**15. Назначение паттерна Builder. В ASP.NET применение паттерна Builder.**  
Паттерн Builder (Строитель) относится к порождающим шаблонам проектирования. Он позволяет пошагово создавать сложный объект, отделяя процесс его конструирования от представления конечного результата. Это особенно полезно, когда объект создаётся из множества взаимосвязанных частей или когда требуется обеспечить разные варианты конфигурации одного и того же объекта.  
В ASP.NET (и ASP.NET Core) этот паттерн применяется, например, при конфигурации и создании хоста приложения. Методы вроде WebHost.CreateDefaultBuilder, Host.CreateDefaultBuilder и интерфейс IApplicationBuilder используют концепцию построения – вы поэтапно добавляете необходимые компоненты (middleware, сервисы, настройки) для формирования конечного приложения.

**16. Назначение паттерна Chain of Responsibility. Применение в ASP.NET.**  
Паттерн Chain of Responsibility (Цепочка обязанностей) позволяет передавать запрос по цепочке обработчиков, где каждый обработчик либо обрабатывает запрос, либо передаёт его дальше. Это снижает связанность кода, поскольку отправитель запроса не знает, кто его в итоге обработает.  
В ASP.NET Core этот паттерн лежит в основе системы middleware: HTTP-запрос проходит через цепочку компонентов (middleware), каждый из которых может выполнить какую-то логику (аутентификация, логирование, обработка ошибок и т.д.) и передать управление следующему компоненту в цепочке.

**17. Понятие Middleware.**  
Middleware – это программные компоненты (средства промежуточной обработки), которые формируют конвейер (pipeline) обработки HTTP-запросов в ASP.NET Core. Каждый middleware может выполнять некоторую логику до и после вызова следующего компонента в цепочке. Это позволяет реализовывать такие функции, как аутентификация, обработка ошибок, логирование, маршрутизация, кэширование и многое другое.

**18. Понятие статического файла.**  
Статический файл – это ресурс, который хранится на сервере и передаётся клиенту без дополнительной динамической обработки. К таким файлам относятся HTML, CSS, JavaScript, изображения, видео и другие файлы, которые можно отправить напрямую в ответ на запрос, не генерируя их программно.

**19. Отличительный признак наименования подключаемого Middleware-компонента.**  
По общепринятой конвенции, метод-расширение для подключения middleware-компонента начинается со слова **Use** (например, UseStaticFiles, UseRouting, UseAuthentication). Такое именование подчёркивает, что компонент «подключается» к конвейеру обработки запросов.

**20. Известные встроенные Middleware-компоненты.**  
Среди встроенных middleware в ASP.NET Core можно выделить:

* **Static Files Middleware** (UseStaticFiles) – для обслуживания статических файлов.
* **Routing Middleware** (UseRouting и UseEndpoints) – для маршрутизации запросов.
* **Authentication и Authorization Middleware** (UseAuthentication, UseAuthorization) – для аутентификации и авторизации.
* **Exception Handling Middleware** (UseDeveloperExceptionPage, UseExceptionHandler) – для обработки ошибок.
* **HTTPS Redirection Middleware** (UseHttpsRedirection) – для перенаправления запросов на HTTPS.
* **CORS Middleware** (UseCors) – для обработки политики кросс-доменных запросов.

**21. Имя директории для статических файлов по умолчанию в ASP.NET.**  
По умолчанию в ASP.NET Core статические файлы размещаются в папке **wwwroot**. Именно эта папка используется сервером для обслуживания файлов без дополнительной обработки.

**22. Кто такой Джон фон Нейман?**  
Джон фон Нейман – выдающийся математик, физик и один из основателей теории вычислительных машин. Он внёс огромный вклад в развитие квантовой механики, теории игр, статистики и, самое главное, в разработку архитектуры современных компьютеров (так называемая «фон Неймановская архитектура»). Его идеи легли в основу создания программируемых электронных вычислительных машин.



