**1. Для чего используют многоуровневые архитектуры?**

Для создания приложений, которые являются модульными, легко расширяемыми, поддерживают принцип единственности.

**2. Опишите назначение слоев многоуровневой архитектуре. Business layer**

**(уровень бизнес-логики), Data Access layer (уровень доступа к данным).**

Уровень бизнес-логики обычно содержит код, который управляет бизнес-логикой приложения. Он может содержать правила бизнес-логики, сервисы и другие классы, которые управляют поведением приложения. Уровень доступа к данным (Data Access layer) содержит код, который обеспечивает доступ к данным, таким как запросы к базе данных, чтение и запись данных, и т. д.

**3. Поясните назначение и принцип использования паттерна Repository?**

Шаблон проектирования Repository позволяет скрыть детали работы с базой данных, предоставляя удобный интерфейс для взаимодействия с данными. В основе шаблона лежит идея разделения ответственности: объекты бизнес-логики не должны иметь прямого доступа к базе данных, а должны взаимодействовать только с объектами репозитория. Репозиторий, в свою очередь, должен реализовывать все необходимые операции для работы с данными: чтение, запись, удаление и т.д.

**4. Поясните назначение и принцип использования паттерна Unit of Work?**

Шаблон Unit of Work (единица работы) предназначен для управления транзакциями и сохранения изменений в базе данных. Он позволяет группировать несколько операций в одну транзакцию и сохранять все изменения в базе данных только после успешного завершения всех операций. Также шаблон Unit of Work позволяет отслеживать изменения объектов и сохранять их в базу данных при необходимости.

Обычно эти два шаблона используются вместе: репозитории предоставляют удобный интерфейс для взаимодействия с данными, а Unit of Work управляет транзакциями и сохранением изменений в базе данных. Это позволяет сделать код более гибким, удобным и легко поддерживаемым.

**5. Опишите основные особенности библиотеки Entity Framework?**

Библиотека Entity Framework - это ORM-фреймворк для работы с базами данных. Она предоставляет удобный способ работы с базами данных, позволяет разработчикам работать с данными как с объектами в памяти, а не с SQL-запросами, и предоставляет удобный API для работы с различными типами баз данных.

**6. Какие преимущества обеспечивает использование слабосвязанного**

**кода?**

Использование слабосвязанного кода может обеспечить лучшую гибкость и возможность повторного использования кода. Когда код слабосвязан, его можно легко заменить или модифицировать без влияния на другие части системы. Это может упростить тестирование и развертывание, а также сделать код более модульным и легко читаемым.