**Sử dụng CountDownLatch trong Java**

*Đăng vào 09/03/2018 . Được đăng bởi*[**GP Coder**](https://gpcoder.com/author/gpadmin/)*.****4357****Lượt xem . Toàn màn hình*

[**CountDownLatch**](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/CountDownLatch.html) được sử dụng để đảm bảo rằng một tác vụ chờ các **Thread** khác hoàn thành trước khi nó bắt đầu thực thi.

Khi chúng ta tạo ra một đối tượng của **CountDownLatch**, chúng ta chỉ định số lượng các Thread cần chờ đợi, tất cả các Thread đó được yêu cầu phải đếm ngược bằng cách gọi **CountDownLatch.countDown()** khi chúng đã hoàn thành. Ngay sau khi tính đến số 0, nhiệm vụ chờ đợi bắt đầu chạy.

[Diagram

Description automatically generated](https://gpcoder.com/wp-content/uploads/2018/03/CountdownLatch.png)

Ví dụ: Một chương trình chỉ được start sau khi 3 service của nó đã được start. Với yêu cầu này, chúng ta có thể sử dụng phương thức [**join()**](https://gpcoder.com/3484-lap-trinh-da-luong-trong-java-java-multi-threading/#Su_dung_join_vajoinlong) của lớp **Thread** hoặc sử dụng lớp **CountDownLatch**.

**Nội dung**[[Ẩn](https://gpcoder.com/3594-su-dung-countdownlatch-trong-java/)]

* [1 Ví dụ sử dụng Thread.join()](https://gpcoder.com/3594-su-dung-countdownlatch-trong-java/#Vi_du_su_dung_Threadjoin)
* [2 Ví dụ sử dụng CountDownLatch](https://gpcoder.com/3594-su-dung-countdownlatch-trong-java/#Vi_du_su_dungCountDownLatch)

**Ví dụ sử dụng Thread.join()**

Đầu tiên chúng ta hãy xem chương trình sử dụng phương thức join() của lớp Thead:

ServiceOne.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | **package** com.gpcoder.countdownlatch.join;    **public** **class** ServiceOne **implements** Runnable {        @Override  **public** **void** run() {  **try** {              Thread.sleep(2000);          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Started service One");      }  } |

ServiceTwo.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **package** com.gpcoder.countdownlatch.join;    **public** **class** ServiceTwo **implements** Runnable {        @Override  **public** **void** run() {  **try** {              Thread.sleep(1000);          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Started service Two");      }    } |

ServiceThree.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **package** com.gpcoder.countdownlatch.join;    **public** **class** ServiceThree **implements** Runnable {        @Override  **public** **void** run() {  **try** {              Thread.sleep(3000);          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Started service three");      }    } |

Chương trình chính phải chờ 3 service ServiceOne và ServiceTwo hoàn thành trước khi start.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | **package** com.gpcoder.countdownlatch.join;      **public** **class** ThreadJoinExample {    **public** **static** **void** main(String[] args) {            // making two threads for 3 services          Thread service1 = **new** Thread(**new** ServiceOne());          Thread service2 = **new** Thread(**new** ServiceTwo());          Thread service3 = **new** Thread(**new** ServiceThree());            service1.start();          service2.start();          service3.start();            // using thread.join() to make sure that the execution of main thread only          // finishes ones 3 services have executed  **try** {              System.out.println("Waiting for 3 services have started ... ");              service1.join();              service2.join();              service3.join();              System.out.println("Starting main Thread ... ");          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Done!!!");      }    } |

Thực thi chương trình trên, ta có kết quả như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | Waiting **for** 3 services have started ...  Started service Two  Started service One  Started service three  Starting main Thread ...  Done!!! |

**Ví dụ sử dụng CountDownLatch**

ServiceOne.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | **package** com.gpcoder.countdownlatch;    **import** java.util.concurrent.CountDownLatch;    **public** **class** ServiceOne **implements** Runnable {  **private** **final** CountDownLatch latch;    **public** ServiceOne(CountDownLatch latch) {  **this**.latch = latch;      }        @Override  **public** **void** run() {  **try** {              Thread.sleep(2000);          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Started service One");          latch.countDown();      }  } |

ServiceTwo.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **package** com.gpcoder.countdownlatch;    **import** java.util.concurrent.CountDownLatch;    **public** **class** ServiceTwo **implements** Runnable {    **private** **final** CountDownLatch latch;    **public** ServiceTwo(CountDownLatch latch) {  **this**.latch = latch;      }        @Override  **public** **void** run() {  **try** {              Thread.sleep(1000);          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Started service Two");          latch.countDown();      }    } |

ServiceThree.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **package** com.gpcoder.countdownlatch;    **import** java.util.concurrent.CountDownLatch;    **public** **class** ServiceThree **implements** Runnable {    **private** **final** CountDownLatch latch;    **public** ServiceThree(CountDownLatch latch) {  **this**.latch = latch;      }        @Override  **public** **void** run() {  **try** {              Thread.sleep(3000);          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Started service three");          latch.countDown();      }    } |

Chương trình chính phải chờ 3 service ServiceOne và ServiceTwo hoàn thành trước khi start.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | **package** com.gpcoder.countdownlatch;    **import** java.util.concurrent.CountDownLatch;    **public** **class** CountDownLatchExample {    **public** **static** **void** main(String[] args) {          // intialising count down latch by 3, as it will wait for 3 threads to          // finish execution  **final** CountDownLatch latch = **new** CountDownLatch(3);            // making two threads for 3 services          Thread service1 = **new** Thread(**new** ServiceOne(latch));          Thread service2 = **new** Thread(**new** ServiceTwo(latch));          Thread service3 = **new** Thread(**new** ServiceThree(latch));            service1.start();          service2.start();          service3.start();            // latch waits till the count becomes 0          // this way we can make sure that the execution of main thread only          // finishes ones 3 services have executed  **try** {              System.out.println("Waiting for 3 services have started ... ");              latch.await();              System.out.println("Starting main Thread ... ");          } **catch** (InterruptedException e) {              e.printStackTrace();          }          System.out.println("Done!!!");      }    } |

Như bạn thấy, tôi đã khởi tạo đối tượng CountDownLatch  với tham số là 3 để chỉ ra rằng chương trình chính phải chờ 3 Thread hoàn thành, tức là countdown là 0 mới được phép thực thi.

Mỗi Thread (ServiceOne, ServiceTwo) khi hoàn thành phải gọi phương thức **CountDownLatch.countDown()** để thông báo là Thead này đã hoàn thành, tức là giảm giá trị countdown (countdown–).

Sử dụng phương thức **CountDownLatch.await()** để thông báo chương trình chính phải chờ các Thread hoàn thành trước khi nó bắt đầu.

**Lưu ý:** chúng ta có thể sử dụng phương thức **CountDownLatch.await(long timeout, TimeUnit unit)** để chỉ ra rằng chương trình chính phải chờ cho tới khi countdown là 0 hoặc sau khoản thời gian timeout.

Thực thi chương trình ta có kết quả như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | Waiting **for** 3 services have started ...  Started service Two  Started service One  Started service three  Starting main Thread ...  Done!!! |

Như bạn thấy, cả 2 cách sử dụng phương thức **Thread.join()** và **CountDownLatch** kết quả của chương trình cũng không khác biệt. Tuy nhiên, nếu số lượng Thread của chương trình tăng lên thì bạn cần phải khai báo Thread.join() cho mỗi Thread. Điều này khá phiền phức, thay vào đó chúng ta nên sử dụng CountDownLatch cho trường hợp này.

Trong các ứng dụng thực tế, CountDownLatch thường được sử dụng để test các ứng dụng đa luồng. Bằng cách sử dụng CountDownLatch, nó có thể đảm bảo rằng nhiều luồng được kích hoạt yêu cầu đồng thời hoặc thực hiện mã cùng một lúc.

Trong bài tiếp theo, chúng ta sẽ tìm hiểu cách sử dụng **CyclicBarrier**, một lớp tiện ích khác cũng nằm trong gói java.util.Concurrent.

**5.0**

13

**Nếu bạn thấy hay thì hãy chia sẻ bài viết cho mọi người nhé!**

SHARES

*Chuyên mục:*[**Multi-Thread**](https://gpcoder.com/category/java-core/multi-thread/)*Được gắn thẻ:*[**CountDownLatch**](https://gpcoder.com/tag/countdownlatch/)*,*[**Multithreading**](https://gpcoder.com/tag/multithreading/)

[Sử dụng Fork/Join Framework với ForkJoinPool trong Java](https://gpcoder.com/3573-su-dung-fork-join-framework-voi-forkjoinpool-trong-java/)

[Sử dụng CyclicBarrier trong Java](https://gpcoder.com/3603-su-dung-cyclicbarrier-trong-java/)

**Có thể bạn muốn xem:**

* [Sử dụng CyclicBarrier trong Java](https://gpcoder.com/3603-su-dung-cyclicbarrier-trong-java/) **(13/03/2018)**
* [Lập trình đa luồng trong Java (Java Multi-threading)](https://gpcoder.com/3484-lap-trinh-da-luong-trong-java-java-multi-threading/) **(12/02/2018)**
* [Semaphore trong Java](https://gpcoder.com/6187-semaphore-trong-java/) **(18/09/2019)**
* [Đồng bộ hóa các luồng trong Java](https://gpcoder.com/3514-dong-bo-hoa-cac-luong-trong-java/) **(20/02/2018)**
* [Lập trình đa luồng với Callable và Future trong Java](https://gpcoder.com/3565-lap-trinh-da-luong-voi-callable-va-future-trong-java/) **(02/03/2018)**

**Bình luận**

2 bình luận

[](https://www.facebook.com/profile.php?id=100010036307790)

[**Bùi Đức Dương**](https://www.facebook.com/profile.php?id=100010036307790)

Tui thấy k hợp lý lắm. cái Join() nó là đợi Thread hoàn thành. Còn CountDownLatch là đến thời điểm xác định thì mới có thể chạy tiếp Thread hiện tại. (ở đây là 3 thằng kia khởi động xong, vẫn còn code vẫn chạy tiếp song song với thằng main) .

[Thích](https://www.facebook.com/plugins/feedback.php?app_id=510733472615500&channel=https%3A%2F%2Fstaticxx.facebook.com%2Fx%2Fconnect%2Fxd_arbiter%2F%3Fversion%3D46%23cb%3Df24a3601b00d16%26domain%3Dgpcoder.com%26is_canvas%3Dfalse%26origin%3Dhttps%253A%252F%252Fgpcoder.com%252Ff25f4040bcfb87c%26relation%3Dparent.parent&color_scheme=light&container_width=691&height=100&href=https%3A%2F%2Fgpcoder.com%2F3594-su-dung-countdownlatch-trong-java%2F&locale=vi_VN&numposts=5&sdk=joey&width) · [Phản hồi](https://www.facebook.com/plugins/feedback.php?app_id=510733472615500&channel=https%3A%2F%2Fstaticxx.facebook.com%2Fx%2Fconnect%2Fxd_arbiter%2F%3Fversion%3D46%23cb%3Df24a3601b00d16%26domain%3Dgpcoder.com%26is_canvas%3Dfalse%26origin%3Dhttps%253A%252F%252Fgpcoder.com%252Ff25f4040bcfb87c%26relation%3Dparent.parent&color_scheme=light&container_width=691&height=100&href=https%3A%2F%2Fgpcoder.com%2F3594-su-dung-countdownlatch-trong-java%2F&locale=vi_VN&numposts=5&sdk=joey&width) · [1 năm](https://gpcoder.com/3594-su-dung-countdownlatch-trong-java/?fb_comment_id=6438948289510236)

[](https://www.facebook.com/profile.php?id=100023171116103)

[**Vũ Đạt**](https://www.facebook.com/profile.php?id=100023171116103)

Cho em khi chạy chương trình java trên netbean nó có Logger cái này,nó có ghi lỗi vào một file nào đấy không anh.Hoặc muốn đọc lỗi khi chạy chương trình java đã buil thì đọc ở đâu hay chỉ có cách là mình tự viết hàm ghi lỗi ra vậy anh

[Thích](https://www.facebook.com/plugins/feedback.php?app_id=510733472615500&channel=https%3A%2F%2Fstaticxx.facebook.com%2Fx%2Fconnect%2Fxd_arbiter%2F%3Fversion%3D46%23cb%3Df24a3601b00d16%26domain%3Dgpcoder.com%26is_canvas%3Dfalse%26origin%3Dhttps%253A%252F%252Fgpcoder.com%252Ff25f4040bcfb87c%26relation%3Dparent.parent&color_scheme=light&container_width=691&height=100&href=https%3A%2F%2Fgpcoder.com%2F3594-su-dung-countdownlatch-trong-java%2F&locale=vi_VN&numposts=5&sdk=joey&width) · [Phản hồi](https://www.facebook.com/plugins/feedback.php?app_id=510733472615500&channel=https%3A%2F%2Fstaticxx.facebook.com%2Fx%2Fconnect%2Fxd_arbiter%2F%3Fversion%3D46%23cb%3Df24a3601b00d16%26domain%3Dgpcoder.com%26is_canvas%3Dfalse%26origin%3Dhttps%253A%252F%252Fgpcoder.com%252Ff25f4040bcfb87c%26relation%3Dparent.parent&color_scheme=light&container_width=691&height=100&href=https%3A%2F%2Fgpcoder.com%2F3594-su-dung-countdownlatch-trong-java%2F&locale=vi_VN&numposts=5&sdk=joey&width) · [4 năm](https://gpcoder.com/3594-su-dung-countdownlatch-trong-java/?fb_comment_id=5613769632028110)

[](https://facebook.com/GPCoder)

[**GP Coder - Lập trình Java**](https://facebook.com/GPCoder)

Class System.out chỉ show log trên console. Nếu muốn ghi vào file e cần sử dụng một thư viện Logger bên ngoài. Nó cho phép log theo từng loại như error, info, warning ...  
  
Chi tiết e xem trong bài này: [https://gpcoder.com/5500-gioi-thieu-java-logging/](https://l.facebook.com/l.php?u=https%3A%2F%2Fgpcoder.com%2F5500-gioi-thieu-java-logging%2F%3Ffbclid%3DIwAR3PRnXvYmOboSFnZWRevvh2p_7Kf_fvjXEYhE2n9g0whLuwuQGyDZWMLh0&h=AT1uqA2yxvzr_7ZfoX6NG7Ak7Vb07K3IU09aZf8gp4HgnkSzBN43vn11f3H65ZtVvoGYoAcafkwhOdQc5zFpc_RMPLLisgLGRZrdFH8bMw0ayo3qxxtyctoEzimllWnxeHXg8A)

[Thích](https://www.facebook.com/plugins/feedback.php?app_id=510733472615500&channel=https%3A%2F%2Fstaticxx.facebook.com%2Fx%2Fconnect%2Fxd_arbiter%2F%3Fversion%3D46%23cb%3Df24a3601b00d16%26domain%3Dgpcoder.com%26is_canvas%3Dfalse%26origin%3Dhttps%253A%252F%252Fgpcoder.com%252Ff25f4040bcfb87c%26relation%3Dparent.parent&color_scheme=light&container_width=691&height=100&href=https%3A%2F%2Fgpcoder.com%2F3594-su-dung-countdownlatch-trong-java%2F&locale=vi_VN&numposts=5&sdk=joey&width) · [Phản hồi](https://www.facebook.com/plugins/feedback.php?app_id=510733472615500&channel=https%3A%2F%2Fstaticxx.facebook.com%2Fx%2Fconnect%2Fxd_arbiter%2F%3Fversion%3D46%23cb%3Df24a3601b00d16%26domain%3Dgpcoder.com%26is_canvas%3Dfalse%26origin%3Dhttps%253A%252F%252Fgpcoder.com%252Ff25f4040bcfb87c%26relation%3Dparent.parent&color_scheme=light&container_width=691&height=100&href=https%3A%2F%2Fgpcoder.com%2F3594-su-dung-countdownlatch-trong-java%2F&locale=vi_VN&numposts=5&sdk=joey&width) · 1 năm