**Aspect Oriented Programming – Lập trình hướng khía cạnh (AOP)**

Là mô hình được phát triển nhằm tang hiệu quả của OOP và tang cường tối đa khả năng tái sử dụng của code.

Là một mẫu lập trình nhằm tang tính module bằng cách cho phép phân tách những mối quan tâm xuyên suốt.

Nó làm như vậy bằng cách them hành vi bổ sung vào code đã tồn tại mà ko sửa đổi code đó, thay vì xác định cụ thể code nào đc sửa đổi thông qua đặc tả “Điểm cắt” (pointcut).

Điều này cho phép những hành vi ko phải là trung tâm của business đc them vào app mà ko cần làm ảnh hưởng đến các chức năng.

AOP đòi hỏi phải phá vỡ logic chương trình thành các phần riêng biệt.

AOP vận hành được thì phải có một container để quản lý, cụ thể là các server hay framework, lúc này chỉ cần báo cho container biết rằng phải làm gì qua các file cấu hình, thì container sẽ khởi tạo , dọn dẹp , thực thi…

Nên nếu muốn thay đổi nó ra sao thì chỉ việc thay đổi lại các file cấu hình và Annotiation nên việc ảnh hưởng đến cấu trúc là rất thấp, tang cao khả năng sử dụng lại.

OAP là giao quyền cho 1 container làm ( Đảo ngược sự điều khiển ) . Nhờ vậy mà các blackbox ko kết nối cứng với nhau nên dẽ tách hơn.

Trong ứng dụng thực thim thì Java có CDI và hệ thống Annotaion giúp cho việc làm AOP rất trực quan và hiệu năng rất cao.

AOP giải quyết vấn đề các mối quan tâm xuyên suốt, sẽ là bất kì code nào đc lặp lại trong các method khác nhau và thường ko thể được tái cấu trúc hoàn toàn thành module riêng của nó. Vì vậy , với AOP bạn có thể để những thứ đó ra khỏi code chính và định nghĩa nó theo chiều dọc