

| | |
|-------------------------|---|
| 1. Record Nr. | UNINA9910145794503321 |
| Autore | Muller Peter |
| Titolo | Modular Specification and Verification of Object-Oriented Programs // by Peter Müller |
| Pubbl/distr/stampa | Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2002 |
| ISBN | 3-540-45651-1 |
| Edizione | [1st ed. 2002.] |
| Descrizione fisica | 1 online resource (XIV, 298 p.) |
| Collana | Lecture Notes in Computer Science, , 0302-9743 ; ; 2262 |
| Disciplina | 005.1 |
| Soggetti | Computer programming Programming languages (Electronic computers) Software engineering Computer logic Programming Techniques Programming Languages, Compilers, Interpreters Software Engineering/Programming and Operating Systems Software Engineering Logics and Meanings of Programs |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Based on the author's thesis (doctoral)--Fernuniversitat Hagen, 2001. |
| Nota di bibliografia | Includes bibliographical references and index. |
| Nota di contenuto | Mojave and the Universe Type System -- The Semantics of Mojave -- Modular Specification and Verification of Functional Behavior -- Modular Specification and Verification of Frame Properties -- Modular Specification and Verification of Type Invariants -- Conclusion -- Formal Background and Notations -- Predefined Type Declarations -- Examples -- Auxiliary Lemmas, Proofs, and Models. |
| Sommario/riassunto | Software systems play an increasingly important role in modern societies. Smart cards for personal identification, e-banking, software-controlled medical tools, airbags in cars, and autopilots for aircraft control are only some examples that illustrate how everyday life depends on the good behavior of software. Consequently, techniques and methods for the development of high-quality, dependable software systems are a central research topic in computer science. A fundamental approach to this area is to use formal specification and |

verification. Specification languages allow one to describe the crucial properties of software systems in an abstract, mathematically precise, and implementation-independent way. By formal verification, one can then prove that an implementation really has the desired, specified properties. Although this formal methods approach has been a research topic for more than 30 years, its practical success is still restricted to domains in which development costs are of minor importance. Two aspects are crucial to widen the application area of formal methods: – Formal specification techniques have to be smoothly integrated into the software and program development process. – The techniques have to be applicable to reusable software components. This way, the quality gain can be exploited for more than one system, thereby justifying the higher development costs. Starting from these considerations, Peter Muller " has developed new techniques for the formal specification and verification of object-oriented software. The specification techniques are declarative and implementation-independent. They can be used for object-oriented design and programming.
