**Câu 1**

Interface nào được dùng để định nghĩa một tác vụ có thể chạy đồng thời và có kết quả trả về?  
A. Runnable  
B. Callable  
C. Future  
D. Executor

**Câu 2**

Phương thức nào trong Thread sẽ tạm dừng thread hiện tại một thời gian cụ thể?  
A. yield()  
B. join()  
C. sleep()  
D. wait()

**Câu 3**

Phương thức start() của lớp Thread có tác dụng gì?  
A. Gọi trực tiếp hàm run()  
B. Tạo một thread mới và thực thi run()  
C. Kết thúc thread  
D. Tạm dừng thread

**Câu 4**

Lệnh nào tạo thread mới và bắt đầu chạy?  
A. Thread.run()  
B. new Thread().execute()  
C. new Thread().start()  
D. Runnable.run()

**Câu 5**

Phương thức join() dùng để?  
A. Chờ cho đến khi thread kia kết thúc  
B. Kết nối nhiều thread  
C. Ngắt thread khác  
D. Gán kết quả cho thread

**Câu 6**

Điều gì xảy ra nếu gọi start() hai lần trên cùng một Thread object?  
A. Thread chạy lại  
B. Không có gì xảy ra  
C. Throw IllegalThreadStateException  
D. Thread kết thúc sớm

**Câu 7**

Thread nào mặc định được khởi chạy đầu tiên trong chương trình Java?  
A. UI Thread  
B. System Thread  
C. Worker Thread  
D. Main Thread

**Câu 8**

Lớp nào đại diện cho một thread trong Java?  
A. Runnable  
B. Executor  
C. Thread   
D. Callable

**Câu 9**

Phương thức yield() có tác dụng gì?  
A. Tạm thời nhường CPU cho thread khác cùng độ ưu tiên  
B. Dừng vĩnh viễn thread  
C. Tăng độ ưu tiên thread  
D. Làm thread ngủ

**Câu 10**

Lệnh nào cho phép một thread chờ đến khi một thread khác kết thúc?  
A. wait()  
B. notify()  
C. sleep()  
D. join()

**Câu 11**

Thread-safe nghĩa là gì?  
A. Không thể bị lỗi  
B. Có khả năng đồng bộ dữ liệu khi dùng nhiều thread  
C. Chạy trên mọi hệ điều hành  
D. Không cần synchronized

**Câu 12**

Khi nào nên dùng notifyAll() thay vì notify()?  
A. Khi có một thread chờ  
B. Khi muốn đánh thức tất cả các thread đang chờ  
C. Khi không có thread nào chờ  
D. Khi dùng ExecutorService

**Câu 13**

Interface ExecutorService cung cấp gì?  
A. Tạo thread mới  
B. Quản lý thread pool  
C. Ngủ thread  
D. Gọi trực tiếp hàm run

**Câu 14**

Trong ExecutorService, phương thức nào chấm dứt executor?  
A. stop()  
B. end()  
C. shutdown()  
D. exit()

**Câu 15**

Sự khác biệt chính giữa Runnable và Callable là gì?  
A. Callable chạy chậm hơn  
B. Runnable có kết quả trả về  
C. Callable có thể ném Exception và trả về giá trị  
D. Runnable chỉ chạy 1 lần

**Câu 16**

Thread nào JVM sẽ tự động dừng nếu không còn user thread nào?  
A. Main Thread  
B. Garbage Collector  
C. Daemon Thread  
D. GUI Thread

**Câu 17**

Thread daemon khác gì so với user thread?  
A. Không bao giờ kết thúc  
B. Không được JVM quản lý  
C. JVM sẽ kết thúc daemon thread khi không còn user thread  
D. Có độ ưu tiên cao hơn

**Câu 18**

Điều gì xảy ra khi gọi wait() bên trong phương thức synchronized?  
A. Thread bị treo vĩnh viễn  
B. Thread bị ngắt ngay lập tức  
C. Thread giải phóng lock và vào danh sách chờ  
D. Thread tiếp tục chạy

**Câu 19**

Từ khóa nào dùng để đồng bộ phương thức?  
A. async  
B. volatile  
C. static  
D. synchronized

**Câu 20**

Race condition là gì?  
A. Một cuộc thi giữa các thread  
B. Xảy ra khi nhiều thread cùng truy cập và thay đổi dữ liệu dùng chung mà không đồng bộ  
C. Một cách quản lý thread  
D. Tình huống dùng synchronized sai

**Câu 21**

Khi nào cần dùng synchronized block thay vì synchronized method?  
A. Khi muốn lock toàn bộ object  
B. Khi không có shared data  
C. Khi chỉ muốn đồng bộ một phần code  
D. Khi dùng static method

**Câu 22**

Điều gì có thể xảy ra nếu không xử lý đúng InterruptedException?  
A. Thread chạy chậm hơn  
B. Thread không bao giờ bị đánh thức  
C. Thread không phản hồi khi bị ngắt  
D. JVM dừng toàn bộ ứng dụng

**Câu 23**

Khi gọi wait() trên object, thread sẽ?  
A. Chờ vô thời hạn đến khi có notify  
B. Chạy tiếp ngay lập tức  
C. Bị dừng vĩnh viễn  
D. Thoát khỏi chương trình

**Câu 24**

Deadlock xảy ra khi?  
A. Thread bị sleep  
B. Thread chờ nhau vô thời hạn giữ các lock  
C. Thread bị crash  
D. JVM lỗi

**Câu 25**

Phương thức interrupt() dùng để?  
A. Dừng thread ngay lập tức  
B. Đánh dấu thread bị yêu cầu ngắt  
C. Gán độ ưu tiên cho thread  
D. Dọn bộ nhớ

**Câu 26**

Thread pool giúp?  
A. Tạo thread mới mỗi lần  
B. Giảm overhead tạo thread nhiều lần  
C. Tăng tốc độ sleep()  
D. Tránh synchronized

**Câu 27**

Thread nào đảm nhiệm xử lý sự kiện GUI trong Swing?  
A. UI Thread  
B. Event Dispatch Thread  
C. System Thread  
D. Main Thread

**Câu 28**

Tại sao không nên thao tác GUI từ các thread khác ngoài Event Dispatch Thread?  
A. Vì GUI không hỗ trợ nhiều luồng  
B. Gây lỗi hoặc hành vi không mong muốn  
C. Gây deadlock  
D. Gây lag ứng dụng

**Câu 29**

Phương thức nào để thực thi tác vụ bất đồng bộ trong Swing mà không khóa GUI?  
A. Runnable.run()  
B. SwingUtilities.invokeLater()  
C. Thread.join()  
D. Thread.run()

**Câu 30**

Điểm khác nhau giữa shutdown() và shutdownNow() trong ExecutorService là gì?  
A. shutdownNow() dừng gấp và trả lại task chưa chạy  
B. shutdown() dừng ngay lập tức  
C. shutdownNow() đợi task chạy xong  
D. shutdown() ném lỗi InterruptedException