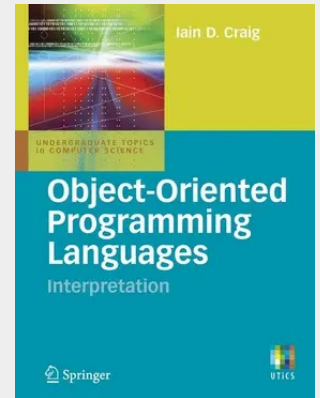


Object-Oriented Programming Languages: Interpretation

This comprehensive examination of the main approaches to object-oriented language explains the key features of the languages in use today. Class-based, prototypes and Actor languages are all looked at and compared in terms of their semantic concepts. In providing such a wide-ranging comparison, this book provides a unique overview of the main approaches to object-oriented languages.

Exercises of varying length, some of which can be extended into mini-projects are included at the end of each chapter. This book can be used as part of courses on Comparative Programming Languages or Programming Language Semantics at Second or Third Year Undergraduate Level. Some understanding of programming language concepts is required.

1.1 Introduction Object-oriented programming has opened a great many perspectives on the concept of software and has been hailed as part of the solution to the so-called 'software crisis'. It has given the possibility that software components can be constructed and reused with considerably more credibility. There are now many case studies in which the reuse of object-oriented components has been made and analysed. Object-oriented programming relates the programming activity to that of modelling or simulation; objects are identified by a correspondence with the objects found in the application area of the program and are used to model those domain operations. Object-oriented programming also opens the prospect of more 'flexible' software that is able to respond dynamically to the needs of the application at runtime. It is very easy to think that object-oriented programming can be performed in only one way. The prevalence of C++ and Java suggests that they are the only way to approach the problem of what an object-oriented programming language should look like. There are many approaches to this way of programming and C++ and Java exemplify just one of these different approaches. Indeed, the way in which the concept of the object is interpreted differs between approaches and between languages. The two main approaches found in object-oriented programming languages are, respectively, class-based and prototype-based languages. Class-based languages are exemplified by Smalltalk [34], C++ [75, 74] and Java [47]. This 2 1. Introduction approach is based upon the identification of common properties of objects and their description in terms of a definitional structure called a class. The objects manipulated by class-based programs are the result of instantiating classes.



37,40 €

34,95 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9781846287732

Medium: Buch

ISBN: 978-1-84628-773-2

Verlag: Springer

Erscheinungstermin: 26.04.2007

Sprache(n): Englisch

Auflage: 1. Auflage. 2007

Serie: Undergraduate Topics in Computer Science

Produktform: Kartoniert

Gewicht: 458 g

Seiten: 256

Format (B x H): 170 x 242 mm

