# Khái niệm

Dependency Injection hay gọi khác là IOC (Inversion of Control)

Cái này dựa trên nguyên tắc là: Dependency inversion principle (phụ thuộc động)

\_ Mỗi thành phần hệ thống chỉ nên phụ thuộc vào các abstraction, không nên phụ thuộc vào các implementation cụ thể

Ví dụ thực tế:

Vì sao có rất nhiều nhà sản xuất main, VGA, chip,… khác nhau. Nhưng vẫn có thể kết hợp lại với nhau 1 các hoàn hảo. Vì chúng đều phụ thuộc vào abstraction, nghĩa là phụ thuộc vào các chuẩn có sẵn.

Ví dụ denpendency injection:

<https://stormpath.com/blog/spring-boot-dependency-injection>

<https://nhungdongcodevui.com/2017/04/19/solid-la-gi-nguyen-tac-5-tinh-tuong-thich-dong-dependency-inversion-principle-dip/>

# Dấu hiệu

Ví dụ mình có object Student.

Bây giờ trong student có 1 đoạn ghi log, thì ghi log đó có thể ghi vào file hoặc ghi vào mail.

Nhưng mình chỉ sử dụng interface ghi log thôi, còn việc ghi log sẽ được xét bên ngoài, chứ k phải trong nội hàm Student

Student student\_1 = new Student();

//Việc gọi hàm applyForScholarship() và ghi log theo mong muốn rất dễ dàng

student\_1.setHandleLog(new ExportEmailLog());