Конспект Архитектуры ПО. Ulbi TV

Определения

Модули представляют собой отдельные части кода, которые выполняют определенную функцию или содержат определенный набор данных.

Хаотичные связи между модулями с ростом проекта усложняют поддержку, рефакторинг и добавление новых фичей в проект.

Модули должны обладать сильной связанностью, т.е. направлены на решение одной четкой задачи.

Слабой зацепленностью, т.е. быть как можно менее зависимыми от других модулей.

Архитектура ПО – это

- набор модулей и компонентов операционной системы

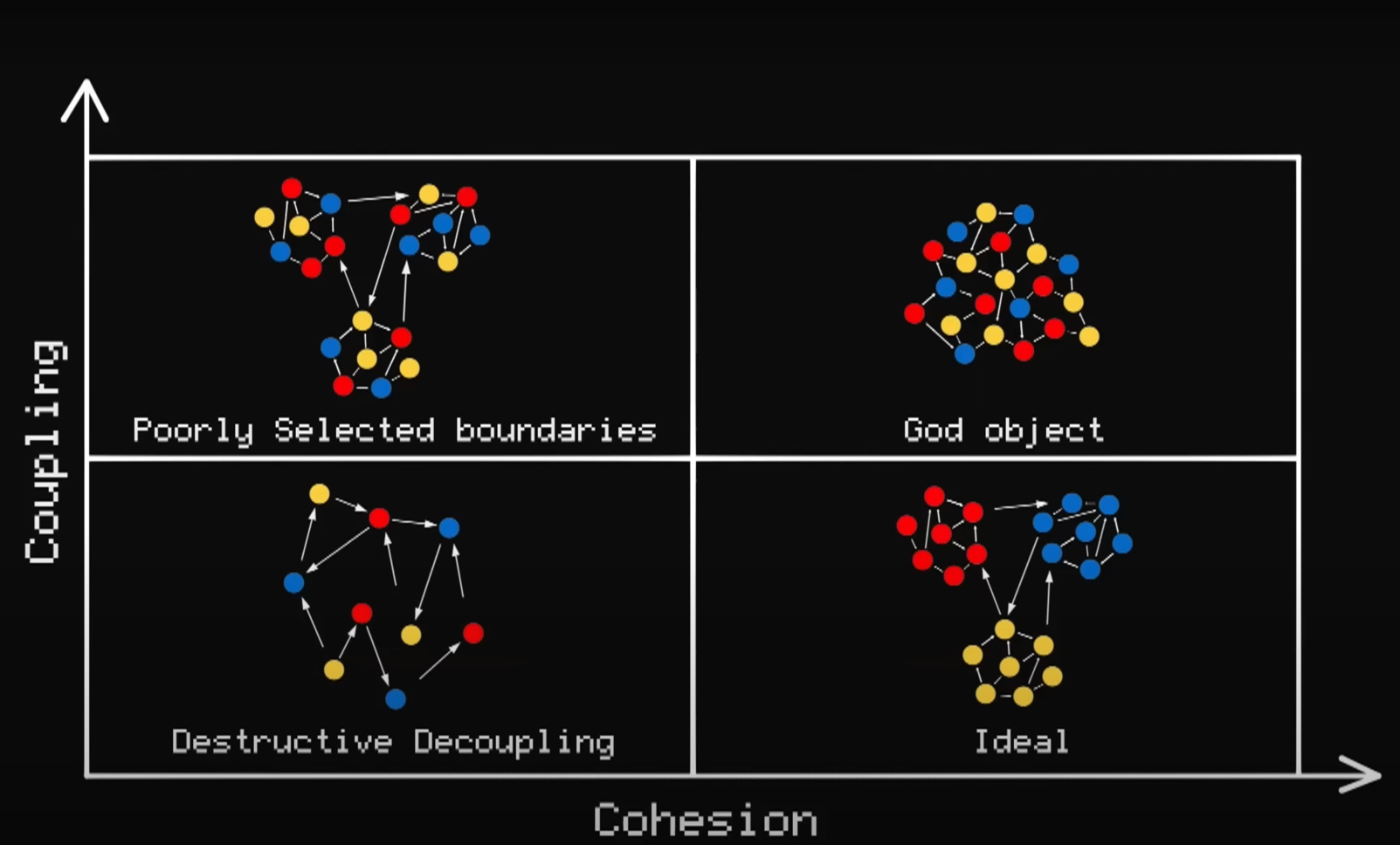
- описание того, как эти модули должны разрабатываться

- описание связей между моделями

- создание интерфейсов, описывающих предназначение модуля

Примеры архитектур:

1. Когда плохо выбраны границы модулей. Модули слишком сильно связаны друг с другом, в связи с чем, внесение изменений в один модуль потребуют изменений в другом.
2. Разрушительное разъединение. Модули слишком слабо связаны друг с другом. При выполнении определенной команды требуется множество вызовов между модуля, что приводит к чрезмерной нагрузке устройства.
3. Присутствует «объект бога». Один модуль содержит слишком много ответственности и функциональности. Поддержка такого модуля становится крайне тяжелой.
4. Хороший вариант. Стремится к слабой связности между модулями и сильной связности внутри каждого модуля.



Принципы:

- try

- kiss

- SOLID

- patterns

Удаление/изменение модулей должно быть простым/безболезненным по отношению к другим.

Уровни архитектур:

- микросервисная (взаимодействие через главный шлюз с сервисами, имеющими в свою очередь свои модули)

- монолитная

Виды архитектур:

- MVC

- Чистая архитектура

- ONION (многослойная)

- Архитектура frontend-приложений

- Реактивная архитектура

Singleton

Singleton (одиночка) - это паттерн проектирования, который гарантирует, что у класса есть только один экземпляр, и предоставляет глобальную точку доступа к этому экземпляру. Он обеспечивает возможность доступа к этому экземпляру из любой части программы и управление им.

Основные черты singleton паттерна:

* Один экземпляр: Класс singleton имеет только один экземпляр, который создается во время первого обращения к нему, а затем повторно используется во всех последующих обращениях.
* Глобальная точка доступа: Статический метод или свойство класса singleton предоставляет глобальную точку доступа к его экземпляру, что позволяет легко обращаться к нему из любой части программы.
* Простота использования: Singleton паттерн обеспечивает простоту использования и понимания, так как разработчики знают, что у класса есть только один экземпляр, и они могут обращаться к нему через глобальную точку доступа.
* Ленивая инициализация: Экземпляр класса singleton обычно создается лениво, то есть только когда он реально нужен, что помогает оптимизировать использование ресурсов.
* Паттерн в программировании: Singleton широко используется в программировании для реализации различных задач, таких как управление доступом к ресурсам, управление конфигурацией приложения, кэширование данных и другие.