

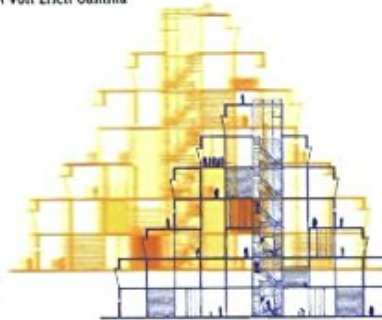
PROFESSIONELLE SOFTWAREENTWICKLUNG Praktik Martin Fowler

Refactoring

Oder: Wie Sie das Design
vorhandener Software
verbessern

Mit Beiträgen von Kent Beck,
John Brant, William Opdyke,
Don Roberts

Vorwort von Erich Gamma



ADDISON-WESLEY

Unüberprüflich geschütztes Material

Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice)



Download



Online Lesen

[Click here](#) if your download doesn't start automatically

Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice)

Martin Fowler

Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) Martin Fowler

 [Download Refactoring . Wie Sie das Design vorhandener Softw ...pdf](#)

 [Online lesen Refactoring . Wie Sie das Design vorhandener Sof ...pdf](#)

Downloaden und kostenlos lesen Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) Martin Fowler

464 Seiten

Amazon.de

Ihre Hausbibliothek reicht Ihnen also, aber könnte sie nicht noch besser sein? *Refactoring* zeigt, wie man mit Refactoring die Erstellung und die Pflege von objektorientiertem Programmcode vereinfachen kann.

Während Refactoring heute noch umfassende Entwurfskenntnisse voraussetzt, dürfte die Verbesserung von Programmcode mit Hilfe von Refactoring-Techniken schon bald keine Seltenheit mehr sein, dann nämlich, wenn erst einmal die entsprechenden Tools auf den Markt kommen. Neben einer Einführung in die Prinzipien des Refactoring bietet dieses Handbuch auch einen Katalog mit zahlreichen Tipps zur Verbesserung von Programmcodes. Das Beste an *Refactoring* ist hierbei der außerordentlich klare Aufbau des Buches sowie die ausgezeichneten praktischen Ratschläge vom Experten auf dem Gebiet der objektorientierten Programmierung: Martin Fowler. Die langjährige Erfahrung des Autors, der außerdem ein international anerkannter Experte auf dem Gebiet von Softwaremustern und UML ist, ist mitverantwortlich für die Qualität dieses Buches, das ein jeder objektorientierter Entwickler mit mehr oder minder ausgereiften Kenntnissen sein Eigen nennen sollte. Wie schon im Falle von Entwurfsmustern hat auch in diesem Buch jeder der Tipps einen einfachen Namen, eine "Motivation", und wird von Beispielen unter Verwendung von Java und UML illustriert. In den ersten Kapiteln wird die Bedeutung von Testdurchläufen bei erfolgreichem Refactoring unterstrichen. (Wer Programmcode verbessert, sollte diesen auch stets testen, um sicherzustellen, dass er auch funktioniert.) Nachdem der Autor erläutert, woran man schlechten Code erkennen kann, geht es ans Eingemachte -- den Katalog von über 70 Refactoring-Tipps zur Verbesserung und Vereinfachung beim Entwurf von Klassen. Neben einer Erklärung werden die Tipps anhand von Beispielcode nach dem Schema "vorher/nachher" -- erläutert. In späteren Kapiteln gewährt der Autor einen kurzen Einblick in den derzeitigen Forschungsstand des Refactoring. Wie im Falle von Softwaremustern ist jetzt vielleicht auch die Zeit für das Refactoring gekommen. Dieses bahnbrechende Buch wird der Idee des Refactoring sicherlich zum Durchbruch unter Programmierern verhelfen. Die eindeutigen Ratschläge auf einem brandneuen Gebiet machen *Refactoring* zu unverzichtbarem Lesestoff für alle diejenigen, die objektorientierte Software schreiben oder mit ihr arbeiten. --*Richard Dragan* Pressestimmen

Die vorgestellten Techniken des Refaktorisierens sind ein Muss, um die Wartbarkeit, Erweiterbarkeit und Robustheit von komplexen Projekten auch in Zukunft zu garantieren. Dieses Buch sollte jeder professionelle Programmierer gelesen haben: Es wird zweifellos Einfluss auf die kommenden Codezeilen haben.

(dotnet.magazin, Dezember 2005) Kurzbeschreibung

Refactoring ist eine Methode, bereits geschriebene Software im Entwurf zu verbessern, ohne ihr Verhalten zu ändern. Das Buch beschreibt die Prinzipien des Refactorings anschaulich anhand eines einführenden Beispiels. Danach wird in Form eines Katalogs auf ca. 70 getestete Refactorings eingegangen. Der Code ist in Java geschrieben, die zugrunde liegenden Prinzipien sind auf alle objektorientierten Sprachen übertragbar. Deutsche Übersetzung von Prof. Dr. Bernd Kahlbrandt.

Download and Read Online Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) Martin Fowler #PXWN6J72RBS

Lesen Sie Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler für online ebook Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler Kostenlose PDF d0wnl0ad, Hörbücher, Bücher zu lesen, gute Bücher zu lesen, billige Bücher, gute Bücher, Online-Bücher, Bücher online, Buchbesprechungen epub, Bücher lesen online, Bücher online zu lesen, Online-Bibliothek, greatbooks zu lesen, PDF Beste Bücher zu lesen, Top-Bücher zu lesen Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler Bücher online zu lesen. Online Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler ebook PDF herunterladen Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler Doc Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler Mobipocket Refactoring . Wie Sie das Design vorhandener Software verbessern (Programmer's Choice) von Martin Fowler EPub