# Data Structures Exam: Organization

## Description

You need to implement a structure that will be used by organizations to track their employees. You need to support the following operations (and they should be **fast**):

* Add() – Add an employee to the organization. You will need to implement the Contains() method as well.
* Contains(Person) – Checks if an employee is present in the organization
* ContainsByName(string) – Checks if an employee with the given name exists in the organization
* Count – Returns the number of employees in the organization
* GetAtIndex(int) – Return the **N**-th employee that started working in the organization. The index is based on insertion in the organization structure. If such index is not present, throw IndexOutOfRangeException
* GetByName(string) – Returns all employees with the given name
* FirstByInsertOrder(int) – Returns the first **N**-th employees that started working in the organization. In java, there should be override without parameter that returns the first element only
* GetWithNameSize(int) – Returns all employees with the given name **length**. If there are no such employees, throw new ArgumentException
* SearchWithNameSize(int, int) – Returns all employees that have name lengths between the 2 parameters
* PeopleByInsertOrder() – Returns all employees by the order of which they started working in the organization

Duplicates of the person class **are allowed**. That means that the names and salaries of multiple objects might be the same.

## Input / Output

You are given a **Visual Studio C# project skeleton** (unfinished project) / **IntelliJ Java project** holding the interface IOrganization, the unfinished classes Organization and Person. **Tests** covering the Organization **functionality** and **performance**.

Your task is to **finish this class** to make the tests run correctly.

* You are **not allowed to change the tests**.
* You are **not allowed to change the interface**.
* You can add to the Person class, but don't remove anything.
* You can edit the Organization class if it implements the IOrganization interface.

## Interface

The interface IOrganization in C# looks like the code below:

|  |
| --- |
| public interface IOrganization : IEnumerable<Person>  {  int Count { get; }  bool Contains(Person person);  bool ContainsByName(string name);  void Add(Person person);  Person GetAtIndex(int index);  IEnumerable<Person> GetByName(string name);  IEnumerable<Person> FirstByInsertOrder(int count = 1);  IEnumerable<Person> SearchWithNameSize(int minLength, int maxLength);  IEnumerable<Person> GetWithNameSize(int length);  IEnumerable<Person> PeopleByInsertOrder();  } |

The interface Organization in Java looks like the code below:

|  |
| --- |
| **public interface** Organization **extends** Iterable<Person>{   **int** getCount();  **boolean** contains(Person person);  **boolean** containsByName(String name);  **void** add(Person person);  Person getAtIndex(**int** index);   Iterable<Person> getByName(String name);  Iterable<Person> firstByInsertOrder();  Iterable<Person> firstByInsertOrder(**int** count);  Iterable<Person> searchWithNameSize(**int** minLength, **int** maxLength);  Iterable<Person> getWithNameSize(**int** length);  Iterable<Person> peopleByInsertOrder(); } |

## Submission

Submit an archive (.zip) of the source code. Your code **mustn't** contain namespaces/packages.

## Scoring

Each implemented method brings you a specific amount of points, some of the points are awarded for correct behavior, others for performance. **The performance tests might not work on your PC**. You need to cover all tests in each group to receive points. Bellow is a breakdown of all points by methods:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Correct Behavior | Performance | Total |
| Overall | 35 | 65 | 100 |