# Company statistic

# *You are an employee of the Human Resources Department of a large company. Your boss constantly asks you for various statistics on the company's staff. To make your job easier, you decide to apply your knowledge of the “C # Advance” course and write a program that will make it easier for you to process employee data.*

## Preparation

Download the skeleton provided in Judge. **Do not** change the **StartUp** class or its **namespace**.

## Problem description

Your task is to create a repository, which stores Employees by creating the classes described below.

First, write a C# class **Employee** with the following properties:

* **Name: string**
* **Department: string**
* **Age: int**
* **Salary: decimal**

The class **constructor** should receive **name, department, age** and **salary** and override the **ToString()** method in the following format:

**"****{name} - {department} - {age} – {salary}"**

**Next**, write a C# class **Company** that has **employees** (a collection, which stores the entity **Employee**). All entities inside the repository have the **same properties**. Also, the Company class should have those properties:

* **Name: string**
* **MaxPersonel: int**

The class **constructor** should receive **name** and **maxPersonel**, also it should initialize the **data** with a new instance of the collection**.** Implement the following features:

* Field **data** – **collection** that holds added Employees.
* Method Add(Employee employee) – **adds** an **entity** to the data **if** **there** **is** an **empty slot** for the Employee.
* Method Remove(string name, string department) – removes the Employee by **given** name **and** department**,** if such **exists**, and **returns bool**.
* Method GetDepartmentOldestEmployee(string department) – returns the oldest Employee in given department.
* Method **GetDepartmentLessPaidEmployee(string** department**)** – returns the Employee with the **lowest salary** in **given** department.
* Getter Count – **returns** the **number** of Employees in the Company.
* **GetStatistics()** – **returns** a **string** in the following **format**:
  + **"****The employees in {Company Name}:  
    {Employee1}  
    {Employee1}  
    (…)**"

## Constraints

* The **combinations** of **Employee name** and **Employee department** will be **always unique**.
* The **age** of the Employee will **always** be in **range [18 – 60]**.
* The **maxPersonel** will always be in **range [1 – 30]**.
* The **salary** will **always** be positive number.

## Examples

This is an example how the **Company** class is **intended to be used**.

|  |
| --- |
| Sample code usage |
| // Initialize the repository  Company company = new Company("Nike Inc.", 5);  // Initialize entity  Employee firstEmployee = new Employee("Petar", "Finance", 29, 2000);  // Print Employee  Console.WriteLine(firstEmployee); // Petar - Finance - 29 - 2000  // Add Employee  company.Add(firstEmployee);  // Remove Employee  Console.WriteLine(company.Remove("Ivan", "Maintenance")); //False  Console.WriteLine(company.Remove("Petar", "Finance")); //True  Employee secondEmployee = new Employee("Ivan", "Maintenance", 33, 2000);  Employee thirdEmployee = new Employee("Ivan", "Finance", 30, 2100);  Employee fourthEmployee = new Employee("Georgi", "Finance", 29, 2000);  company.Add(secondEmployee);  company.Add(thirdEmployee);  company.Add(fourthEmployee);  // Get oldest employee in given department  Employee oldestEmployee = company.GetDepartmentOldestEmployee("Finance");  Console.WriteLine(oldestEmployee); // Ivan - Finance - 30 - 2100  // Get less paid employee in given department  Employee lessPaid = company.GetDepartmentLessPaidEmployee("Finance");  Console.WriteLine(lessPaid); // Georgi - Finance - 29 - 2000  // Count  Console.WriteLine(company.Count); // 3  // Get Statistics  Console.WriteLine(company.GetStatistics());  //The employees in Nike Inc.:  //Ivan - Maintenance - 33 - 2000  //Ivan - Finance - 30 - 2100  // Georgi - Finance - 29 - 2000 |

## Submission

Zip all the files in the project folder except **bin** and **obj** folders.