# DESCRIPTION

NEW LANGUAGE VISUALIZES PROGRAM ABSTRACTIONS CLEARLY AND PRECISELY

Popular software modelling notations visualize implementation minutiae but fail to scale, to capture design abstractions, and to deliver effective tool support. Tailored to overcome these limitations, Codecharts can elegantly model roadmaps and blueprints for Java, C++, and C# programs of any size clearly, precisely, and at any level of abstraction. More practically, significant productivity gains for programmers using tools supporting Codecharts have been demonstrated in controlled experiments.

Hundreds of figures and examples in this book illustrate how Codecharts are used to:

- Visualize the building-blocks of object-oriented design
- Create bird's-eye roadmaps of large programs with minimal symbols and no clutter
- Model blueprints of patterns, frameworks, and other design decisions
Be exactly sure what diagrams claim about programs and reason rigorously about them.

Tools supporting Codecharts are also shown here to:

- Recover design from plain Java and visualize the program's roadmap
- Verify conformance to design decision with a click of a button

**This classroom-tested book includes two main parts:**

**Practice** (Part I) offers experienced programmers, software designers and software engineering students practical tools for representing and communicating object-oriented design. It demonstrates how to model programs, patterns, libraries, and frameworks using examples from JDK, Java 3D, JUnit, JDOM, Enterprise JavaBeans, and the Composite, Iterator, Factory Method, Abstract Factory, and Proxy design patterns.

**Theory** (Part II) offers a mathematical foundation for Codecharts to graduate students and researchers studying software design, modelling, specification, and verification. It defines a formal semantics and a *satisfies* relation for design verification, and uses them to reason about the relations between patterns and programs (e.g., "java.awt implements Composite" and "Factory Method is an abstraction of Iterator").

---

**ABOUT THE AUTHOR**

**AMNON H. EDEN, PhD**, is a computer scientist with the School of Computer Science and Electronic Engineering at the University of Essex and a research fellow at the Center for Inquiry. Dr. Eden has worked as a programmer and consultant to leading software companies, chaired the Software Engineering Diploma Programme at the Tel Aviv College of Management, and served as the associate editor of Minds and Machines. His publications include an entry in the *Stanford Encyclopedia of Philosophy* and articles in leading software engineering journals.

**JONATHAN NICHOLSON, PhD**, earned his doctorate from the School of Computer Science and Electronic Engineering at the University of Essex under the co-supervision of Prof. Raymond Turner and Dr. Eden. His research centres on object-oriented design, with a focus on the development and implementation of the logic underlying the language of Codecharts.
For additional product details, please visit https://www.wiley.com/en-us