# Bài 1: Course Structure

+ Tổng quan về Logging Overview

+ Java Log API

+ Log4j

# Bài 2: Logging Overview

+ What logging is?

+ Advantages/Disadvantages of Logging

+ Logging Levels

+ Logging Libraries: Các framework thường dùng đối với Java

# Bài 3: Yêu cầu ban đầu

+ Có kiến thức về java core cơ bản

+ Nên biết về Maven

# Bài 4: What is logging?

Là những message cần ghi ra.

Logging là quá trình ghi log ra ngoài các phần khác như: console, files, database

+ Logger là một Object thực hiện quá trình logging của chương trình.

Muốn log các message thì cần phải sử dụng các hàm của Logger

# Bài 5: Ưu điểm của Logging

Nó là cơ chế tự động

Tùy thuộc vào config nếu ghi vào file, ghi ra màn hình.

Developer không cần phải làm gì?

+ Stored, Trackable, Readable: Các mess hỗ trợ user đọc được

+ Configurable: Config thiết lập cho logging

+ Footprints trong production

Dựa vào log mess thì enduser thực hiện những gì?

Ví dụ thanh toán 2 lần của user -> cần kiểm tra và phát hiện duplication ở phần nào.

Các Enterprise Application thì cần phải có log hết, lưu lại toàn bộ activity của người dùng.

Trên production thì bắt buộc phải có logging -> nếu không có thì không thể biết enduser làm gì trên hệ thống production cả.

# Bài 6: Disadvantages – nhược điểm

+ Performance:

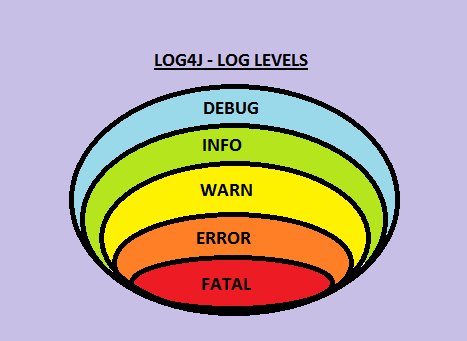
Dựa vào các business code, handler các exception có thể sảy ra.

Thực hiện các logic khác, cần đánh giá được các performance của nó.

+ Sentitive Data: Dữ liệu có thể bị lộ ra ngoài nếu dev làm lộ thông tin.

# Bài 7: Logging Levels

Các mức Level trong Logging, mức độ ảnh hưởng đến quá trình ghi log ra ngoài



Dựa theo từng framework thì có từng loại:

+ DEBUG:

Nó không được dùng trong production

+ INFO: Giới hạn hơn so với DEBUG

Lưu các thông tin như: Server đã được start/stop.

Incoming message, hoặc outcoming message, các activity thông thường của người dùng, like ảnh, share ảnh.

+ WARN: Thông điểm cảnh báo

Loss connection, đặt là threadhold, thông điểm về ổ cứng.

Conncetion pool cảnh báo

+ ERROR: Sử dụng logging exception, error.

Cần phải có trong code viết, handler trường hợp không mong muốn sảy ra.

Làm việc với các framework và thư viện khác thì có các Level khác nhau chút.

# Bài 8: Thư viện dùng cho Logging

+ Java Log API:

Một cái có sẵn của Java rồi, không cần phải sử dụng thêm ở ngoài.

Log4J: Một opensouce:

Logback: cung cấp một cơ chế mới hơn log4j

Log4j2

Slf4j: bản chất nó là Abstraction Layer cho log, không phải log framework

Kết hợp cùng với các logging khác, rất flexcible

Trong loạt bài này sẽ tìm hiểu về: JAVA LOG API và LOG4J

# Bài 9: Sample Project

Sử dụng cho Java Log API và Log4j

# Bài 10: Java Log Intro

+ Log Level

+ Basic Component

Các class thường dùng trong java log

+ Log File Sample

+ Log Factory

Sử dụng Factory pattern để tạo message log

# Bài 11: Java Log Levels

Trong Java API thì log Level có đến 7 giá trị: Thông qua việc define các level log thì tạo ra được các file log mức riêng

+ SEVERE(Error)

+ WARNING

+ INFO

+ CONFIG

+ FINE

+ FINER

+ FINEST

# Bài 12: Java Log Class

Khi làm việc với Java Log thì ta cần làm việc đến các class sau đây:

+ Filter

Cách lọc ra các message mong muốn

+ Formatter

+ Handler

+ Logger

Các instance thực sự tạo ra messge

# Bài 13: Simple Java Log Lab

+ Simple Java Log:

Thực hành việc ghi log ở mức cơ bản nhất

# Bài 14: Log Factory

# Bài 15: Java Log File Lab

# Bài 16: Log4j Intro

+ Log4j với Maven project

Là việc add các thư viện cần thiết cho logging với Log4j

+ Log4j.properties

Là file cấu hình các tham số cho Logging

+ Appenders

Là cơ chế mà dev có thể ghi những dòng log vào

+ Rolling Logs

Cho phép tạo nhiều file khác nhau khi file log cứ đẩy liên tục

# Bài 17: Peparation

# Bài 18: Multiple Classes with Log4j

Tạo nhiều Logging cho nhiều file class khác nhau

Nên tạo thành một LogFactory hoặc tạo getLogger cho từng file riêng biệt.

# Bài 19: Log4j Properties

File có thể thay đổi tham số

rootLogger

Appender

Cách thay đổi các Level để hiển thị ra các log mong muốn.

# Bài 20: FileAdapter

# Bài 21: Rolling File Adapter

Gần như là một yêu cầu bắt buộc với các hệ thống thực của java

Vì file sẽ được đọc một cách nhanh hơn rất nhiều của file 2Mb đối với file 2G

# Bài 22: Summary