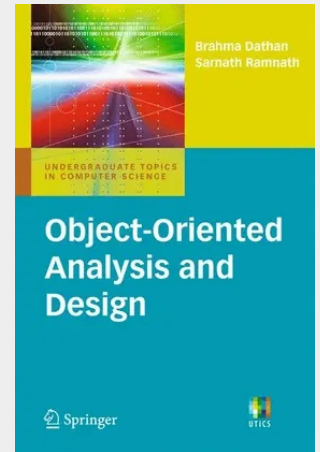


Object-Oriented Analysis and Design

Object-oriented analysis and design (OOAD) has over the years, become a vast field, encompassing such diverse topics as design process and principles, documentation tools, refactoring, and design and architectural patterns. For most students the learning experience is incomplete without implementation. This new textbook provides a comprehensive introduction to OOAD. The salient points of its coverage are: • A sound footing on object-oriented concepts such as classes, objects, interfaces, inheritance, polymorphism, dynamic linking, etc. • A good introduction to the stage of requirements analysis. • Use of UML to document user requirements and design. • An extensive treatment of the design process. • Coverage of implementation issues. • Appropriate use of design and architectural patterns. • Introduction to the art and craft of refactoring. • Pointers to resources that further the reader's knowledge. All the main case-studies used for this book have been implemented by the authors using Java. The text is liberally peppered with snippets of code, which are short and fairly self-explanatory and easy to read. Familiarity with a Java-like syntax and a broad understanding of the structure of Java would be helpful in using the book to its full potential.

Object-oriented analysis and design (OOAD) has over the years, become a vast field, encompassing such diverse topics as design process and principles, documentation tools, refactoring, and design and architectural patterns. For most students the learning experience is incomplete without implementation. This new textbook provides a comprehensive introduction to OOAD. The salient points of its coverage are: • A sound footing on object-oriented concepts such as classes, objects, interfaces, inheritance, polymorphism, dynamic linking, etc. • A good introduction to the stage of requirements analysis. • Use of UML to document user requirements and design. • An extensive treatment of the design process. • Coverage of implementation issues. • Appropriate use of design and architectural patterns. • Introduction to the art and craft of refactoring. • Pointers to resources that further the reader's knowledge. All the main case-studies used for this book have been implemented by the authors using Java. The text is liberally peppered with snippets of code, which are short and fairly self-explanatory and easy to read. Familiarity with a Java-like syntax and a broad understanding of the structure of Java would be helpful in using the book to its full potential. Brahma Dathan is an associate professor in the Department of Information and Computer Sciences at Metropolitan State University, Minnesota. He obtained his BS in engineering with special focus on electronics and communication from the University of Kerala, MTech in computer science from IIT Madras and PhD in computer science from University of Pittsburgh. Sarnath Ramnath received his BTech and MTech degrees from IIT Delhi in 1984 and 1987 respectively, and his PhD in Computer Science from SUNY, Buffalo, in 1994. His areas of interest include algorithm analysis and design, data-structures, computational geometry and object-oriented software design. He is currently Professor and Chair of the Department of Computer Science at Minnesota State University, St Cloud, MN, USA.



37,40 €

34,95 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9781849965217

Medium: Buch

ISBN: 978-1-84996-521-7

Verlag: Springer

Erscheinungstermin: 18.01.2011

Sprache(n): Englisch

Auflage: 1. Auflage. 2011

Serie: Undergraduate Topics in Computer Science

Produktform: Kartoniert

Gewicht: 863 g

Seiten: 440

Format (B x H): 155 x 235 mm

