

---

# 序

當 Sun Microsystem 於 1995 年發表 alpha 版本的 Java 時，全世界的開發者都注意到了。Java 有許多吸引著這些開發者的特質，而不是只有 Sun 用來行銷這程式語言所用的花言巧語而已。Java 不僅是堅固、安全、架構中立、可攜、物件導向、簡單、且 multithreaded。對許多開發者來說，最後兩個字眼似乎是矛盾的：一個 multithreaded 的程式語言怎麼可能會是簡單的？

事實證明至少 Java 的 threading 系統跟其它的 threading 系統相較之下要簡單的多。如此的簡化讓 Java 的 threading 系統很容易學習，甚至對 thread 不熟悉的開發者也較易於著手基本的 thread 程式設計。

在早期的 Java 版本中，簡化也是有利有弊；在其它 threading 系統上可以看到的某些高階功能在 Java 中找不到。Java 2 Standard Edition Version 5.0 (J2SE 5.0) 就不一樣了；它提供了非常多與 thread 相關的新 class 能夠讓撰寫 multithreaded 程式的工作變得簡單許多。

但運用 thread 的程式設計還是一項複雜的工作。本書要對讀者展示如何使用 Java 的 threading 功能來執行基本的 threaded 程式設計工作，以及如何將其擴展至更複雜的程式上以執行更高階的工作。

## 誰該讀這本書？

這本書是寫給想要學習在 Java 程式中使用 thread 之各種經驗程度的程式設計師。這包括了之前就已經使用過 Java 撰寫 threaded 程式的開發者；J2SE 5.0 具有非常多與 thread 相關的新 class 與功能。因此，就算讀者已經寫過 Java 的 threaded 程式，本書還是能夠提供對 Java 新功能的指引以撰寫出更具效益的程式。

本書的前幾章從基本的程度開始起討論 Java 的 threaded 程式設計議題；並未假設開發者已經有很豐富的 threaded 程式設計經驗。隨著章節的進展，不管在資訊的呈現與對開發者要求的經驗上，內容將更為進階。對於剛接觸 threaded 程式設計的開發者來說，這樣的安排應該可以對此議題提供自然的進程。

這本書相當適合著眼於第二波 Java 程式—完全釋放 Java 的 threading 系統威力的複雜程式—之開發者。我們假設本書的讀者已經熟悉 Java 的語法與特質。在某些部分我們會展示出需要對 Java 其他功能具有一定程度了解的複雜程式：例如說 AWT、Swing、NIO、等等。然而，我們所展示的基本原則應該對任何具有基本 Java 知識的人都是可以理解的。我們也發現到討論其它 API 的書都會嚐試對如何在多個 thread 下完整的運用這些 Java 的功能有簡短的說明（反之也是如此；我們並不會試著要解釋與 thread 無關的 Java API）。

雖然本書所展現的內容並沒有設定在需要過往對 thread 的知識，但的確假設讀者已經具有其它領域的 Java API 知識且能夠撰寫簡單的 Java 程式。

## 本書使用的版本

在 Internet 的時代寫本書是很艱苦的事情—我們所依賴的基礎不停的在移動。但我們還是得在基礎上圈起地基，而我們所畫的的這條線就是 Sun Microsystem 的 Java 2 Standard Edition (J2SE) Version 5.0。此軟體之前又被稱為 J2SE Version 1.5。

這個版本之後的 Java 版本很可能會在 threading 系統上面具有一些本書沒有討論到的變化。我們同樣也會在內文的進行中指出 J2SE 5.0 與之前版本的 Java 間相異之處，因此使用較早版本的 Java 開發者同樣也能夠使用本書。

大部分在 J2SE 5.0 中的 threading 新功能對早期版本的 Java來說（包括本書舊版所開發的 class）可透過第三方廠商（以不同的 API）來取得。因此，就算不是使用 J2SE 5.0，你還是可以從本書所涵蓋的題材中得到完整的收穫。

## 本版新增部分

這個版本涵蓋了關於 J2SE 5.0 的資訊。J2SE 5.0 中最重要的改變是引入了 Java Specification Request (JSR) 166，此規格通常又稱為“concurrency utilities（並行功能）”。JSR-166 對已經存在的 API 以及一整組新提供的 API 定義了一群與 thread 相關的加強。

這些新的 API 包括了：

### **Atomic variable**

一組提供 thread 安全性操作而不具 synchronization 的 class。

### **Explicit lock**

可程式化取得與釋放的 synchronization lock。

### **Condition variable**

當特定條件出現時可做為被標定的 notification 主題之變數。

### **Queue**

具 thread 感知的 collection class。

### **Synchronization primitive**

執行複雜類型 synchronization 的新 class。

### **Thread pool**

可管理 thread 的 pool 以執行特定 task 的 class。

### **Thread scheduler**

可在特定的時間點執行 task 的 class。

我們已經在此版的文字中整合了 J2SE 5.0 的新功能。這些新功能可以分為下列三類：

#### **現有功能的新實作**

此語言一直都有執行 data synchronization 與 thread notification 的能力。然而，這些功能的實作是有些限制的；舉例來說，你可以同步化程式碼區段或者整個 method，但跨 method 與 class 間的同步化需要額外的程式設計。在 J2SE 5.0 中，explicit lock 與 condition variable 能夠讓你在使用這些功能時更具彈性。

這些新的實作並沒有對開發者引進新的概念。無論是使用 J2SE 5.0 的 explicit lock 或更基本的 synchronized 關鍵字，想要撰寫 thread 安全性程式的開發者必須要確保他的資料有正確的同步化。因此，兩者皆在述及 data synchronization 時被拿出來一起討論。提供 thread notification 的 condition variable 也是如此，並會跟著 Java 的 wait() 與 notify() 兩 method 一起討論，還有 queue 也會與 Java 其他的 collection class 一併討論。

#### **重要的 thread 功能**

在某些時間點上，幾乎所有撰寫 threaded 程式的開發者會需要使用像是 pool 或 scheduler 之類的基本 thread 功能；其中有許多也同樣會需要使用高等的 synchronization primitive。對此事實的認知正是驅動 JSR-166 的動力之一——在之

前的 Java 版本當然能夠自行開發 thread 的 pool 與 scheduler。但考慮到 threading 在 Java 平台上的重要性，加入這些基本的功能能夠大幅的提高程式設計師的生產力。

### 極簡的 synchronization 功能

Java 新的 atomic class 提供一種方法讓開發者能夠在有必要時避開 synchronization 來撰寫應用程式。這會讓程式有更高的並行性。

如果你讀過本書之前的版本，應該已經熟悉前兩類的功能。在舊版中，我們開發過自訂的 synchronization class、與 thread pool 等。在那些版本中，我們詳細的解釋自訂的實作是如何的運作，並在數個範例中加以運用。在這個版本中，我們只專注於如何有效的運用這些 class。

在本版中這三種類別的資訊是全新寫作的。執行極簡的 synchronization class 需要從新的 virtual machine 本身中取得支援，並且也不能獨立於這些新變化之外來開發。

## 本書內容編排

以下是本書的內容的概要，其中有十五章與一個附錄：

### 第一章《Thread 導論》

這一章對 thread 的議題作基本的介紹：它為何有用以及我們對它的討論方式。

### 第二章《Thread 的建構與管理》

這一章在解釋 thread 如何運作的基本原則同時展示如何建立 thread 與 runnable 物件。

### 第三章《資料同步化》

這一章在基礎程度上討論 thread 間安全的分享資料－於任意時間點協調某個 thread 能夠存取資料。接下來的四章中會有 thread 間分享資料的議題。

### 第四章《Thread Notification》

這一章討論 thread 間在資料異動時用來互相通訊的基本技巧。這可以讓 thread 間回應資料的異動而不是輪替查詢這些異動。

### 第五章《極簡同步化的技巧》

這一章討論以使用最少的同步化來達成資料安全性的 class 與程式設計方式。

### 第六章《進階同步化議題》

在這一章中，我們會完成對資料分享與同步化的檢視，也包括 deadlock、starvation、與其他 locking class 的檢視。

## 第七章《Thread 與 Swing》

Swing class 並不具有 thread 的安全性。這一章討論 multithreaded 的程式如何完全發揮 Swing 的優點。

## 第八章《Thread 與 Collection Class》

Java 的 collection class 是寫給各種不同的狀況。有些是具有 thread 安全性的而某些是沒有的，J2SE 5.0 有特地為 thread 功能引進新的 collection class。在這一章中我們會將它們列出。

## 第九章《Thread 的排程》

排程是單一 CPU 怎樣選擇 thread 來執行的程序。Thread 的排程比較算是作業系統 (OS) 而不是 Java 程式的事情，這一章會討論此領域中 virtual machine 與 OS 之間的關係。

## 第十章《Thread Pool》

這一章討論 thread pool——一群用來執行非特定 task 之 thread 的集合。我們使用 J2SE 5.0 的 thread pool 實作來討論使用 thread pool 的基本法則。

## 第十一章《Task 的排程》

task 的 scheduler 在未來的某一時間點一次或多次的執行某一 task。這一組 class 包括 timer (Java 從 JDK 1.3 開始就有 timer 的 class) 與 J2SE 5.0 中可見的一般 task scheduler。

## 第十二章《Thread 與 I/O》

處理 I/O 是開發者為何在 Java 中使用 thread 的主要原因之一。在這一章中，我們使用 Java 全部的 threading 功能來展示如何在 multithreaded 程式中有效的處理 I/O。

## 第十三章《各種 Thread 議題》

在這一章中，我們藉由檢視 thread security、thread group、thread stack、與其他議題來完成對 Java 中與 thread 有關功能的討論。

## 第十四章《Thread 的效能》

與 thread 有關功能的效能——特別是 synchronization 建構——是撰寫 multithreaded 程式的關鍵。在這一章中，我們測試各種低階程式設計特徵並探索關於 thread 效能的一些真相與迷思。

## 第十五章《多處理器電腦的平行化迴圈》

在這一章中，我們展示運用多處理器電腦的能力來於平行的迴圈中進行 CPU 密集的運算。

### 附錄《廢退的 Threading 功能》

J2SE 5.0 引進許多與 thread 有關的 class。這些 class 其中有許多與本書舊版中所開發的 class 類似；我們在這一個附錄中列出這些 class 以便幫助還不能升級到 J2SE 5.0 的開發者。

## 本書的印刷體裁

下面是本書使用的印刷慣例：

### 斜體字

表示 URL 或者檔案名稱。

### 定寬字

表示內文中的程式碼範例、method、變數、參數、與關鍵字。

### 定寬粗體字

表示使用者輸入的資料，像是你會在命令列輸入的命令。

## 程式範例

所有在本書中展示的範例都是完整、可以執行的應用程式。然而，有許多的程式僅簡短列出，是因為空間與可讀性的考量緣故。完整的範例可從線上下載，網址是：<http://www.oreilly.com/catalog/jthreads3>。

這本書是用來幫助你完成工作的。一般來說，你可以在你的程式與說明文件中使用本書的程式碼。除非你複製大部分的程式碼否則你無須取得我們的使用許可。舉例來說，使用本書數個程式碼的片段來撰寫程式無須事先取得許可。銷售或散佈歐萊禮叢書隨附的 CD-ROM 則必須先取得許可。引用本書與舉本書範例程式碼來回答問題無需許可。在你的產品說明文件中採用本書一定數量的範例程式碼需要取得許可。

我們感謝但不強制要求註明引文出處。引文出處通常包括了書名、作者、出版商、與 ISBN。例如：“Java Threads, Third Edition, by Scott Oaks and Henry Wong. Copyright 2004 O'Reilly Media, 0-596-00782-5”。

對於使用範例程式碼的許可權有任何疑問或意見請洽 [bookquestion@oreilly.com.tw](mailto:bookquestion@oreilly.com.tw)。

## 批評指教

我們誠摯歡迎讀者來信指教。如果您有任何能夠使本書更好的建議，或是發現了本書內容上的錯誤（錯字、贅字、語意不清、描述錯誤），或是書本實體上的損壞（缺頁、掉頁、錯頁、模糊不清、破損），請與台灣歐萊禮公司聯絡。

歐萊禮股份有限公司台灣分公司

台北市大安區（106）復興南路一段295巷21號一樓

電話：（02）2709-9669 # 16

傳真：（02）2703-8802

網站：<http://www.oreilly.com.tw>

行政：[mail@oreilly.com.tw](mailto:mail@oreilly.com.tw)

技術：[bookquestion@oreilly.com.tw](mailto:bookquestion@oreilly.com.tw)

請以電子郵件的方式與我們聯絡，這會比電話和傳統郵件方便。有興趣為本公司翻譯書籍的眾家高手，可與編輯部聯絡；如果你買到的書有印刷品質上的問題，可以寫信到業務部；若你對書籍內容有疑義，或是發現錯字，請寫信到 [bookquestion@oreilly.com.tw](mailto:bookquestion@oreilly.com.tw)。

O'Reilly 的每一本書都有專屬網頁，你可以在此找到關於本書的相關資訊，包括範例程式的下載、勘誤表與相關資源的連結。

<http://www.oreilly.com/catalog/jthreads3> （原文書網頁）

<http://www.oreilly.com.tw/chinese/java/jthreads3.html> （本書中文網頁）

## 誌謝

讀過前言的讀者都知道，編撰出一本書絕對不只是獨占封面的作者一個人的功勞。我們深深的感激下列這些人的幫助與鼓勵：Michael Loukides，在我們認為這是個重要的主題時相信我們並於創意發想期間給我們指引；David Flanagan，對初稿給予無價的指教；Deb Cameron，有條不紊的編輯凌亂文字；Hong Zhang，對 Windows 的 threading 議題給予我們幫助；以及 Reynold Jabbour、Wendy Talmont、Steve Wilson、與 Tim Cramer 過去六年來在工作上的協助。

最重要的，我們必須感謝各自的家人。James，他給予 Scott 必要的支持與鼓勵才能夠完成這本書（以及處理他不斷分心的狀態），還有 Nini，知道要在靈感來時留給 Henry 自己一些時間，並在其餘的時間給予鼓勵—感謝你們所做的一切！

最後，我們必須要感謝許多對本書舊版本給予無價意見的讀者。我們已經盡力的試著回答他們所提出的每一個疑問。請繼續來信！