

## -Dokumentation



**Zentrum für Informatik ZFI AG**

**Refactoring (SREF)**

<http://www.zfi.ch/SREF>

Weitere Infos finden Sie unter [www.zfi.ch](http://www.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>Refactoring</b>
<b>Untertitel</b>	<b>Die interne Struktur von Programmen via Refactoring Schritt für Schritt verbessern.</b>
<b>Einleitung</b>	<b>Mit "Refactoring" fasst man alle Aktivitäten der Software-Entwicklung zusammen, die den Programmcode bestehender Anwendungen verbessern, ohne die Funktionsweise dieser Programme zu ändern. Bestehende Programme werden in der Regel oft angepasst und umgebaut. Dabei soll der Programmcode nicht immer unverständlicher und schwieriger, sondern einfacher und strukturierter werden. Zudem muss das Design beim Umbau des Programmcodes im Auge behalten werden. Heute sind klare Regeln für ein schrittweises Redesign bekannt. Diese werden allgemein unter dem Begriff "Refactoring" zusammengefasst. Dieses Kurs bietet ein vertiefte Einführung in das Refactoring mit einer Programmiersprache wie Java/C# oder C++. Eine Fallstudie zeigt die Vorteile von Refactoring deutlich auf. Zahlreiche Übungen und Beispiele begleiten diesen Kurs.</b>
<b>Ihr Nutzen</b>	<ul style="list-style-type: none"> <li>- Bestehenden Code auf schlechte Gewohnheiten (Bad Smells) hin prüfen können.</li> <li>- Diesen Code via Refactoring Schritt für Schritt verbessern können.</li> <li>- Die Refactoring Denkweise anhand einer Fallstudie nachvollziehen können.</li> </ul>
<b>Voraussetzungen</b>	<b>Programmiererfahrung mit Java/C# oder C++.</b>
<b>Teilnehmerkreis</b>	<b>Software-Entwickler, welche bestehende Software warten, (Applikations- bzw.) Wartungsverantwortliche, Java/C#/C++-Softwareingenieure, die für das Redesign von grösseren Applikationen verantwortlich sind sowie Klassendesigner und Softwarearchitekten</b>
<b>Unterlagen</b>	<ul style="list-style-type: none"> <li>- Begleitbuch</li> <li>- Tutorial</li> <li>- Intranet Site</li> </ul>
<b>Folgekurse</b>	<ul style="list-style-type: none"> <li>- Java Design Pattern Programmierung, Kurscode: JDPP</li> <li>- C# Design Pattern Programmierung, Kurscode: CSDP</li> <li>- C++ Design Pattern Programmierung, Kurscode: CPDP</li> </ul>
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Einführung</li> <li>- Was ist Refactoring?</li> <li>- Warum Refactoring?</li>   <li>- Ein erstes Beispiel</li> <li>- Refactoring Prinzipien</li> <li>- Wann ist Refactoring angesagt?</li> <li>- Probleme mit Refactoring</li> <li>- Refactoring und Design</li>   <li>- Schritte des Refactoring</li> <li>- Codierung ohne Refactoring</li> <li>- Schlechte Gewohnheiten (Bad Smell)</li> <li>- Einige Beispiele</li> <li>- Methoden Refactoring</li> <li>- Extract Method</li> </ul>

- Inline Method
- ...
  
- Objekt Refactoring
- Move Method
- Move Field
- ...
  
- Data Refactoring
- Self Encapsulate Field
- Replace Data Value with Object
- ...
  
- Ausdrücke vereinfachen
- Decompose Conditional
- Remove Control Flag
- ...
  
- Methodenaufrufe vereinfachen
- Rename Method
- Add Parameter
- ...
  
- Generalisieren
- Pull Up Field
- Pull Up Method
- ...

**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).