorted by key, where Key[string] is the unique company name, and Value[int] is the average age of employees in this company. The average age is rounded to the nearest whole number.
2. *CountOfEmployeesForEachCompany* - calculates the count of employees for each unique company and returns the results as a **Dictionary<string, int>** sorted by key, where Key[string] is the unique company name, and Value[int] is the count of employees in this company.
3. *OldestAgeForEachCompany* - finds the oldest aged employee for each unique company and returns the results as a **Dictionary<string, Employee>** sorted by key, where Key[string] is the unique company name, and Value[Employee] is the oldest employee in this company.

Here is the description of the Employee class:

- *FirstName* [string] - the first name of the employee.
- *LastName*[string] - the last name of the employee.
- *Company* [string] - the company of the employee.
- *Age* [int] - the age of the employee.

Your implementation of the class will be tested by a stubbed code on several input files. The input file contains parameters for the function calls (i.e., the employee data). The functions will be called with those parameters, and the result of their executions will be printed to the standard output by the stubbed code.

▼ Input Format For Custom Testing

The first line contains an integer, n, denoting the number of employees on which operations have to be performed.
Each line i of the n subsequent lines (where $0 \leq i < n$) contains space-separated strings, such that the first of them is the first name of the employee, the second is the last name of the employee, the third is the company of the employee, and the fourth is the age of the employee, respectively.

▼ **Sample Case 0**

Sample Input For Custom Testing

```
12
Ainslee Ginsie Galaxy 28
Libbey Apdell Starbucks 44
Illa Stebbings Berkshire 49
Laina Sycamore Berkshire 20
Abbe Parnell Amazon 20
Ludovika Reveley Berkshire 30
Rene Antos Galaxy 44
Vinson Beckenham Berkshire 45
Reed Lynock Amazon 41
Wyndham Bamfield Berkshire 34
Loraine Sappson Amazon 49
Abbe Antonutti Starbucks 47
```

Sample Output

```
The average age for company Amazon is 37
The average age for company Berkshire is 36
The average age for company Galaxy is 36
The average age for company Starbucks is 46
The count of employees for company Amazon is 3
The count of employees for company Berkshire is 5
The count of employees for company Galaxy is 2
The count of employees for company Starbucks is 2
The oldest employee of company Amazon is Loraine Sappson having age 49
The oldest employee of company Berkshire is Illa Stebbings having age
49
The oldest employee of company Galaxy is Rene Antos having age 44
The oldest employee of company Starbucks is Abbe Antonutti having age
47
```

Explanation

Here, 12 employees are presented with their details (first name, last name, company name, and age). Then, based on that data, the 3 methods are called: *AverageAgeForEachCompany*, *CountOfEmployeesForEachCompany*, and *OldestAgeForEachCompany*. The result obtained from these 3 function calls is printed to the standard output.