

Принципы хорошей архитектуры

SOLID

S - single responsibility principle (одна задача одним классом и один класс одну задачу);

O - open-closed principle (закрыто для изменений и открыто для расширений);

L - Liskov substitution principle (ребёнка можно подставить вместо родителя);

I - interface segregation principle (много интерфейсов лучше одного);

D - dependency inversion principle (зависимости на абстракциях, а не реализациях).

Закон Деметры

If $A \rightarrow B \rightarrow C$, то A не должен обращаться к C.

- меньше взаимосвязей \rightarrow меньше связность.

Архитектура ПО

Что это и зачем нужно?

- думаем о долгосрочных решениях;
- обычно неточна.

Критерии хорошей архитектуры

- надёжность;
 - безопасность;
 - производительность;
 - масштабируемость (расширяемость, гибкость).
-
- avoid premature optimizations;
 - лучшее враг хорошего.

Критерии хорошей архитектуры

- масштабируемость процесса разработки;
- тестируемость;
- переиспользование;
- сопровождаемость.

Критерии плохой архитектуры

- жесткость (что-то тяжело изменить);
- хрупкость (изменения нарушают другие модели);
- неподвижность (тяжело извлечь модуль наружу).

High Cohesion + Low Coupling

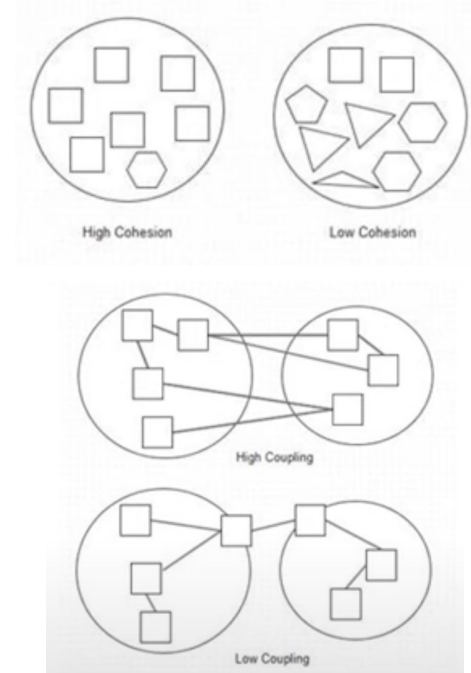
- High Cohesion

Модуль сфокусирован на одной задаче;

- Low Coupling

Слабая связь между модулями.

Есть и другие GRASP принципы (ещё 7).



DRY, WET, KISS, YAGNI.

- do not repeat yourself (!write everything twice);
- keep it simple, stupid;
- you aren't gonna need it;

Ссылочки

- <https://habr.com/ru/post/493782/>