ПИС-3, ПОИТ

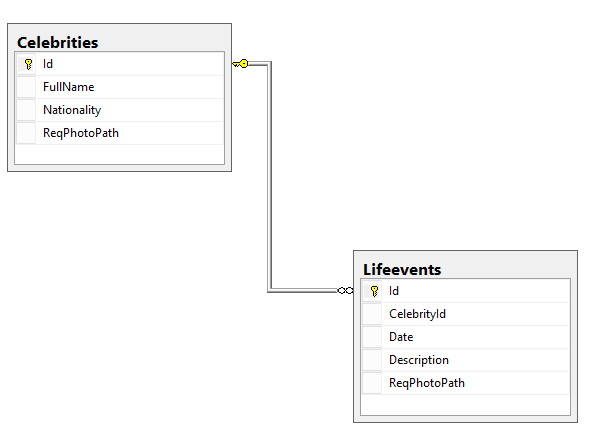
**Лабораторная работа 6**

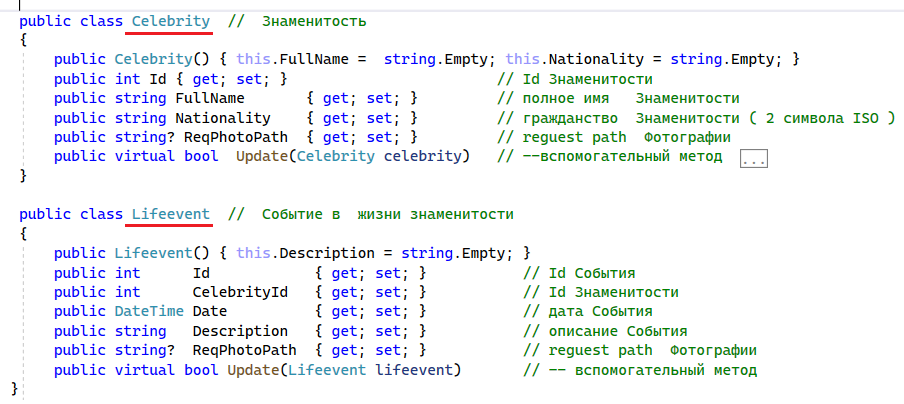
**ASPA** – приложение ASP.NET CORE

♣ – задания, требующие самостоятельного изучения (нет в лекциях)

**Задание 1.**

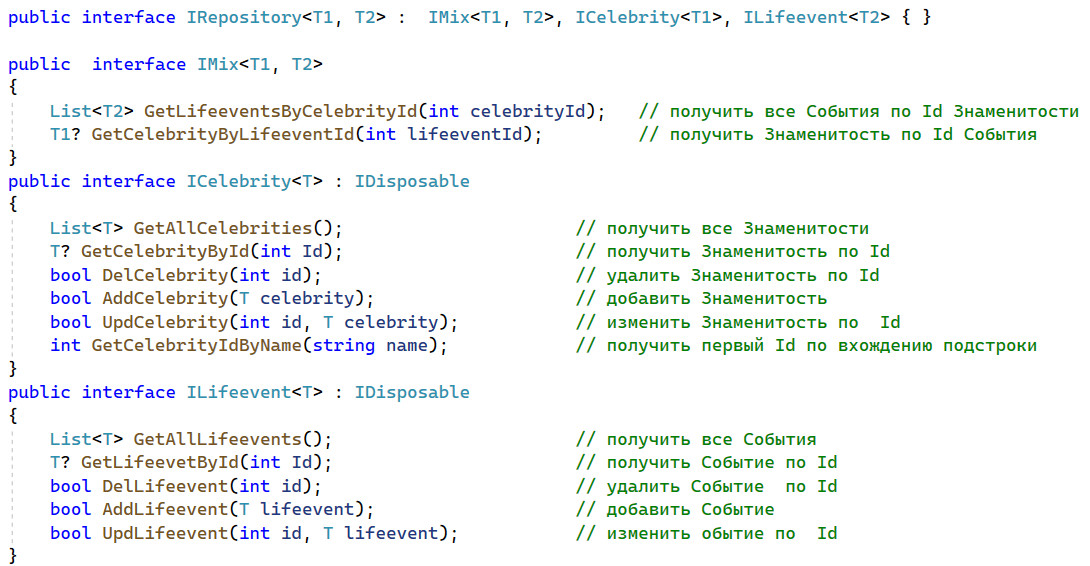
1. Исследуйте логическую схему базы данных «Life Events of Celebrities» (LES) и поясняющие назначение полей классы, соответствующие (ORM) таблицам LES.



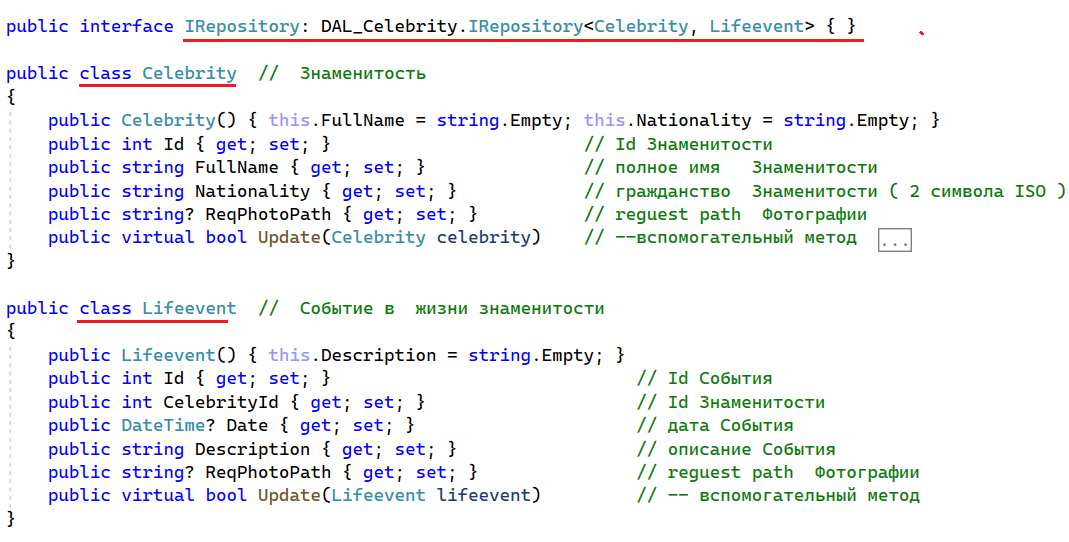


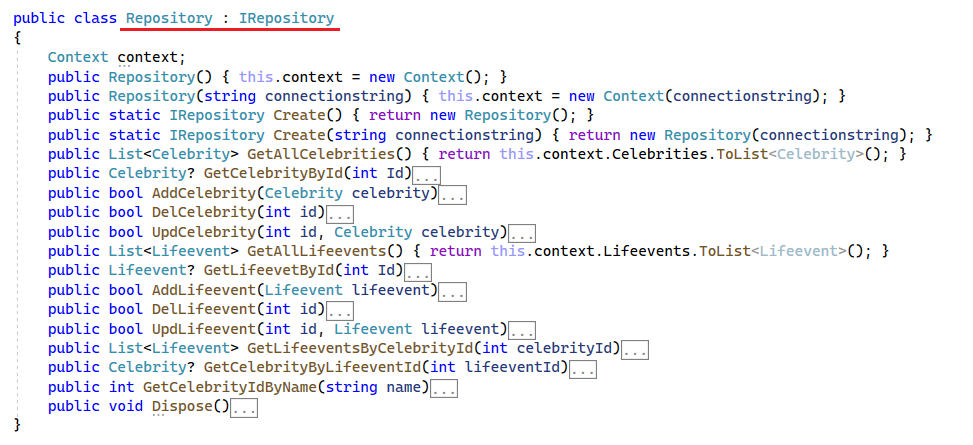
**Задание 2.**

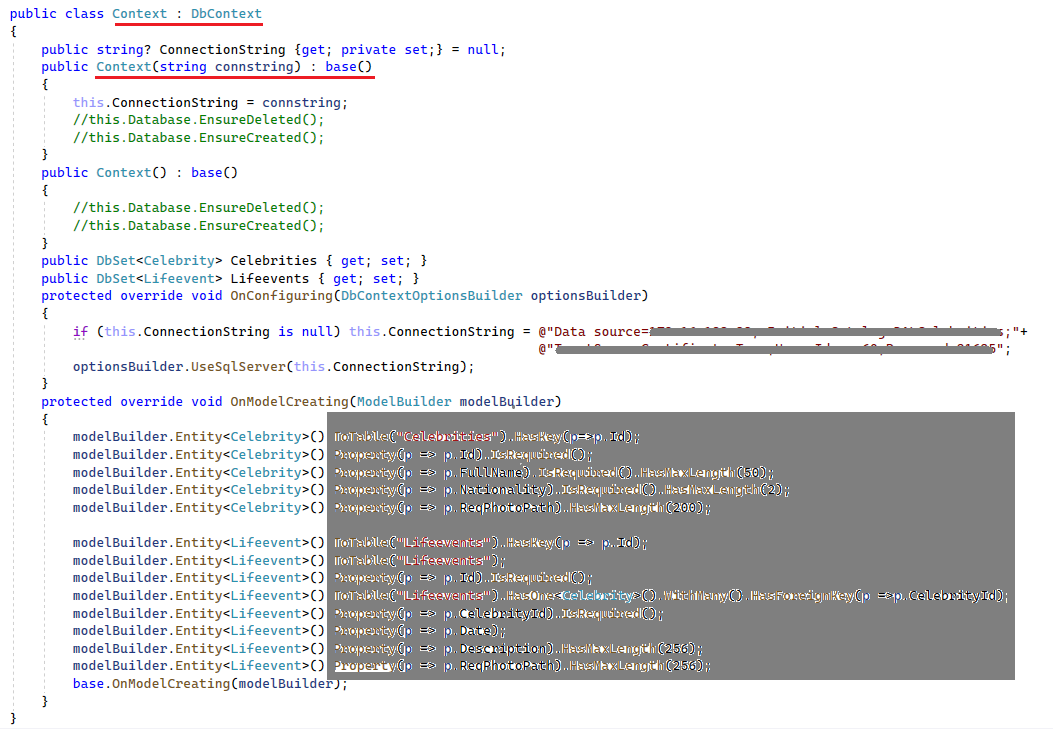
1. Разработайте библиотеку ***DAL\_Celebrity*** содержащую следующие интерфейсы (прилагается файл ***DAL\_Celebrity.cs***)



1. Разработайте библиотеку ***DAL\_Celebrity\_MSSQL*** реализующую интерфейс ***IRepository*** (***DAL\_Celebrity***) с применением Entity Framework (NuGet-пакет).

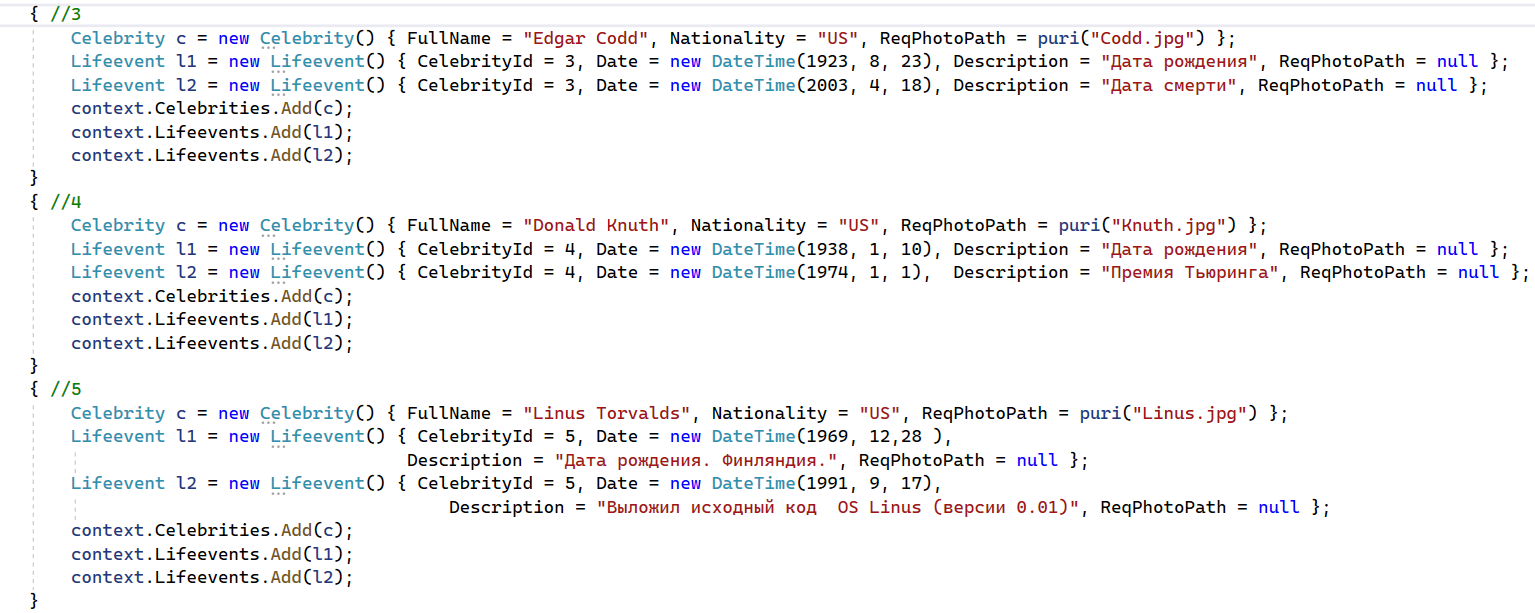


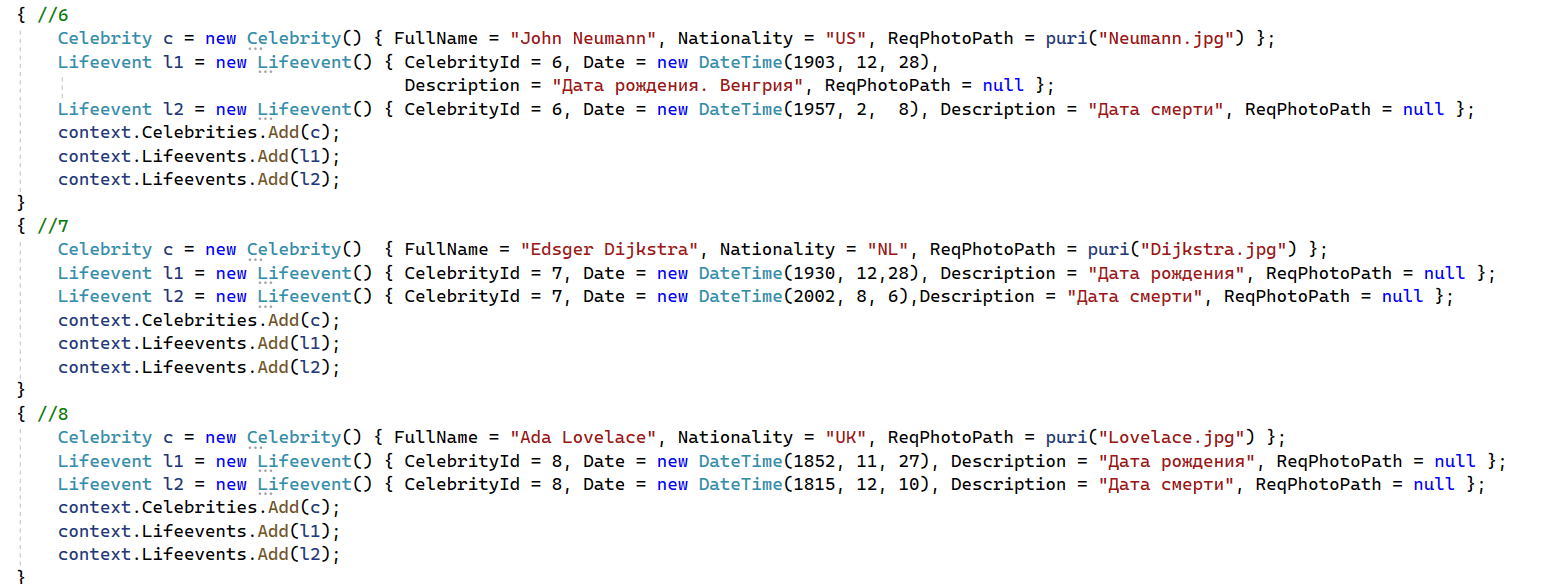


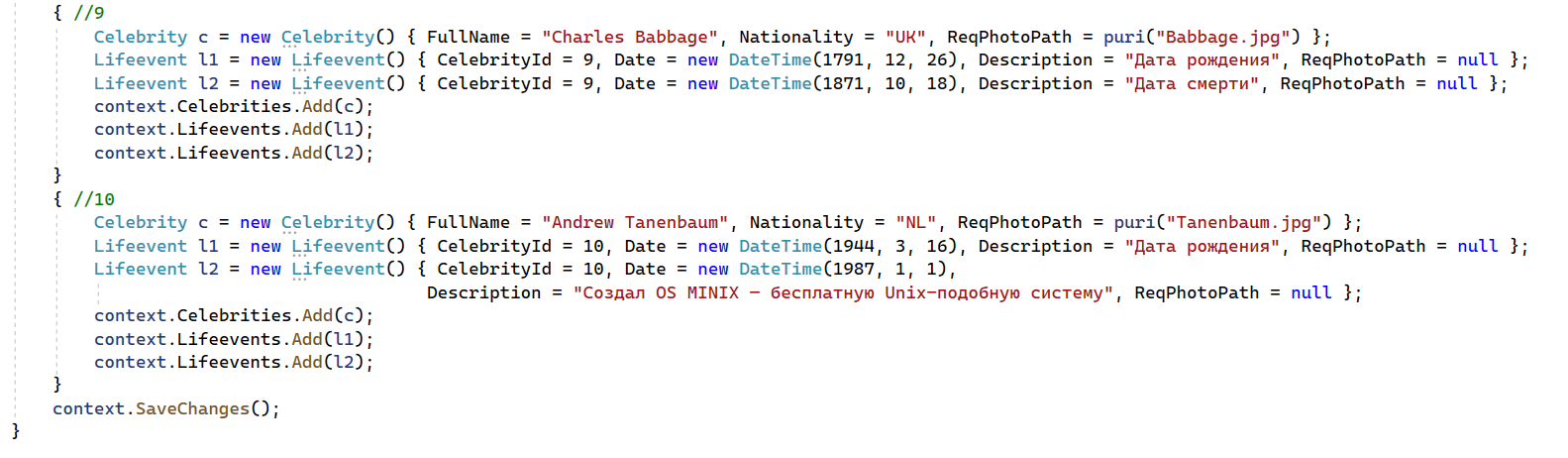


1. Добавьте в библиотеку ***DAL\_Celebrity\_MSSQL*** класс ***Init***, предназначенный для инициализации и тестирования БД (файл ***Init.cs*** прилагается).



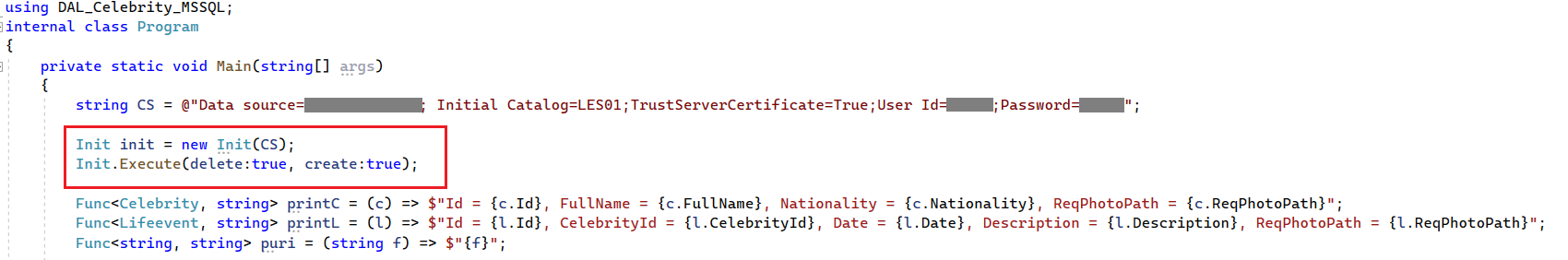






**Задание 3**

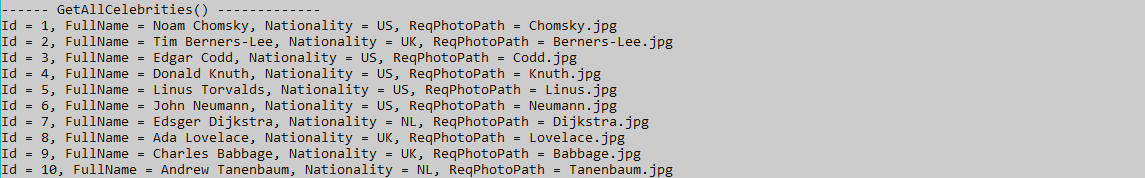
1. Разработайте консольное приложение ***DAL\_Celebrity\_MSSQL\_Test*** для тестирование репозитория(из ***DAL\_Celebrity\_MSSQL***).
2. Программный код теста (файл ***Programm.cs*** прилагается).

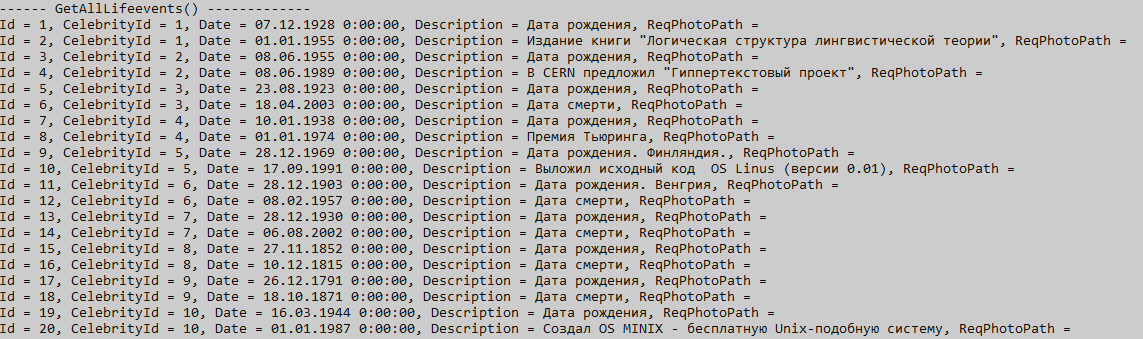


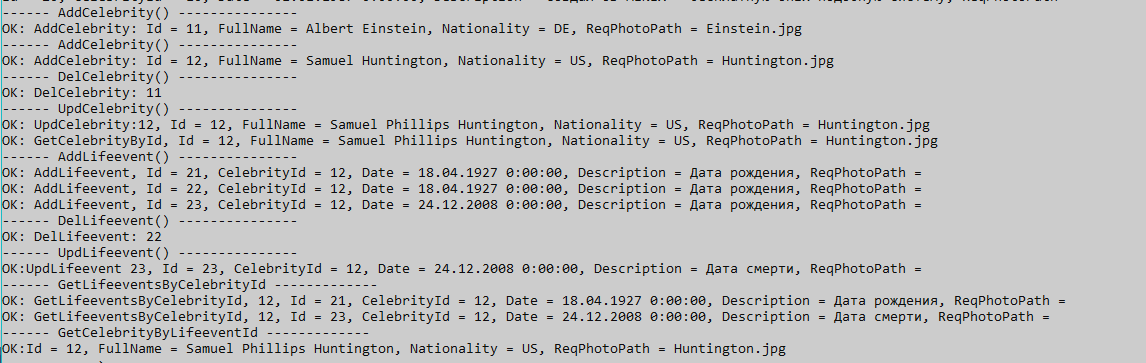




1. Результаты теста



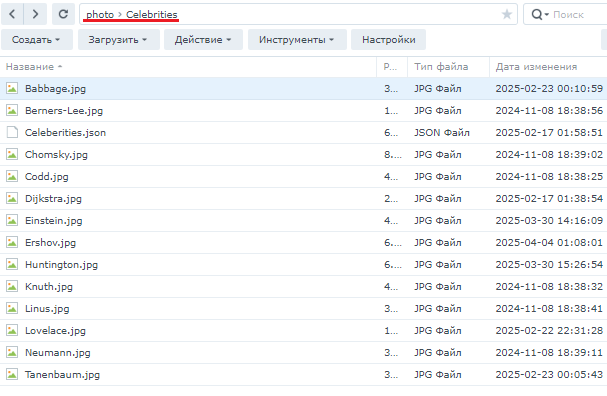




1. Проверьте с помощью теста работоспособность функций репозитория (***DAL\_Celebrity\_MSSQL***) и добейтесь успешного выполнения теста.

**Задание 4**

1. Фотографии, которые далее будут использоваться в приложениях, размещены ***diskstation*** (student/fitfit)



1. Разработайте ASPA, применив следующий шаблон.

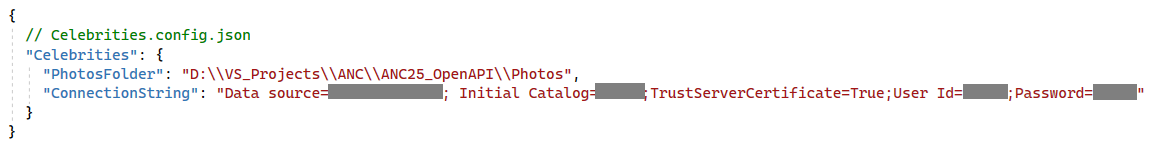
Имя решения: **ASPA**

Имя проекта: **ASPA006\_1**

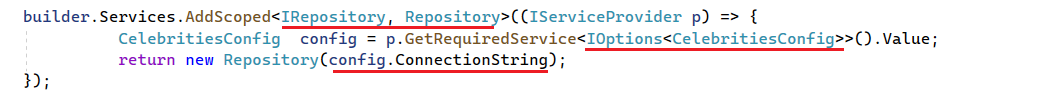
Версия .NET: **7.0** или **8.0**

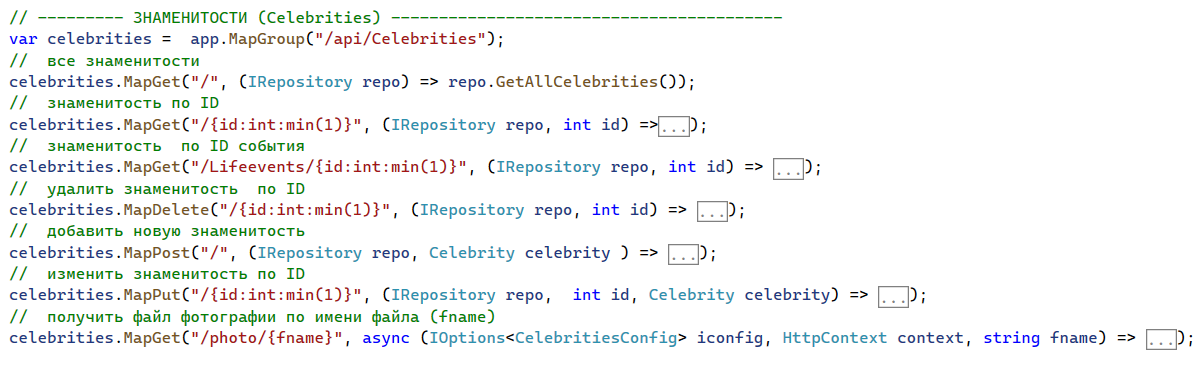


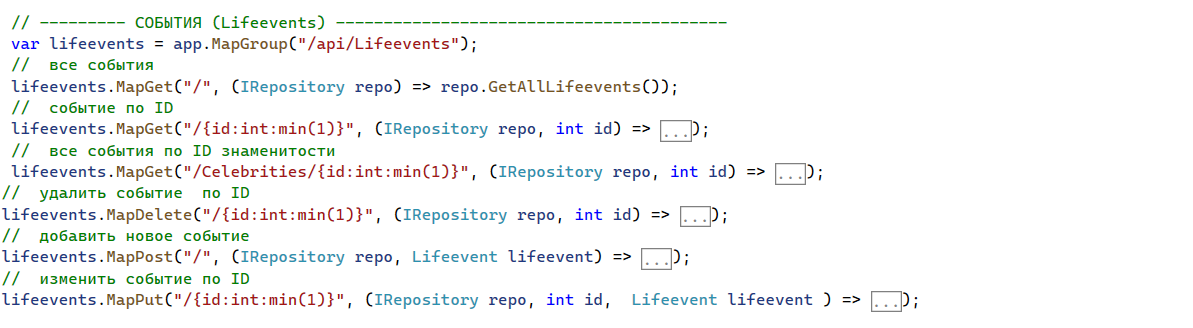
1. Подключите библиотеку ***DAL\_Celebrity\_MSSQL*** к **ASPA006\_1.**
2. Создайте конфигурационный файл ***Celebrities.config.json***, позволяющий параметризировать директорий для фотографий и строку подключения к серверу СУБД.



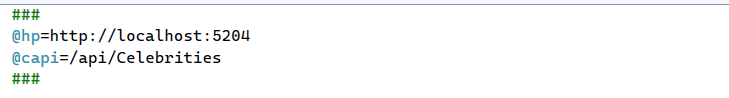
1. Обеспечьте доступ к конфигурации (строка подключения и папка для фотографий) с помощью объекта класса ***CelebritiesConfig*** интерфейса***.***

1. Создайте Soped-сервис, реализующий интерфейс ***IRepository*** (задание 2), как это сделано в следующем примере. 
2. Разработайте следующие конечные точки использующие сервис, реализующий ***IRepository***:





1. Для обработки ошибок в конечных точках разработайте middleware-обработчик ошибок.
2. Тест (файл ***app.http,*** прилагается) для конечных точек.

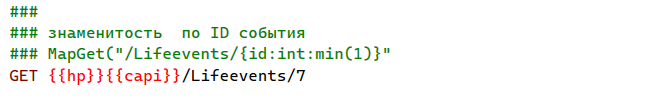
****

****

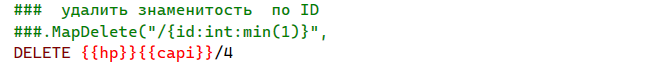
****

****

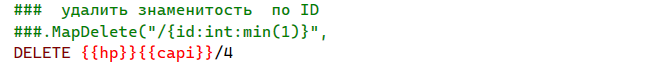
****

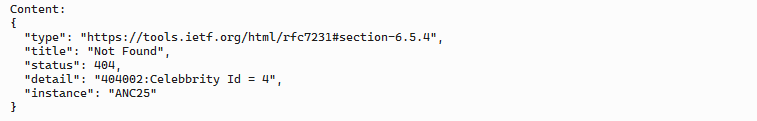
****

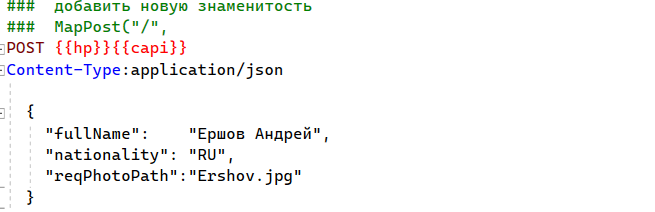
****

****

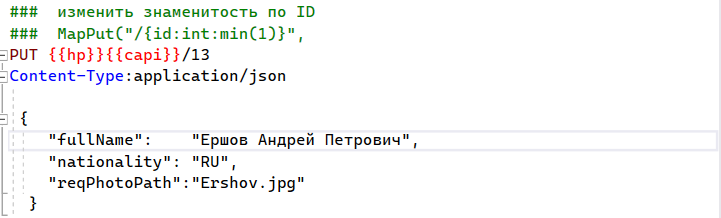
****

****

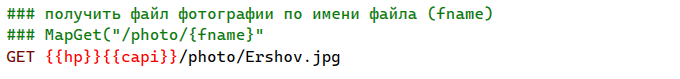
****

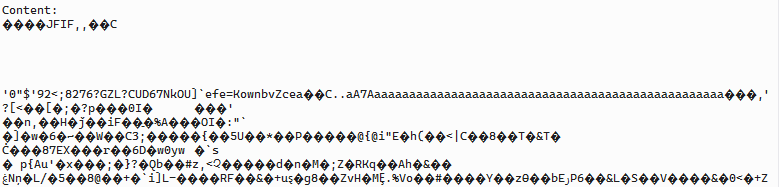
****

****

****

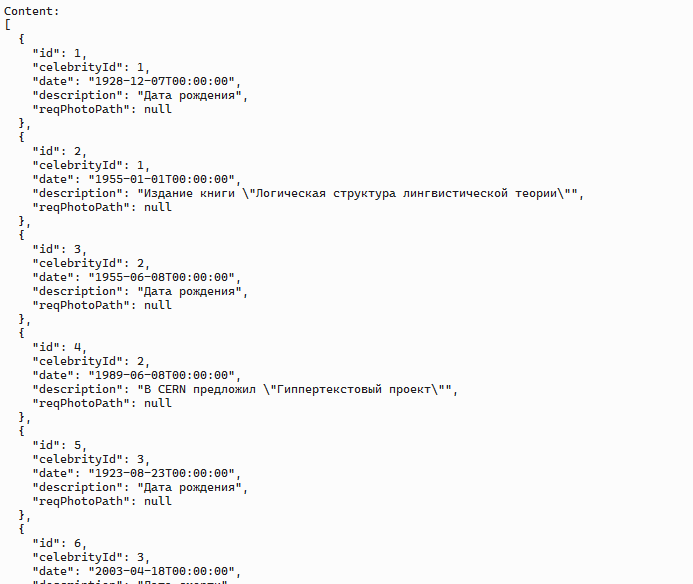
****

****

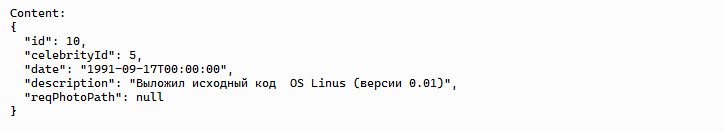
****

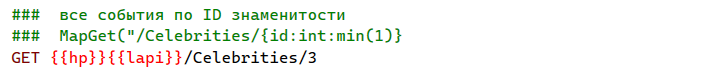
****

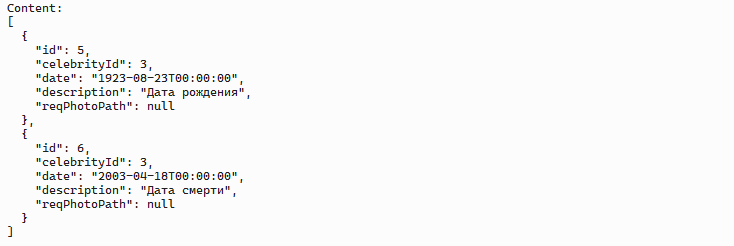
****

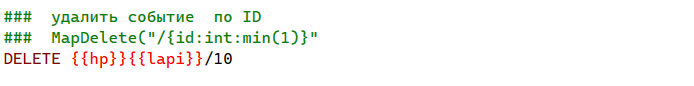
****

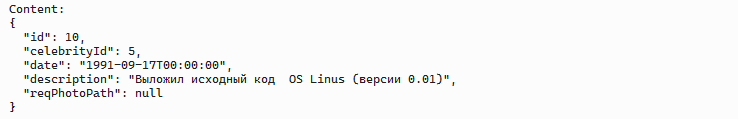
****

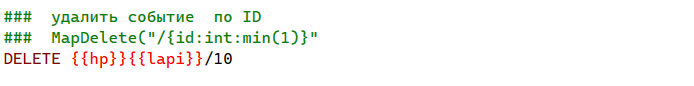
****

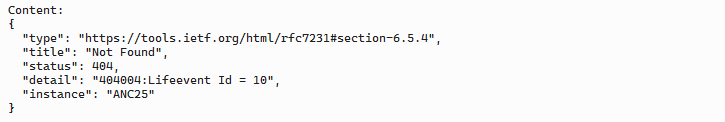
****

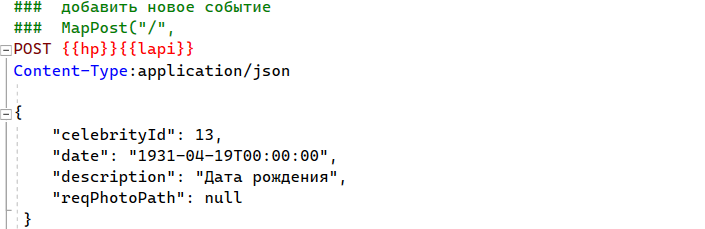
****

****

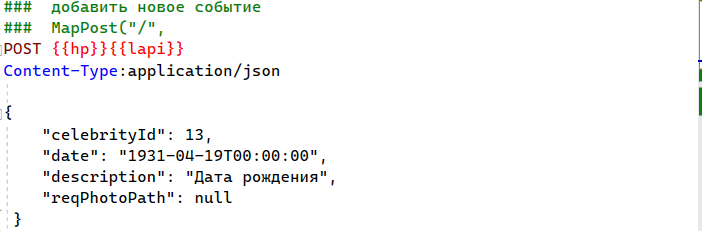
****

****

****

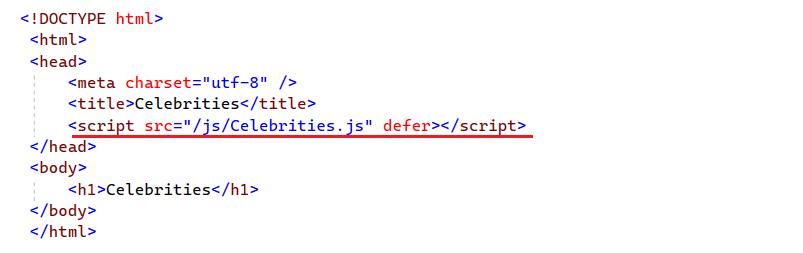
****

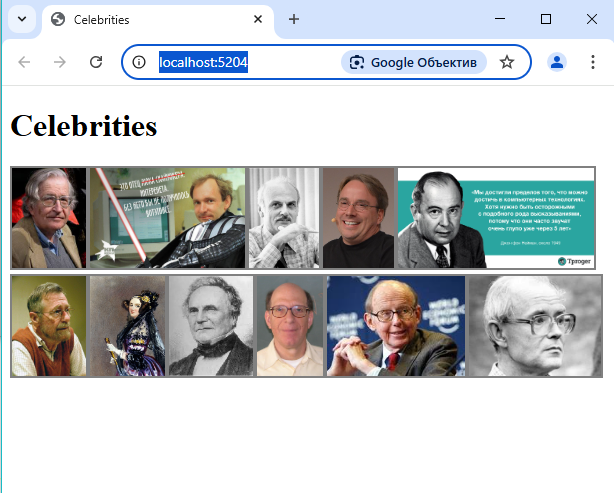
****

****

****

1. Разработайте html-страницу ***Index.html*** следующего вида. Обеспечьте отображение этой страницы при запуске приложения по умолчанию. Ниже приводится внешний вид страницы при запуске приложения.

****

****

1. При одинарном клике мышью фотографии, внизу отображается, строки с данными о событиях.

