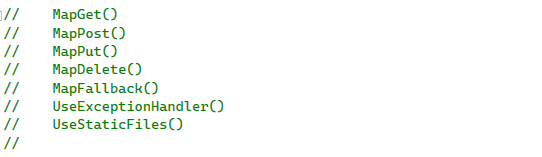
**Задание 5.Ответьте на следующие вопросы**

1. Поясните назначение следующих функций



**1. MapGet()**

* **Назначение:** Используется для определения маршрута HTTP GET. Этот метод обрабатывает запросы, которые запрашивают данные с сервера. Например, он может использоваться для получения списка ресурсов или конкретного ресурса по его идентификатору.

**2. MapPost()**

* **Назначение:** Используется для определения маршрута HTTP POST. Этот метод обрабатывает запросы, которые отправляют данные на сервер. Обычно используется для создания новых ресурсов или отправки данных, таких как формы.

**3. MapPut()**

* **Назначение:** Используется для определения маршрута HTTP PUT. Этот метод обрабатывает запросы, которые обновляют существующие ресурсы на сервере. PUT обычно используется для замены всего представления ресурса.

**4. MapDelete()**

* **Назначение:** Используется для определения маршрута HTTP DELETE. Этот метод обрабатывает запросы, которые удаляют ресурс на сервере. Например, он может использоваться для удаления записи из базы данных.

**5. MapFallback()**

* **Назначение:** Используется для определения маршрута, который будет срабатывать, если ни один из других маршрутов не подходит. Это полезно для обработки ошибок 404 и предоставления пользовательских страниц ошибок или перенаправлений.

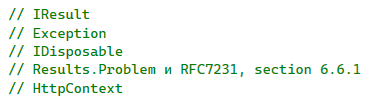
**6. UseExceptionHandler()**

* **Назначение:** Позволяет настраивать обработчик исключений для приложения. Этот метод перехватывает необработанные исключения и перенаправляет их на указанный маршрут, где можно обработать и отобразить ошибку пользователю, например, показывая страницу с сообщением об ошибке.

**7. UseStaticFiles()**

* **Назначение:** Включает поддержку статических файлов в веб-приложении. Это позволяет серверу обслуживать файлы, такие как CSS, JavaScript и изображения, находящиеся в определенных папках, например, wwwroot.

1. Поясните назначение следующих объектов и интерфейсов



**1. IResult**

* **Назначение:** Это интерфейс, который представляет результат обработки HTTP-запроса. Он используется для абстракции различных типов ответов, которые может отправить сервер. Поддерживает множество реализаций, таких как Results.Ok(), Results.NotFound(), Results.BadRequest() и т.д., что позволяет легко формировать и возвращать ответы в удобном формате.

**2. Exception**

* **Назначение:** Это базовый класс для всех исключений в .NET. Он предоставляет механизм для обработки ошибок и исключительных ситуаций в приложении. Когда возникает ошибка, можно выбросить исключение, чтобы сигнализировать о проблеме, и затем перехватить его для обработки, например, с помощью try-catch блоков или обработчиков ошибок.

**3. IDisposable**

* **Назначение:** Это интерфейс, который определяет метод Dispose(), предназначенный для освобождения ресурсов, занимаемых объектом. Он используется для управления неуправляемыми ресурсами (например, файловыми дескрипторами, соединениями с базами данных) и позволяет явно освобождать ресурсы, когда они больше не нужны, предотвращая утечки памяти.

**4. Results.Problem**

* **Назначение:** Это метод, который возвращает стандартный ответ с проблемой (ошибкой) в формате RFC 7807 (Problem Details for HTTP APIs). Он создает ответ с кодом состояния 500 (Internal Server Error) или другим статусом и включает информацию о проблеме, что позволяет клиентам лучше понять, что пошло не так. Этот подход улучшает обработку ошибок в API.

**5. HttpContext**

* **Назначение:** Это объект, который содержит всю информацию о текущем HTTP-запросе и ответе. Он предоставляет доступ к таким данным, как запросы, ответы, сеансы, пользовательские данные и многое другое. HttpContext позволяет разработчикам обрабатывать запросы и управлять состоянием приложения на уровне запроса.

1. ♣Чем известны Ada Lovelace и Charles Babbage?☺