

# Kurs-Dokumentation



**Zentrum für Informatik ZFI AG**

**.NET Parallel Programming (NCPG)**

<http://mobile.zfi.ch/NCPG>

Weitere Infos finden Sie unter [mobile.zfi.ch](http://mobile.zfi.ch) oder via Adresse:

Zentrum für Informatik ZFI AG  
Zentralsekretariat  
Rütistrasse 28  
CH-8952 Zürich-Schlieren  
Telefon: 044 732 40 00  
Telefax: 041 530 31 68

Zürich, Basel, Bern, ZÄ¼rich, Schweiz

<b>Titel</b>	<b>.NET Parallel Programming</b>
<b>Untertitel</b>	<b>Multicore- und Multiprozessor-Computer richtig ausnutzen</b>
<b>Einleitung</b>	<p>Parallelität und Nebenläufigkeit gewinnt in heutigen Softwaresystemen zunehmende Bedeutung, sei es zur Beschleunigung der Performance auf Multi-Cores oder zur Realisierung von ausgeklügelten nicht-blockierendem Programmverhalten. Aus diesem Grund bieten moderne Systeme wie das Microsoft .NET Framework ein umfangreiches Spektrum von Features für die parallele und asynchrone Programmierung an.</p> <p>Jedoch stellt die Parallelität den Softwareentwickler gleichzeitig vor anspruchsvolle technische Herausforderungen: Wann und wie lässt sich überhaupt mit dem Einsatz von Threads und Thread Pools die Software beschleunigen? Wie lassen sich die Gefahren von Data Races, Race Conditions, Deadlocks, Livelocks und Starvation verhindern? Welche Software-Architekturen, Entwurfsmuster und Modelle eignen sich für die Parallelisierung und Nebenläufigkeit? Welche spezifischen Probleme und Lösungen gibt es hierbei in .NET? Wie lässt sich die .NET Task Parallel Library optimal einsetzen? Welche Besonderheiten gilt es für das C# async/await-Modell zu beachten?</p> <p>In diesem zweitägigen Kurs wird der effiziente und sichere Einsatz von Parallelität in der Praxis für das neueste Microsoft .NET Framework behandelt. Die neueste Version der Sprache C# wird dabei mit Visual Studio für Programmbeispiele eingesetzt. Die Teilnehmer haben ferner die Gelegenheit, ihr gewonnenes Wissen in praktischen Übungen anzuwenden und zu vertiefen.</p>
<b>Ihr Nutzen</b>	<p>Der Kurs vermittelt einen detaillierten Überblick über die Konzepte der Nebenläufigkeit in .NET sowie deren sicheren und effizienten Anwendung in der Praxis. Der Kurs ist so gestaltet, dass er nicht nur .NET-Standardwissen vermittelt, sondern auch über den Horizont der aktuellen Literatur hinausgeht. So werden auch weitergehende Konzepte und besondere Risiken in .NET diskutiert.</p>
<b>Voraussetzungen</b>	<b>Erfahrung als Software-Entwickler von professioneller Software oder als System/Software-Architekt</b>
<b>Teilnehmerkreis</b>	<p>Der Kurs richtet sich vor allem an Software-Ingenieure und Architekten, die sich mit der Nebenläufigkeit in .NET vertieft auseinandersetzen möchten. Er kann auch für Software-Projektleiter und IT-Verantwortliche dienen, sich einen genaueren Überblick über die Möglichkeiten und Herausforderungen der Nebenläufigkeit in .NET zu verschaffen.</p>
<b>Unterlagen</b>	<b>Praxisorientiertes Studienmaterial</b>
<b>Folgekurse</b>	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Introduction</li>   <li>- Motivation</li>   <li>- Levels of parallelism</li>   <li>- System support for parallelism</li> </ul>

- **Threads basics**
  
- **Thread start and join**
- **Thread lifecycle**
  
  
  
  
  
  
  
  
  
  
- **Critical sections and mutual exclusion**
  
  
  
  
  
  
  
  
  
  
- **shared resources**
- **Critical sections**
- **Lock statement**
  
  
  
  
  
  
  
  
  
  
- **Monitor concept**
  
  
  
  
  
  
  
  
  
  
- **Monitor in .NET**
- **Typical pitfalls**
- **.NET monitor specifics**
  
  
  
  
  
  
  
  
  
  
- **Synchronization primitives**
  
  
  
  
  
  
  
  
  
  
- **Barrier**
- **Count down event**
- **Semaphores**
- **Mutex**

- Reader-writer lock
- Manual and auto reset event
  
- Concurrency errors and correctness
  
- Race conditions
- Data races
- Deadlocks
- Livelocks
- Starvation
- Correctness criteria
  
- Thread pools and task parallelism with TPL
  
- Thread pool concept
- .NET Task Parallel Library (TPL)
- Using tasks
- Internal TPL architecture
  
- Data parallelism with TPL
  
- Parallel invoke and parallel loops
- Partitioning
- PLINQ

- Asynchronous programming
- Asynchronous calls with TPL
- task chaining
- Async & await in C#
- Typical pitfalls and recommendations
  
- Threading and GUI
- UI threading model
- Dispatching events
- Non-blocking UI with async/await

**Beitrag**

Der Teilnehmerbeitrag versteht sich rein netto. Das ZFI ist (gemäss MwSt-Gesetz) nicht Mehrwertsteuerpflichtig und erhebt somit keine MwSt. Bei länger als einen Monat dauernden Lehrgängen ist die Zahlung des Teilnehmerbeitrages in mehreren Raten möglich (pro rata temporis).