LESSON 1

1. ONION (HEXAGONAL) ARCHITECTURE

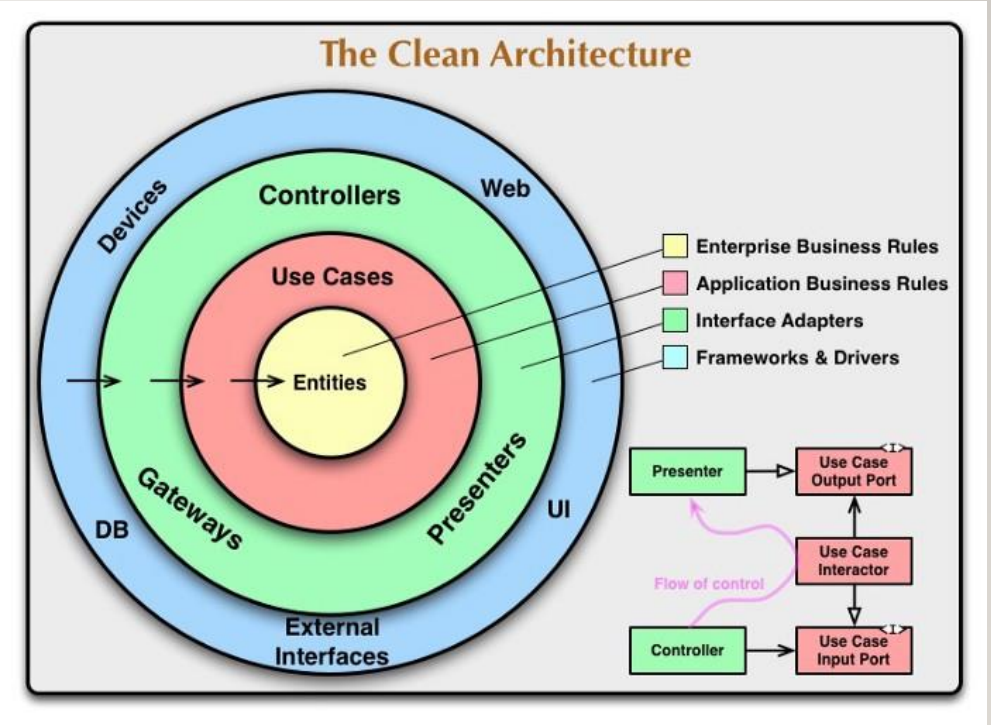
1. ONION (HEXAGONAL) ARCHITECTURE

- Clean Architecture
- DDD
- Hexagonal Architecture
- Layers of the Onion Architecture

CLEAN ARCHITECTURE

Чистая архитектура — это философия дизайна программного обеспечения, которая разделяет компоненты на определенные уровни.

Если компонент зависит от всех других компонентов, мы не знаем, какие побочные эффекты будет иметь изменение одного компонента, что затрудняет поддержку кодовой базы и еще более затрудняет ее расширение и декомпозицию.



DDD

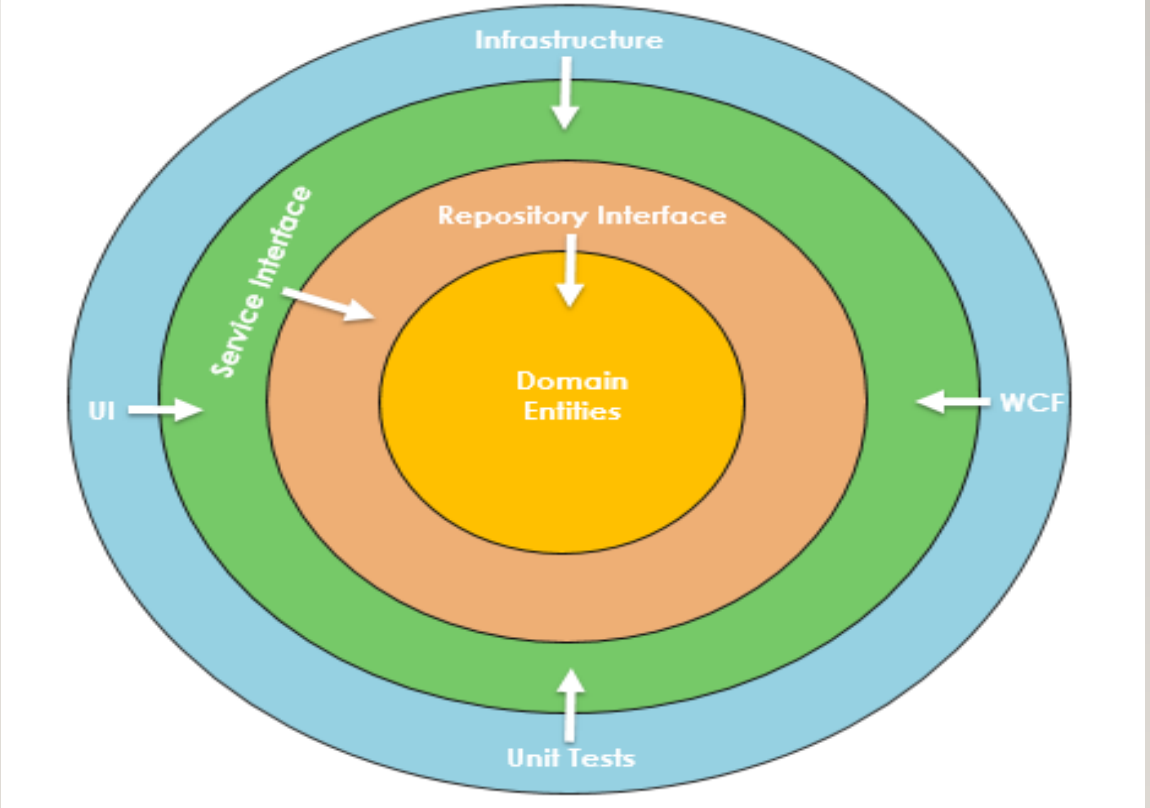
Domain Drive Design(предметно-ориентированное проектирование) - подход к разработке приложений, основанный на выделении доменов (domain).

Использование подхода «Domain driven design» при проектировании архитектуры сложных корпоративных информационных систем позволяет сделать проще поддержание изменений в проекте и эффективнее тестировать новые релизы.

HEXAGONAL ARCHITECTURE

Шестиугольная архитектура — это модель разработки программных приложений на основе доменной логики , чтобы изолировать ее от внешних факторов.

Логика предметной области указана в бизнес-ядре, которое мы назовем внутренней частью, а остальные — внешними частями. Доступ к логике домена извне возможен через порты и адаптеры .



LAYERS OF THE ONION ARCHITECTURE

Слой домена - в центральной части луковой архитектуры существует доменный уровень, этот слой представляет объекты бизнеса. Идея состоит в том, чтобы иметь все объекты домена в этом ядре. Он содержит все объекты домена приложения. Помимо объектов домена, у вас также могут быть интерфейсы домена. Эти объекты домена не имеют никаких зависимостей.

LAYERS OF THE ONION ARCHITECTURE

Слой репозиториев - этот уровень создает абстракцию между объектами предметной области и бизнес-логикой приложения. На этом уровне мы обычно добавляем интерфейсы, которые обеспечивают поведение сохранения и извлечения объектов, как правило, с использованием базы данных. Этот уровень состоит из шаблона доступа к данным, который представляет собой более слабо связанный подход к доступу к данным.

LAYERS OF THE ONION ARCHITECTURE

Слой сервисов - Уровень сервиса содержит интерфейсы с общими операциями, такими как «Добавить», «Сохранить», «Редактировать» и «Удалить». Кроме того, этот уровень используется для связи между слоем пользовательского интерфейса и слоем репозиториев. Уровень сервисов также может содержать бизнес-логику для сущности. На этом уровне интерфейсы служб отделены от их реализации, учитывая слабую связь и разделение задач.

LAYERS OF THE ONION ARCHITECTURE

Слой пользовательского интерфейса - Это самый внешний уровень, в котором хранятся периферийные функции, такие как пользовательский интерфейс и тесты. Для веб-приложения он представляет проект веб-API или модульного тестирования. На этом уровне реализован принцип внедрения зависимостей, так что приложение создает слабосвязанную структуру и может взаимодействовать с внутренним уровнем через интерфейсы.