Ex4: Learning basic Spring Boot

1, Tight-coupling

* Tight-coupling xảy ra khi có một nhóm các lớp có sự ràng buộc, phụ thuộc cao tới các lớp khác trong chương trình.
* Khi chúng ta thay đổi, nâng cấp bảo trì một class sẽ kéo theo sự ảnh hưởng tới toàn bộ class.
* Nên hạn chế trong khi xây dựng và phát triển ứng dụng.

2, Loosely-couped

* Là mô hình thiết kế khuyến khích bạn nên thực hiện các nhiệm vụ với vai trò riêng biệt, ít liên kết ràng buộc với nhau
* Khi có lỗi hoặc thay đổi, nâng cấp, bảo trì phầm mềm sẽ ít ảnh hưởng tới toàn bộ hệ thống giúp dễ dàng mở rộng, cải thiện ứng dụng.
* Tiết kiệm tài nguyên phát triển ứng dụng, hạn chế sự lấn chiếm tài nguyên

3, Dependency Injection

* Dependency injection (DI) là một kỹ thuật lập trình giúp tách một class độc lập với các biến phụ thuộc.
* Nó thực hiện bằng cách cho phép ta truyền các phụ thuộc vào một thành phần thay vì định nghĩa chúng bên trong thành phần ấy.
* Các loại Dependency Injection:
* Constructor Injection: các biến phụ thuộc được cung cấp thông qua constructor.
* Setter injection: các biến phụ thuộc được truyền vào một class thông qua các setter class
* Interface injection: một giao diện được sử dụng để định nghĩa một phương thức chung để truyền các phụ thuộc. Lớp cần phụ thuộc sẽ implement giao diện này và triển khai phương thức tiêm phụ thuộc.
* Ưu điểm
* Linh hoạt và giảm ràng buộc
* Tăng khả năng kiểm tra
* Tăng tính module và dễ bảo trì
* Khả năng mở rộng và tái sử dụng cao
* Hạn chế
* Có sự ràng buộc về kiểu dữ liệu
* Quản lý cà cấu hình phức tạp
* Tăng độ phức tạp nếu sử dụng không đúng cách
* Nguy cơ tiềm ẩn và vòng lặp phụ thuộc

4, Các annotation thường được sử dụng

* @Autowire tự động nhúng các bean được Spring Container sinh ra vào các class được khai báo @Autowire (theo cơ chế DI).
* @Configuration class khai báo sử dụng @Configuration sẽ khai báo một hoặc nhiều @Bean method trong class đó
* @Bean đánh dấu trên method thông báo cho Spring, method đó sẽ sinh ra một bean và được quản lí bởi Spring.
* @ComponentScan thông báo có Spring container: “vào các package nào trong dự án để quét các Annotation và tạo Bean
* @Component class sẽ tạo thành 1 Bean và truyền vào các lớp nào cần dùng tới nó
* @Sevice là annotation đặc biệt của [@Component](https://viblo.asia/u/Component). Được dùng để sử lý nghiệp vụ, logic.
* @Reponsitory dùng để tao tác với cơ sở dữ liệu,