

<<面向对象软件工程>>

图书基本信息

书名：<<面向对象软件工程>>

13位ISBN编号：9787302243243

10位ISBN编号：7302243247

出版时间：2011-2

出版时间：清华大学出版社

作者：（美）布鲁格 等著

页数：778

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<面向对象软件工程>>

### 内容概要

本书由B.Bruegge和A.H.Dutoit编写的，是卡耐基-梅隆大学（CMU）高年级本科生和研究生的教材。在第3版本中，作者以循序渐进的方式给出一个完整案例，并把它作为贯穿全书的主线，为读者运用各种软件工程工具创造一个现实环境。所采用的自底向上方法，有助于学生或专业人员循序渐进地学习。本教材向读者展示了如何成功地处理大型复杂软件系统的构建和维护。作者先介绍了开发人员应用软件工程技术所需的基本技能，接下来关注于使开发人员明确、设计并实现复杂系统的方法和技术，最后展示了如何支持贯穿软件生命周期的系统变化，是“软件工程”课程的理想教材。

## <<面向对象软件工程>>

### 作者简介

作者：（美国）布鲁格（Bernd Bruegge）（美国）Allen H.Dutoit

<<面向对象软件工程>>

书籍目录

ForewordPrefaceAcknowledgmentsPART I Getting Started Chapter 1 Introduction to Software Engineering  
Chapter 2 Modeling with UML Chapter 3 Project Organization and CommunicationPART H Dealing with  
Complexity Chapter 4 Requirements Elicitation Chapter 5 Analysis Chapter 6 System Design: Decomposing  
the System Chapter 7 System Design: Addressing Design Goals Chapter 8 Object Design: Reusing Pattern  
Solutions Chapter 9 Object Design: Specifying Interfaces Chapter 10 Mapping Models to Code Chapter 11  
TestingPARTIII Managing Change Chapter 12 Rationale Management Chapter 13 Configuration  
Management Chapter 14 Project Management Chapter 15 Software Life Cycle Chapter 16 Methodologies:  
Putting It All TogetherPART IV Appendices

## 章节摘录

版权页：插图：Today's high-energy physicists are in a position similar to that of a fossil biologist who has found most of the bones. Physicists are building a model of matter and energy and how they fit together at the most basic, subatomic level. Many years of experiments with particle accelerators have given high-energy physicists enough confidence that their models reflect reality and that the remaining pieces that are not yet found will fit into the so-called standard model. This is an example of a model for a system that is claimed to exist. Both system modelers, fossil biologists and high-energy physicists, deal with two types of entities: the real-world system, observed in terms of a set of phenomena, and the application domain model, represented as a set of interdependent concepts. The system in the real world is a dinosaur or subatomic particles. The application domain model is a description of those aspects of the real-world system that are relevant to the problem under consideration. Software engineers face similar challenges as fossil biologists and high-energy physicists. First, software engineers need to understand the environment in which the system has to operate. For a train traffic control system, software engineers need to know train signaling procedures. For a stock trading system, software engineers need to know trading rules. The software engineer does not need to become a fully certified train dispatcher or a stock broker; they only need to learn the application domain concepts that are relevant to the system. In other terms, they need to build a model of the application domain. Second, software engineers need to understand the systems they could build, to evaluate different solutions and trade-offs. Most systems are too complex to be understood by any one person, and most systems are expensive to build. To address these challenges, software engineers describe important aspects of the alternative systems they investigate. In other terms, they need to build a model of the solution domain. Object-oriented methods combine the application domain and solution domain modeling activities into one. The application domain is first modeled as a set of objects and relationships. This model is then used by the system to represent the real-world concepts it manipulates. A train traffic control system includes train objects representing the trains it monitors. A stock trading system includes transaction objects representing the buying and selling of commodities. Then, solution domain concepts are also modeled as objects. The set of lines used to depict a train or a financial transaction are objects that are part of the solution domain. The idea of object-oriented methods is that the solution domain model is a transformation of the application domain model. Developing software translates into the activities necessary to identify and describe a system as a set of models that addresses the end user's problem. We describe in more detail modeling and the concepts of objects in Chapter 2, Modeling with UML.

## <<面向对象软件工程>>

### 编辑推荐

《面向对象软件工程使用UML、模式与Java(第3版)》：大学计算机教育国外著名教材系列

<<面向对象软件工程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>