Java Core Tutorial 12 – Java Synchronized, Thread Safe và Non-Thread Safe, Switch vs Enum

+ Cấu trúc của finnaly

+ Java Collections

Sử dụng với ConcurrentHashMap -> nó sẽ khác với HashMap -> sẽ làm tốt cho việc xử lý đồng thời.

Nếu mà viết trên môi trường Web thì chắc chắn nó sẽ là đa luồng.

Trong tại một thời điểm, nếu một Thread push giá trị vào -> nhưng đồng thời một Thread khác cũng push vào -> lúc đó nó sẽ là chỉ một.

Đa luồng cũng có lỗi khi làm việc với List.

+ Lỗi làm việc với List

Giải thích:

Chương trình làm việc với 2 thread.

Một thread sẽ làm nhiệm vụ khởi tạo phần tử, giá trị cho mảng.

Một thread khác sẽ làm nhiệm vụ đọc toàn bộ các phần tử của mảng đó lên.

Điều gì sẽ sảy ra khi gặp trường hợp này.

+ Bài toán video tiktok

Nếu hệ thống đang thực hiện việc duyệt các video.

Nếu nó đang duyệt mà lại gặp trường hợp insert mới một video vào nữa.

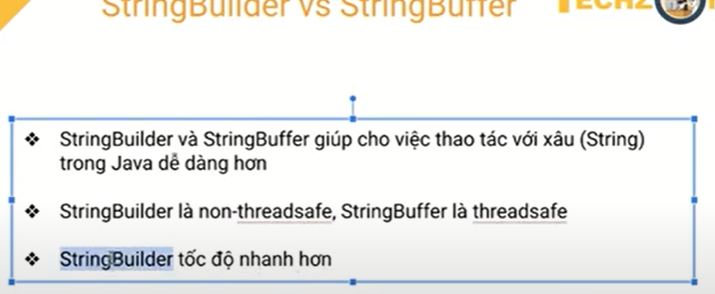
Lúc này sẽ sinh ra trường hợp hệ thống chết.

Chi ra một một tic tắc nào đó 2 thread vô tình nó trùng tại một thời điểm.

Trong ArrayList nó không có kiểu Concurrent

Lúc nó thao tác với List -> nó sẽ copy ra một list mới -> lúc đó sẽ không bị ảnh hưởng.

+ StringBuilder và StringBuffer

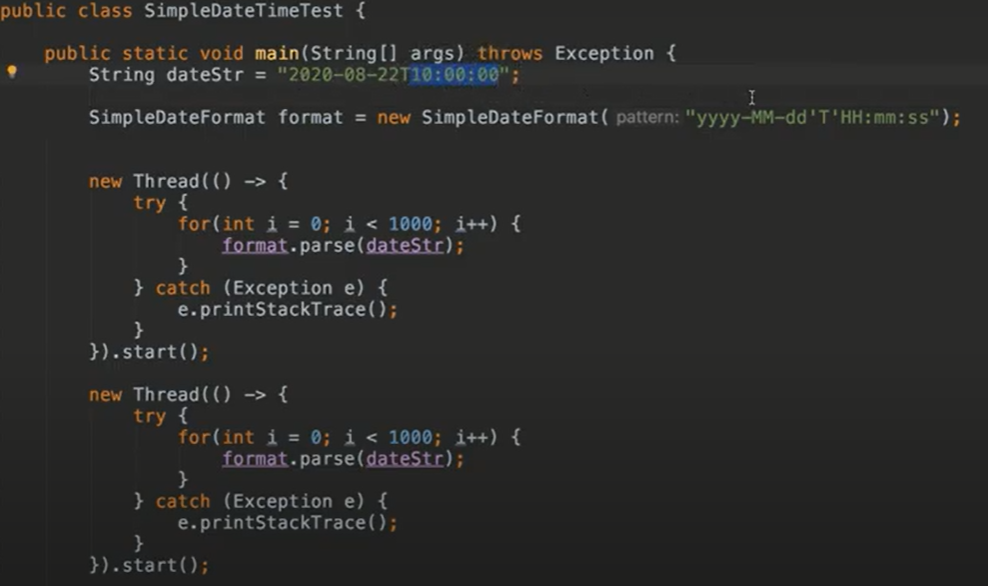


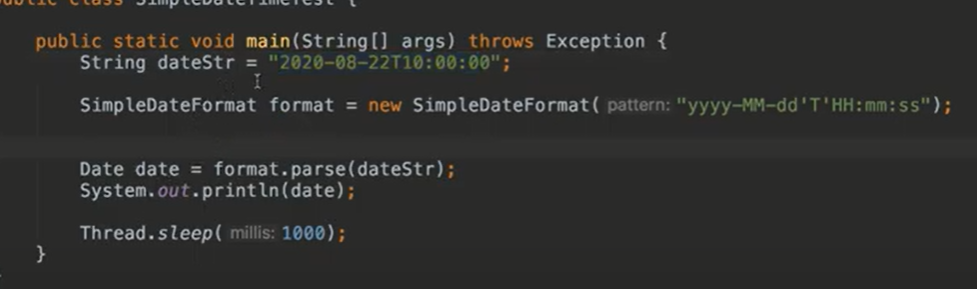
Vì liên quan đến khóa. StringBuffer tốc độ chập hơn.

+ SimpleDateFormat vs Thead

SimpleDateFormat là non threadsafe.

Thử việc convert các thời gian





Định nghĩa ThreadSafe có nghĩa là nó an toàn với đa luồng.

Còn non-threadsafe là không an toàn với đa luồng.

++ nếu dùng format date trong môi trường đa luồng -> thì SimpleDateFormat sẽ không control được dữ liệu đó.

Mô tả: Dùng 2 thread -> parse chúng khoảng 10k lần -> kiểm tra độ chính xác giữa các lần format đó.

Để tránh việc non-threadsafe thì phải dùng đến cơ chế là lock và unlock cho các block code. Và tăng thời gian sleep lên khoảng 5k.

+ Cách sử dụng cơ bản của Thread

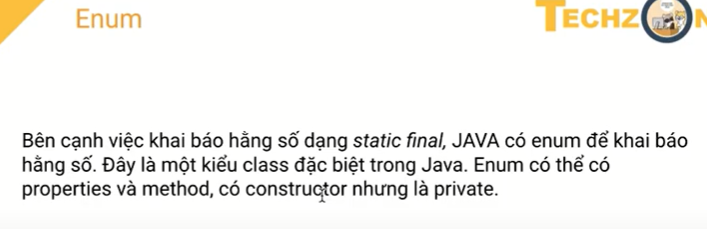
+ Các rủi ro làm với môi trường đa luồng

+ Cách để ngăn chặn các rỏi ro đó

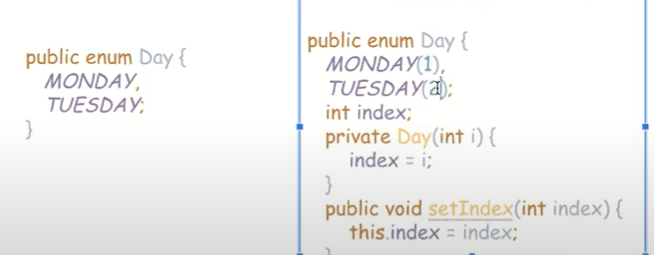
Hoặc có thể ngăn chặn việc sử dụng chung tài nguyên -> Hãy tách ra cho mỗi thread sử dụng một formatDate khác nhau -> khi đó việc non-threadsafe sẽ bị tránh.

+ Java Enum và Switch

+ Enum



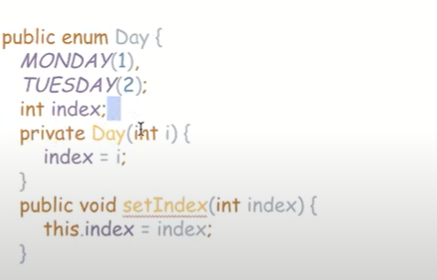
Phục vụ cho việc khai báo hằng số.



Khai báo và đưa hằng số cho chúng luôn.

Cùng là một kiểu Day -> nhưng có nhiều các tham số, hằng số đi kèm.

Kiểu thứ 2: Khai báo theo class



Lúc này các giá trị constructor của nó phải mang giá trị là private.

+ Bài tập:

Bài 1: Tính toán thời gian chạy của StringBuilder và StringBuffer với khoảng 50 triệu kí tự.

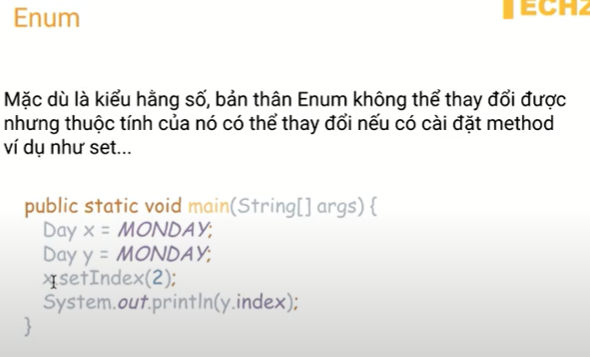
Hướng dẫn: System.getCurrentTime

Theo lý thuyết thì StringBuilder sẽ nhanh hơn. Còn StringBuffer sẽ chậm hơn.

Môi trường đa luồng thì phải giá trị lớn thì mới thấy sự khác biệt.

Bài 2: Phân biệt StringBuidder và StringBuffer? Ví dụ minh họa

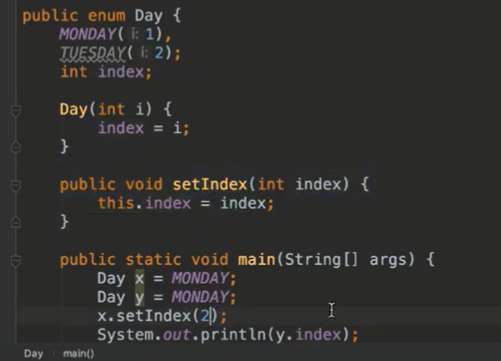
Bài 3: Kiểu giá trị Enum là gì? Ưu nhược điểm của nó mang lại là gì? Có bao nhiêu cách triển khai kiểu giá trị Enum trong thực tế. Đưa ra ví dụ minh họa.



Nó tương tự như các Object -> thuộc tính có thể thay đổi được nếu bên ngoài gọi đến các method thay đổi như setter.

?? Kiểm tra việc nếu 2 thằng là hằng số -> nếu thay đổi x thì y có thay đổi hay không? Vì 2 biến đang cùng trỏ vào một vị trí chứa giá trị là MONDAY.

?? Có thể đặt luôn enum là một class chứa hàm main luôn sao?



Hình: 12.x Class Day chứa luôn cả method main.

Phút thứ 59.