

- + [Java Cơ Bản](#)
- + [Các Khái Niệm Java OOPs](#)
- + [Java String](#)
- + [Xử Lý Ngoại Lệ](#)
- + [Các Lớp Lồng Nhau](#)
- + [Đa Luồng](#)
- ([Multithreading](#))
- ↳ [Java Thread](#)
- ↳ [Đa luồng trong java](#)
- ↳ [Vòng đời của Thread](#)
- ↳ [Tạo Thread](#)
- ↳ [Thread Scheduler](#)
- ↳ [Sleep một thread](#)
- ↳ [Start một thread hai lần](#)
- ↳ [Gọi phương thức run\(\)](#)
- ↳ [Phương thức join\(\)](#)
- ↳ [Đặt tên cho thread](#)
- ↳ [Mức ưu tiên của một Thread](#)
- ↳ [Luồng hiểm](#)
- ↳ [Thread Pool](#)
- ↳ [Thread Group](#)
- ↳ [Shutdown Hook](#)
- ↳ [Thực hiện nhiều task](#)
- ↳ [Garbage Collection](#)
- ↳ [Lớp Runtime](#)
- + [Java Synchronization](#)
- ↳ [Phương thức synchronization](#)
- ↳ [Khởi synchronized](#)
- ↳ [Đồng bộ static](#)
- ↳ [Deadlock trong java](#)
- ↳ [Giao tiếp giữa các thread](#)
- ↳ [Interrupt một Thread](#)
- + [Java AWT](#)
- + [Java Swing](#)
- + [Java I/O](#)
- + [Lập Trình Mạng Với Java](#)
- + [Java Date](#)
- + [Chuyển Đổi Kiểu Dữ Liệu](#)
- + [Java Collections](#)

## Tuyển Project / Product Manager cứng

Lương  
đến  
\$2,500+


[Apply Now](#)

## Tạo thread trong java

🕒 [Vòng đời của thread trong java](#)

[Thread scheduler trong java](#) 🕒

## Làm thế nào để tạo thread trong java

Có hai cách để tạo ra một thread:

1. Bởi extends lớp Thread
2. Bởi implements Runnable interface.

## Lớp Thread

Lớp Thread cung cấp các constructor và phương thức để tạo và thực hiện các hoạt động trên một thread. Lớp Thread extends từ lớp Object và implements Runnable interface.

Các constructor thường được sử dụng của lớp Thread

1. Thread()
2. Thread(String name)
3. Thread(Runnable r)
4. Thread(Runnable r,String name)

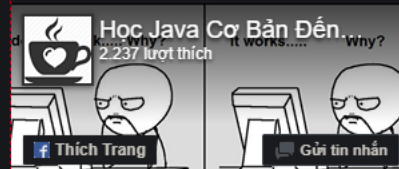
Các phương thức thường được sử dụng của lớp Thread

1. **public void run():** Được sử dụng để thực hiện hành động cho một thread..
2. **public void start():** Bắt đầu thực hiện thread. JVM gọi phương thức run() trên thread.
3. **public void sleep(long milliseconds):** Làm cho thread hiện tại tạm ngừng thực thi cho số mili giây quy định.
4. **public void join():** Đợi cho một thread chết.
5. **public void join(long milliseconds):** Đợi cho một thread chết với các mili giây quy định.
6. **public int getPriority():** Trả về mức độ ưu tiên của thread.
7. **public int setPriority(int priority):** Thay đổi mức độ ưu tiên của thread.
8. **public String getName():** Trả về tên của thread.
9. **public void setName(String name):** Thay đổi tên của thread.
10. **public Thread currentThread():** Trả về tham chiếu của thread đang được thi hành.
11. **public int getId():** Trả về id của thread.

## Recent Updates

- ↳ [Appium là gì?](#)
- ↳ [Kiểm thử tự động là gì? Automation Testing](#)
- ↳ [Tính đa hình trong java](#)
- ↳ [Tính kế thừa trong java](#)
- ↳ [Tính đóng gói trong java](#)
- ↳ [Từ khóa this trong java](#)
- ↳ [Từ khóa final trong java](#)
- ↳ [Từ khóa static trong java](#)
- ↳ [Access Modifier trong Java](#)
- ↳ [Lập trình hướng đối tượng\(OOPs\) trong java](#)
- ↳ [Bài tập java có lời giải](#)
- ↳ [Collection trong java](#)

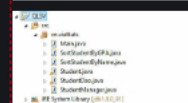
## VietTuts on facebook



1 người bạn thích nội dung này



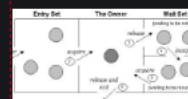
**Thực hiện nhiều task - học Java miễn phí hay nhất**



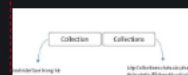
**Bài tập quản lý sinh viên trong java - bài tập java có lời giải**



**Deadlock trong java - học Java miễn phí hay nhất**



**Giao tiếp giữa các thread trong Java - học Java miễn phí...**





**Collection trong java - học Java miễn phí hay nhất**

+ Java JDBC

+ Các Tính Năng Mới Trong Java


+ Bài Tập Java Có Lời Giải

+ Câu Hỏi Phỏng Vấn Java




**-46%**

XEM



**-42%**

XEM



XEM

12. **public Thread.State getState():** Trả về trạng thái của thread.
13. **public boolean isAlive():** Kiểm tra nếu thread còn sống.
14. **public void yield():** Làm cho các đối tượng thread đang thực thi để tạm thời tạm dừng và cho phép các thread khác để thực thi.
15. **public void suspend():** Được sử dụng để hoãn lại các thread (không dùng nữa).
16. **public void resume():** Được sử dụng để tiếp tục các thread đang bị hoãn (không dùng nữa).
17. **public void stop():** Được sử dụng để dừng thread (không dùng nữa).
18. **public boolean isDaemon():** Kiểm tra nếu thread là một luồng hiểm.
19. **public void setDaemon(boolean b):** Đánh dấu thread là luồng hiểm hoặc luồng người dùng.
20. **public void interrupt():** Ngắt thread.
21. **public boolean isInterrupted():** Kiểm tra nếu thread đã bị ngắt.
22. **public static boolean interrupted():** Kiểm tra nếu thread hiện tại đã bị ngắt.

## Runnable interface

Giao tiếp Runnable nên được cài đặt bởi bất kỳ lớp nào mà thể hiện của lớp đó dự định sẽ được thực thi bởi thread. Giao tiếp Runnable chỉ có một phương thức run().

1. **public void run():** Được sử dụng để thực hiện hành động cho một thread.

## Start một thread

Phương thức start() của lớp Thread được sử dụng để bắt đầu một thread mới được tạo. Nó thực hiện các nhiệm vụ sau:

- Start một new thread(với new callstack).
- Thread chuyển từ trạng thái New sang trạng thái Runnable.
- Khi thread được một cơ hội để thực thi, phương thức run() của nó sẽ chạy.

## Ví dụ về thread trong java bởi extends lớp Thread

```
class Multi extends Thread {  
    public void run() {  
        System.out.println("thread is running...");  
    }  
  
    public static void main(String args[]) {  
        Multi t1 = new Multi();  
        t1.start();  
    }  
}
```

Output:

```
thread is running...
```

## Ví dụ về thread trong java bởi extends Runnable interface



Vòng đời của thread trong java - học Java miễn phí hay...

Bộ nhớ stack và heap trong java - học Java cơ bản...

Phương thức đồng bộ static - học Java miễn phí hay nhất

AWT trong java

Hibernate là gì



Más privacidad que con otros reproductores.

Conocer más



```

class Multi3 implements Runnable {
    public void run() {
        System.out.println("thread is running...");
    }

    public static void main(String args[]) {
        Multi3 m1 = new Multi3();
        Thread t1 = new Thread(m1);
        t1.start();
    }
}

```

Output:

```
thread is running...
```

Nếu bạn không extends lớp Thread, đối tượng lớp của bạn sẽ không được xử lý như là một đối tượng của thread. Đối tượng của lớp mà implements Runnable để phương thức run() của lớp đó có thể được thực thi.

🕒 Vòng đời của thread trong java

Thread scheduler trong java 🕒



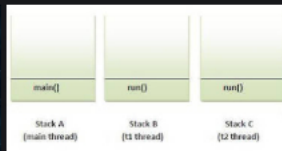
Zodiac Signs Astrology

QC AstroMary



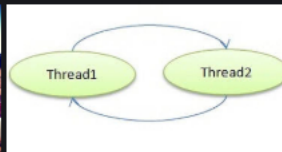
七夕在家看美剧, VPN加速优惠 - 现在购买只需10元-优惠 30%

QC ssrol.net



Thực hiện nhiều task - học Java miễn phí hay nhất

viettuts.vn



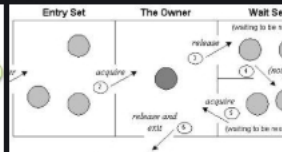
Deadlock trong java - học Java miễn phí hay nhất

viettuts.vn



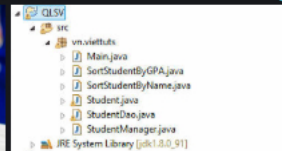
Đang được giới trẻ săn lùng

QC Pantu Mobile



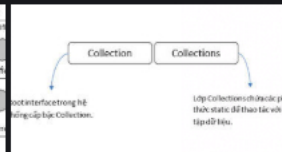
Giao tiếp giữa các thread trong Java - học Java miễn phí hay nhất

viettuts.vn



Bài tập quản lý sinh viên trong java - bài tập java có lời giải

viettuts.vn



Collection trong java - học Java miễn phí hay nhất

viettuts.vn

**Học Bổng Lên Đến 5,000,000 VNĐ**

Phát triển khả năng giao tiếp và khả năng tư duy của công dân toàn cầu cùng ILA ILA Vietnam

**ĐĂNG KÝ**

0 bình luận

Sắp xếp theo


Cũ nhất



Thêm bình luận



Thêm bình luận...

 Plugin bình luận trên Facebook

## Danh sách bài học

[Học java](#)  
[Học servlet](#)  
[Học jsp](#)  
[Học Hibernate](#)  
[Học Struts2](#)  
[Học Spring](#)  
[Học SQL](#)

## Câu hỏi phỏng vấn

[201 câu hỏi phỏng vấn java](#)  
[25 câu hỏi phỏng vấn servlet](#)  
[75 câu hỏi phỏng vấn jsp](#)  
[52 câu hỏi phỏng vấn Hibernate](#)  
[70 câu hỏi phỏng vấn Struts2](#)  
[70 câu hỏi phỏng vấn Spring](#)  
[57 câu hỏi phỏng vấn SQL](#)

## About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...

