**Ex4\_Spring Boot Basic**

Phạm Hải Triều

Báo cáo

### Tightly Coupled Objects (Các đối tượng được ghép nối chặt chẽ)

**Tightly Coupled** là khi một nhóm các lớp phụ thuộc nhiều vào nhau.

Tình huống này xảy ra khi một lớp đảm nhận quá nhiều trách nhiệm hoặc khi một mối quan tâm được dàn trải trên nhiều lớp thay vì có lớp riêng của nó.

Ví dụ về **Tightly Coupled**

class CustomerRepository

{

private readonly Database database;

public CustomerRepository(Database database)

{

this.database = database;

}

public void Add(string CustomerName)

{

database.AddRow("Customer", CustomerName);

}

}

class Database

{

public void AddRow(string Table, string Value)

{

}

}

### Loosely Coupled Objects (Các đối tượng được ghép nối lỏng lẻo )

Loosely Coupled đạt được nhờ một thiết kế thúc đẩy trách nhiệm đơn lẻ và tách biệt các mối quan tâm.

Một lớp liên kết lỏng lẻo có thể được sử dụng và kiểm tra độc lập với các lớp (cụ thể) khác.

Ví dụ về Loosely Coupled:

class CustomerRepository

{

private readonly IDatabase database;

public CustomerRepository(IDatabase database)

{

this.database = database;

}

public void Add(string CustomerName)

{

database.AddRow("Customer", CustomerName);

}

}

interface IDatabase

{

void AddRow(string Table, string Value);

}

class Database implements IDatabase

{

public void AddRow(string Table, string Value)

{

}

}

## Dependency Injection

**Dependency Injection ( tiêm phụ thuộc )**  là một thuật ngữ được sử dụng trong Lập trình hướng đối tượng , theo đó các Đối tượng sẽ tập trung vào việc thực hiện chức năng được giao và sử dụng các đối tượng khác. Các cấu hình và khởi tạo cần thiết sẽ không được xử lý bởi các đối tượng. Tuy nhiên, các Đối tượng sẽ cung cấp một cách để khởi tạo chúng và các phụ thuộc của chúng bằng cách gán trường, bộ thiết lập trường hoặc bộ tạo. Bằng cách này, các thực thể bên ngoài có thể khởi tạo các thứ chứ không phải các đối tượng thực tế.

Phụ thuộc tiêm làm cho thử nghiệm dễ dàng hơn. Việc tiêm có thể được thực hiện thông qua các nhà **xây dựng** .

SomeClass() có hàm tạo của nó như sau:

public SomeClass() {

myObject = Factory.getObject();

}

**Vấn đề** : Nếu myObjectliên quan đến các tác vụ phức tạp như truy cập đĩa hoặc truy cập mạng, thật **khó** để thực hiện kiểm tra đơn vị SomeClass(). Các lập trình viên phải chế giễu myObjectvà có thể **chặn** cuộc gọi của nhà máy.

**Giải pháp thay thế** :

* Truyền myObjectvào như là một đối số cho các nhà xây dựng

public SomeClass (MyClass myObject) {

this.myObject = myObject;

}

myObject có thể được thông qua trực tiếp mà làm cho thử nghiệm dễ dàng hơn.

* Một thay thế phổ biến là xác định một hàm tạo **không làm gì** . Tiêm phụ thuộc có thể được thực hiện thông qua setters. (h / t @MikeVella).
* [Martin Fowler](http://martinfowler.com/articles/injection.html#InterfaceInjection) ghi lại một giải pháp thay thế thứ ba (h / t @MarcDix), trong đó **các lớp thực hiện rõ ràng một giao diện** cho các lập trình viên phụ thuộc muốn đưa vào.

Khó có thể cô lập các thành phần trong thử nghiệm đơn vị mà không cần tiêm phụ thuộc.