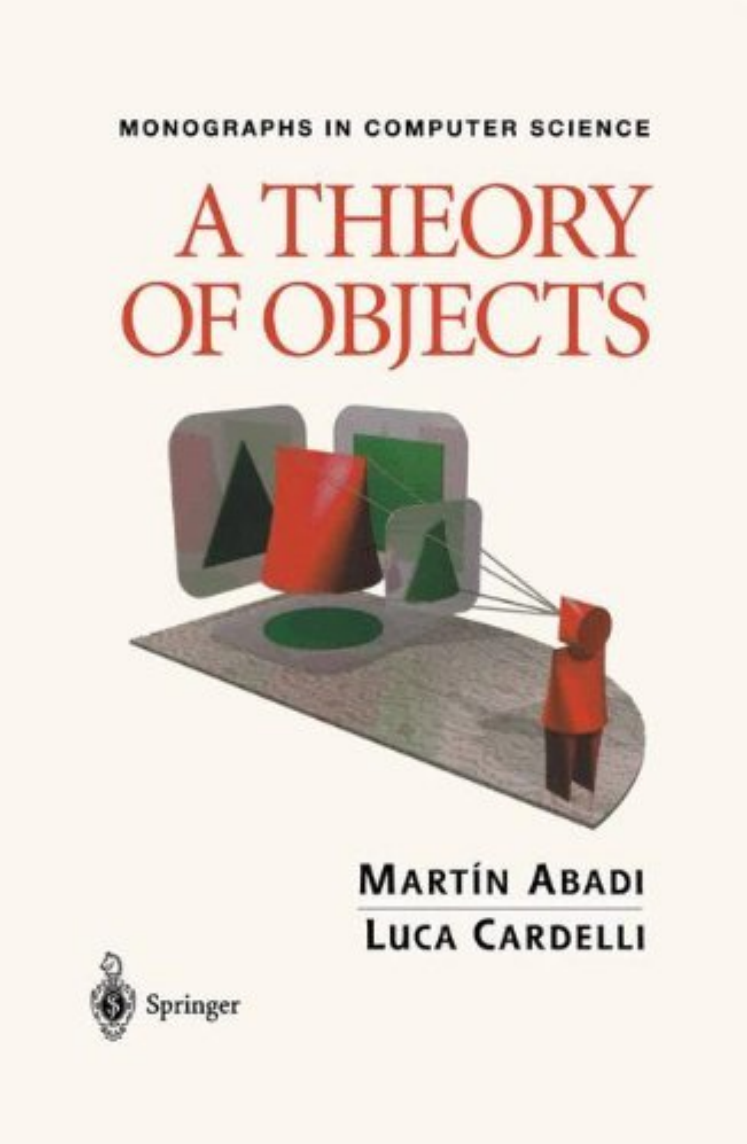
A Theory of Objects (Monographs in Computer Science)

Von Martin Abadi, Luca Cardelli
*audiobook | *ebooks | Download PDF | ePub | DOC*



Produktinformation Verffentlicht am: 2012-09-08Erscheinungsdatum: 2012-09-08File Name: B000PY3NIC | File size: 32.Mb

Von Martin Abadi, Luca Cardelli : A Theory of Objects (Monographs in Computer Science) before purchasing it in order to gage whether or not it would be worth my time, and all praised A Theory of Objects (Monographs in Computer Science):

KurzbeschreibungBy developing object calculi in which objects are treated as primitives, the authors are able to explain both the semantics of objects and their typing rules, and also demonstrate how to develop all of the most important concepts of object-oriented programming languages: self, dynamic dispatch, classes, inheritance, protected and private methods, prototyping, subtyping, covariance and contravariance, and method specialization. An innovative and important approach to the subject for researchers and graduates.KurzbeschreibungBy developing object calculi in which objects are treated as primitives, the authors are able to explain both the semantics of objects and their typing rules, and also demonstrate how to develop all of the most important concepts of object-oriented programming languages: self, dynamic dispatch, classes, inheritance, protected and private methods, prototyping, subtyping, covariance and contravariance, and method specialization. An innovative and important approach to the subject for researchers and graduates.Synopsis Procedural languages are generally well understood and their formal foundations cast in the forms of various lambda-calculi. For object- oriented languages however the situation is not as clear-cut. In this book, the authors propose and develop a different approach by developing object calculi in which objects are treated as primitives. Using object calculi, the authors are able to explain both the semantics of objects and their typing rules and demonstrate how to develop all of the most important concepts of object-oriented programming languages: self, dynamic dispatch, classes, inheritance, protected and private methods, prototyping, subtyping, covariance and contravariance, and method specialization. Many researchers and graduate students will find this an important development of the underpinnings of object-oriented programming.