

⇒ Spring Framework → modular Based
↓
multiple modules

\$ Spring Core → Base module

spring MVC → spring REST

spring JDBC

spring ORM

spring Data JPA

spring AOP

spring Security

— mail

— —

— —

— —

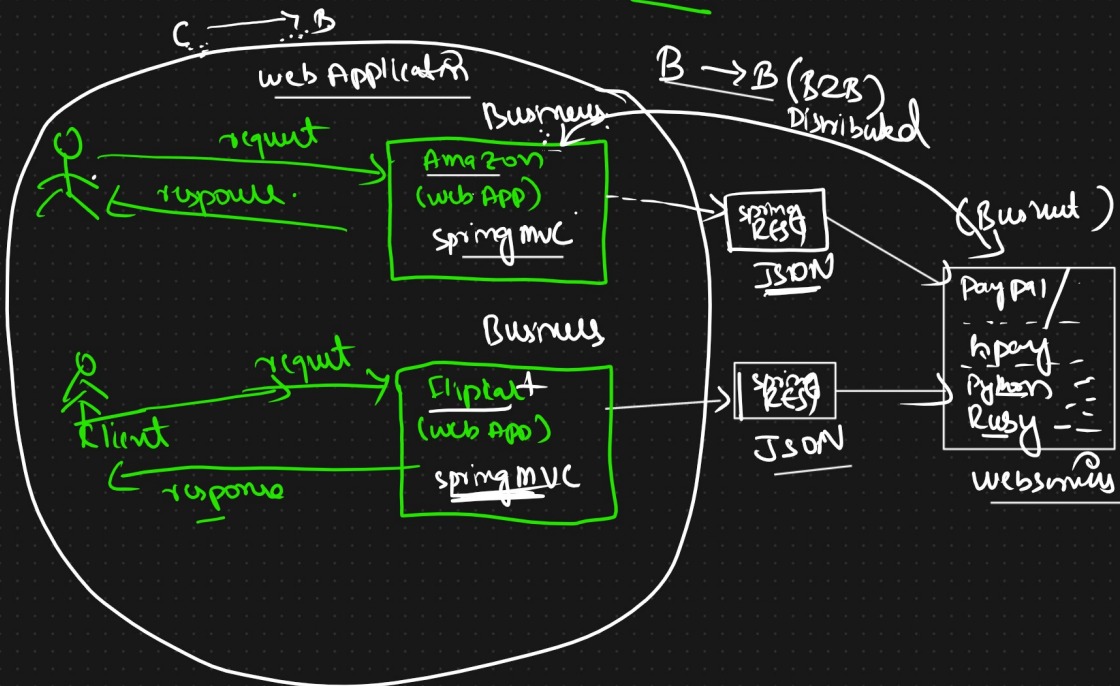
— —

⇒ Spring Framework → versatile → It can be integrated with other modules → other layer with diff tech/framework =>

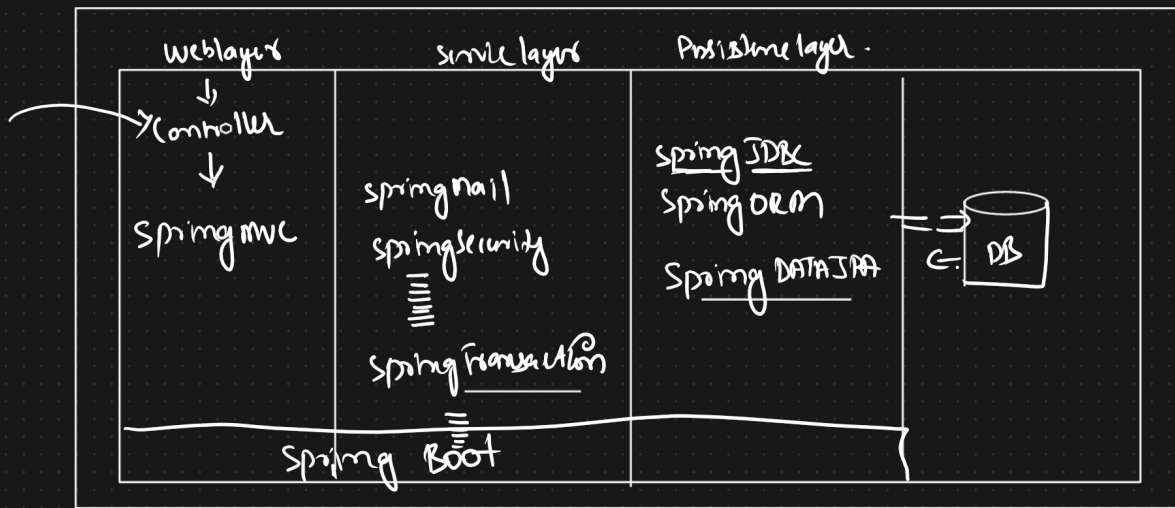
⇒ Non-Invasive ⇒ It will not force you to extend any of its class(es)
implement any interface.

→ End-to-end App (Backend) → All layers ⇒ 1

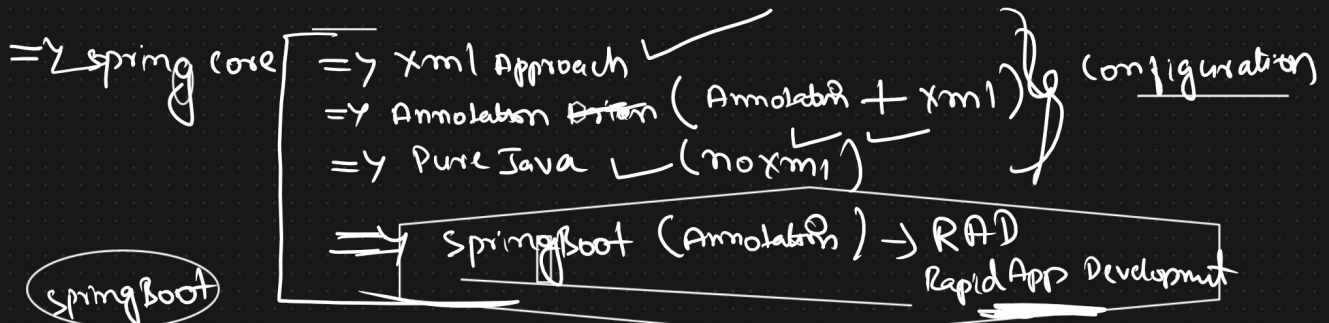
⇒ Enterprise level → Distributed APP



Application



→ spring core →

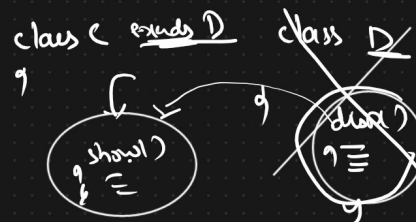


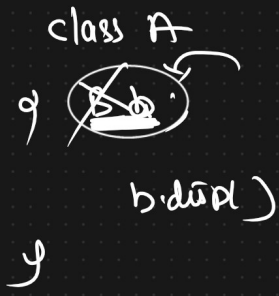
spring boot

- ⇒ spring JDBC
- ⇒ spring Data JPA (Lombok)
- ⇒ spring Profiles
- ⇒ logging → $\log_4 5$ | 5^4
- ⇒ spring mvc
- ⇒ spring AOP
- ⇒ spring REST

spring core ⇒ IOC & DI

Base

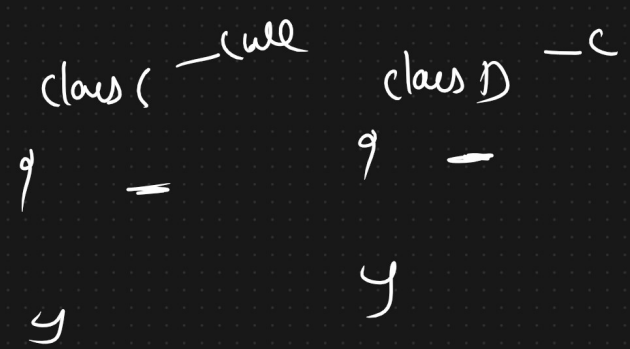
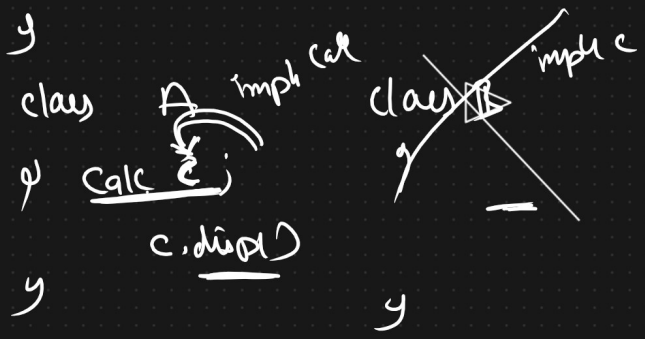




D d = new D(); } composition
 d.dupl();

⇒ Behavioral Design Pattern
 ↳ strategy DP

↳ code to interface → No direct implementation
 ↳ No direct class.



A a = new A();
 ↳
calc(c) = new A();