
Patterns konkret



Patterns in Modell-getriebenen Projekten durchgängig anwenden
Mit Delphi, CLX und C# Beispielen

Ivar Jacobson soll einmal gesagt haben, die Zukunft im Softwarebau gehört den sprachunabhängigen Mustern und der Architektur, will meinen, „Executable UML“ ist in grossen Schritten unterwegs, die Grundlagen und Werkzeuge für modellgetriebene Anwendungen bereitzustellen.

Vieles spricht für ein modellbasiertes Vorgehen: Über ein Diagramm kann man besser debattieren als über tausend Zeilen Code, Lösungen werden klarer und Entwurfsfehler lassen sich früher erkennen.

Das Buch richtet sich an alle Entwickler und Architekten, die einen konkreten Einstieg in die Pattern Welt finden möchten. Als Ideenlieferant dient das Buch auch zur Vertiefung bestehender OO-Techniken. Patterns gehören zweifellos zu den nützlichsten und erfolgreichsten Konzepten des Softwarebaus. Aus einer Bibliothek von Modellen und Mustern lässt sich die Zielsprache wählen, die dann auf die entsprechende Plattform portiert und anschliessend generiert wird.

Im ersten Teil des Buches werden die Prozesse und Techniken vorgestellt, die sich für Patterns eignen und zur Verwendung kommen. Es sind dies unter anderem OCL, MDA, Refactoring, Interfaces, Listen, Delegation oder Komposition, einfach das Rüstzeug um erfolgreich Patterns in Schulung oder Projekten einzusetzen.

Es folgen im zweiten Teil die 24 Design Patterns in reich bebildeter Delphi- und C# Notation. Die häufigsten Code Patterns, unter dem Stichwort CODESIGN, zielen auf den Schnelleinsatz ab.

Im dritten Teil runden die 15 Architekturmuster das Buch als Kompendium ab. Die meisten der konkreten Verwendungen beleuchten die Crossplattform CLX, so dass die Beispiele auch für C++Builder oder Kylix Entwickler von Interesse sind. Ein letzter Schwerpunkt bilden die Komponenten einer verteilten Architektur, wie z.B. WebServices oder Provider dies bieten.

Patterns konkret zeigt im Detail, wie Modelle bei ähnlichen Anforderungen immer wieder einsetzbar sind. In der dunklen Nacht angetrieben mit Modellmacht ist es wie die Ruhe vor dem Sturm. So wird man wohl künftig weniger programmieren und vermehrt einen Modellkatalog wie „Patterns konkret“ konsultieren dürfen. Genau sowenig allerdings wie ein Synthesizer das Komponieren guter Musik garantiert, werden Patterns und Tools den Entwurf guter Software garantieren. Doch wie in der Musik wird sich auch im Softwarebau die Erkenntnis durchsetzen, auf die Notation kommt es an.

Max Kleiner
Dr. Silvia Rothen, C#
Dr. Bernhard Angerer, Beirat, Englisch Ausgabe