1. Для чего используют многоуровневые архитектуры?

Многоуровневые архитектуры используются для разделения функциональности системы на логические уровни, каждый из которых выполняет свои задачи. Это позволяет создавать более гибкие и масштабируемые системы, которые могут быть изменены или расширены без необходимости изменения всей системы.

1. Опишите назначение слоев многоуровневой архитектуре. Business layer (уровень бизнес-логики), Data Access layer (уровень доступа к данным).

Уровень бизнес-логики (Business layer) отвечает за реализацию бизнес-логики системы, включая обработку данных и реализацию бизнес-правил. Уровень доступа к данным (Data Access layer) отвечает за взаимодействие с базой данных и предоставление доступа к данным системы.

1. Поясните назначение и принцип использования паттерна Repository?

Паттерн Repository используется для управления доступом к данным в системе. Он позволяет скрыть детали реализации доступа к данным, а также обеспечивает единый интерфейс для доступа к данным в различных источниках данных. Это повышает гибкость и масштабируемость системы, так как ее компоненты могут использовать различные источники данных, не завися от конкретной реализации.

1. Поясните назначение и принцип использования паттерна Unit of Work?

Паттерн Unit of Work используется для управления транзакциями в системе. Он позволяет группировать операции с данными в рамках одной транзакции, обеспечивая атомарность и консистентность изменений. Это позволяет управлять транзакциями в системе более эффективно и обеспечивает защиту от непредвиденных сбоев.

1. Опишите основные особенности библиотеки Entity Framework?

Библиотека Entity Framework - это ORM (Object-Relational Mapping) для .NET Framework, которая позволяет работать с базой данных, используя объектно-ориентированный подход. Она обеспечивает автоматическое создание и управление схемой базы данных, а также предоставляет возможность создания запросов на языке LINQ. Кроме того, Entity Framework позволяет использовать различные источники данных, такие как SQL Server, MySQL и др.

1. Какие преимущества обеспечивает использование слабосвязанного кода?

Использование слабосвязанного кода позволяет создавать более гибкие и масштабируемые системы. Слабая связность означает, что компоненты системы имеют меньшую зависимость друг от друга, что позволяет изменять одну часть системы без влияния на другие части. Это также повышает возможности повторного использования кода, так как компоненты могут быть легко заменены или перенесены.