|  |  |
| --- | --- |
| **1. Khái niệm Multi-Tasking và Multi-Threading ?** | - Multi-Tasking là khả năng chạy đồng thời nhiều chương trình cùng một lúc trên cùng một hệ điều hành .  - Multi-Thread là khả năng thực hiện đồng thời nhiều tiểu trình trong một chương trình. |
| **2. Thread là gì ?** | - Thread là đơn vị nhỏ nhất của mã thực thi mà đoạn mã đó thực hiện một nhiệm vụ cụ thể. |
| **3. Cách tạo Thread với các class Thread Interface Runnable ?** | - Tạo Thread :  - Hệ thống đa luồng trong java được xây dựng trên lớp Thread và Interface Runnable thuộc gói java.lang  - Có 2 cách tạo Thread mới  + Cách 1 : Kế thừa từ lớp Thread có sẵn  + Cách 2 : Thực thi interface Runnable có sẵn  - Tạo Thread qua kế thừa từ lớp Thread :  + Bước 1 : Tạo lớp MyThread kế thừa từ lớp Thread  + Bước 2 : Override phương thức run()  + Bước 3 : Tạo đối tượng Thread và gọi phương thức start()  - Tạo Thread qua thực thi interface Runnable :  + Bước 1 : Tạo lớp MyRunnable thực thi interface Runnable  + Bước 2 : Implement phương thức run()  + Bước 3 : Tạo đối tượng Thread và gọi phương thức start() |
| **4. Trạng thái của Thread ?** | Các trạng thái vòng đời của Thread :  + **New** : Thread được tạo nhưng chưa đưuọc khởi chạy  + **Runnable** : Thread sẵn sàng chạy nhưng nhưng có thẻ chưa được cấp CPU  + **Running** : Thread đang thực thi (trạng thái này thực tế là một phần của "Runnable", vì không có trạng thái riêng gọi là "Running" trong mô hình của Java).  + **Sleeping** : Thread tạm dừng hoạt động trong một khoản thời gian nhất định , thường bằng phương thức Thread .sleep(ms).  + **Waiting** : Thread chờ vô thời hạn cho tín hiệu từ một thread khác.  + **Blocked** : Thread đang chờ để có quyền truy cập vào một tài nguyên hoặc một đoạn mã đồng bộ hóa.  + **Dead** : Thread đã hoàn thành hoặc bị dừng. |
| **5. Cách dùng start() , join() , sleep() ?** | - **Start()** :  + Dùng để khởi tạo một thread mới .  + Khi gọi Start() , Thread sẽ chuyển từ trạng thái từ trạng thái New sang Runnable và chờ CPU cấp quyền sử lý  - **Join()** :  + Dùng để đảm bảo một thread hoàn thành trước khi thread khác tiếp tục.  + Khi gọi join() , thread hiện tại sẽ chờ cho đến khi thread được gọi hoàn thành.  - **sleep()** :  + Dùng để tạm dừng một thread sẽ chuyển sang trạng thái Timed Waiting trong ms mili giây. |
| **6. Đồng bộ (Synchronous) và Bất dồng bộ (Asynchronous) ?** | - Đồng bộ (Sychronous) :  + Các tác vụ được thực hiện tuần tự , tức là một tác vụ phải hoàn thành trức khi tác vụ tiếp theo bắt đầu  + Dễ hiểu, dễ quản lý, nhưng có thể khiến chương trình bị **chậm**, đặc biệt khi có các tác vụ mất nhiều thời gian.  - Bất đồng bộ (Asynchronous) :  + Các tác vụ **có thể chạy song song**, không cần chờ tác vụ trước hoàn thành  + Hiệu suất tốt hơn khi xử lý các tác vụ tốn thời gian, như gọi API hoặc đọc/ghi dữ liệu.  + Cần lập trình viên quản lý trạng thái và kết quả đúng cách. |