

Тестовое задание для младшего разработчика

C#

Необходимо создать консольное приложение, обрабатывающее текстовый файл, содержащий список сотрудников в формате JSON.

Формат записи о сотруднике:

- столбец Id, тип в C# - int;
- столбец FirstName, тип в C# - string;
- столбец LastName, тип в C# - string;
- столбец SalaryPerHour, тип в C# - decimal.

Приложение принимает входные аргументы (в string[] args метода Main), и на их основе выполняет соответствующую операцию.

Доступны следующие аргументы и операции:

- **-add FirstName:John LastName:Doe Salary:100.50**

Добавляет в файл новую запись. Поля FirstName, LastName и SalaryPerHour заполняются из аргументов (John, Doe, 100.50). Поле Id генерируется автоматически по следующему принципу: самое большое значение столбца Id, из всех имеющихся в файле, + 1.

- **-update Id:123 FirstName:James**

Обновляет запись с Id=123, меняет в нем поле FirstName на указанное (James). Таким образом можно обновлять любые поля, кроме Id. Если не существует записи с таким Id, в консоль выводится строка, сообщающая об ошибке (текст ошибки - на усмотрение разработчика).

- **-get Id:123**

Выводит в консоль строку формата «Id = {Id}, FirstName = {FirstName}, LastName = {LastName}, SalaryPerHour = {SalaryPerHour}», вместо {Id}, {FirstName}, {LastName}, {SalaryPerHour} должны быть подставлены соответствующие поля из записи с Id=123 из файла. Если не существует записи с таким Id, в консоль выводится строка, сообщающая об ошибке (текст ошибки - на усмотрение разработчика).

- **-delete Id:123**

Удаляет запись с Id=123 из файла. Если не существует записи с таким Id, в консоль выводится строка, сообщающая об ошибке (текст ошибки - на усмотрение разработчика).

- **-getall**

Возвращает список всех сотрудников (формат аналогичен приведенному в описании аргумента - get).

Дополнительные условия:

- один из методов, на выбор разработчика, должен быть протестирован unit-тестами (любой тестовый фреймворк);
- наибольшее внимание следует уделить качеству коду.

Допускается использовать любые сторонние библиотеки (nuget-пакеты).

Для работы с JSON мы обычно используем Newtonsoft.JSON, но разработчик может использовать любое решение, которое сочтет подходящим.

