**Задание 1 (классы, объекты и методы)**

1. Необходимо создать классы для пользователей двух видов: кандидат {Candidate} – тот, кто находится в поиске работы (идентификатор Guid {id}, имя {FirstName}, фамилия {LastName}, желаемая должность {JobTitle}, краткое описание должности {JobDescription}, желаемая заработная плата {JobSalary}, причина увольнения с прошлого места работы {DismissalReason}); работник {Employee} – тот, кто занимает какую-то определённую должность в какой-либо компании (идентификатор Giud {id}, имя {FirstName}, фамилия {LastName}, должность {JobTitle}, заработная плата {JobSalary}, краткое описание должности {JobDescription}, компания {Company}, её название {Company.Name}, страна {Company.Country}, город {Company.City}, физический адрес {Company.Address}).

Причина увольнения для кандидата должна задаваться перечислением со следующими значениями:

|  |  |
| --- | --- |
| FamilyReasons | семейная ситуация |
| ProfessionalGrowthLack | отсутствие профессионального роста |
| LowSalary | низкая заработная плата |
| BadTeamMicroclimate | плохой микроклимат в коллективе |
| LackManagementUnderstanding | отсутствие взаимопонимания с руководством |
| Other | другие |

При этом для кандидатов, у которых не было предыдущего места работы, причина увольнения будет равна значению **null**.

Для любого пользователя создать вычисляемое свойство (доступное только для чтения) с полным именем пользователя {FullName}: "{FirstName} {LastName}"

2. Создать и реализовать интерфейс IConsoleOutput, который будет содержать метод, выводящий в консоли данные о пользователе.

***Пример:***

Hello, I am {FullName}.

I want to be a {JobTitle} ({JobDescription}) with a salary from {JobSalary}.

I quit my previous job for a reason of {DismissalReason}.

***Или***

Hello, I am {FullName}.

I want to be a {JobTitle} ({JobDescription}) with a salary from {JobSalary}.

I haven't worked anywhere before.

***Или***

Hello, I am {FullName}, {JobTitle} in {Company.Name} ({Company.Country}, {Company.City}, {Company.Address}) and my salary {JobSalary}.

3. Реализовать создание случайного количества кандидатов и сотрудников.

\*4. Заполнить все данные о пользователях и компаниях используя библиотеку Bogus.

<https://github.com/bchavez/Bogus>

\*5. Создать класс UserFactory, который в зависимости от типа пользователя (Candidate, Employee) будет возвращать необходимый объект.

\*6. Создать интерфейс IReportGenerator.

Добавить класс EmployeeReportGenerator, который отсортирует всех работников по возрастанию названия компании и по убыванию заработной платы.

***Вывод:***

{User.Id} | {Company.Name} | {User.FullName} | {User.Salary}

Добавить класс CandidateReportGenerator, который отсортирует всех кандидатов по позиции и по возрастанию заработной платы.

***Вывод:***

{User.Id} | {User.JobTitle} | {User.FullName} | {User.Salary}