



java.util.concurrent.Phaser

Współbieżność w Java 7 małymi krokami

Jacek Laskowski
jacek@japila.pl
<http://www.JacekLaskowski.pl>

Jacek Laskowski

- **Entuzjasta** Java EE, OSGi, SCA oraz programowania funkcyjnego (Clojure)
- Założyciel i lider **Warszawa JUG**
- Organizator konferencji **Javarsovia, Confitura, warsjava**
- Członek zespołu **NetBeans DreamTeam**
- Blogger na <http://JacekLaskowski.pl>
- Blogger na <http://blog.japila.pl>
- Twituje jako **@jaceklaskowski**
- Członek zespołów **Apache Geronimo** i **Apache OpenEJB**
- Specjalista produktów **IBM WebSphere** w IBM Polska

konferencja społeczności jadowej w Polsce

Confitura 2012

<http://confitura.pl>

8-9 lipiec

Warszawa

Problem: Synchronizacja wątków

- W jaki sposób wstrzymać zadanie do czasu zakończenia pracy przez inne zadanie?
- **Wątek** - współbieżne zadanie
- **Szlaban** (bariera) - punkt koordynacji i synchronizacji wątków
- Wątek **await()** i kontynuuje pracę
- j.u.c.{**CountDownLatch** | **CyclicBarrier** | **Phaser** }

CyclicBarrier - Java 5

- **Niezmienna liczba uczestników** (wątków)
- *Cyclic* - Możliwe **ponowne użycie** po zwolnieniu oczekujących wątków
- Pojedyncze **odpalenie *Runnable***
 - po dotarciu ostatniego wątku, a przed ich zwolnieniem
 - aktualizacja współdzielonego stanu
- Dostępne od Java 5

CyclicBarrier API

- **CyclicBarrier**(int parties)
- **CyclicBarrier**(int parties, Runnable barrierAction)
- int **getParties**()
- int **await**() throws InterruptedException, BrokenBarrierException
- int **await**(long timeout, TimeUnit unit) throws InterruptedException, BrokenBarrierException

Phaser - historia

- Doug Lea w [concurrency-interest] Phasers (were:TaskBarriers)

<http://altair.cs.oswego.edu/pipermail/concurrency-interest/2008-July/005345.html>

- Początkowo użyte do synchronizacji w Fork/Join

- New Java Concurrency Feature: Phasers

<http://www.infoq.com/news/2008/07/phasers>

- “Phasers: a Unified Deadlock-Free Construct for Collective and Point-to-point Synchronization”

<http://www.cs.rice.edu/~vsarkar/PDF/SPSS08-phasers.pdf>

Phaser - Java 7

- Zmienna liczba uczestników (wątków)
- Wielokrotne użycie
- Struktura drzewiasta zależnych Phaser'ów
- Liczba zarejestrowanych wyłącznie dla celów “księgowych”
- Numerowane fazy (etap, iteracja) - (0...Integer.MAX_VALUE)
 - dotarcie wszystkich wątków podbija numer
- Odpalenie akcji

Phaser API

- **Phaser(...)**
- int **register()** lub int **bulkRegister**(int parties)
- int **arrive()** lub int **arriveAndDeregister()**
- int **arriveAndAwaitAdvance()**
- protected boolean **onAdvance**(int phase, int registeredParties)

```
void runTasks(List<Runnable> tasks) {
    final Phaser phaser = new Phaser(1); // "1" to register self
    // create and start threads
    for (final Runnable task : tasks) {
        phaser.register();
        new Thread() {
            public void run() {
                phaser.arriveAndAwaitAdvance(); // await all creation
                task.run();
            }
        }.start();
    }

    // allow threads to start and deregister self
    phaser.arriveAndDeregister();
}
```

```

void startTasks(List<Runnable> tasks, final int iterations) {
    final Phaser phaser = new Phaser() {
        protected boolean onAdvance(int phase, int registeredParties) {
            return phase >= iterations || registeredParties == 0;
        }
    };
    phaser.register();
    for (final Runnable task : tasks) {
        phaser.register();
        new Thread() {
            public void run() {
                do {
                    task.run();
                    phaser.arriveAndAwaitAdvance();
                } while (!phaser.isTerminated());
            }
        }.start();
    }
    phaser.arriveAndDeregister(); // deregister self, don't wait
}

```

Warto odwiedzić

- Dokumentacja j.u.c.Phaser
<http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Phaser.html>
- Java 7's j.u.c.Phaser - a short tutorial
<http://javaforu.blogspot.com/2011/08/java-7s-jucphaser-short-tutorial.html>
- Java 7 concurrency utilities: Face the Phaser, intro
<http://olegnatenko.livejournal.com/16771.html>
- Advanced Topics in Programming Languages: The Java Memory...
<http://www.youtube.com/watch?v=IFX4zco0ziY>

Dziękuję!