1. Record Nr. UNINA9910254844303321

Autore	Sestoft Peter
Titolo	Programming Language Concepts / / by Peter Sestoft
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2017
ISBN	3-319-60789-8
Edizione	[2nd ed. 2017.]
Descrizione fisica	1 online resource (XV, 341 p. 87 illus.)
Collana	Undergraduate Topics in Computer Science, , 1863-7310
Disciplina	005.73
Soggetti	Programming languages (Electronic computers) Data structures (Computer science) Programming Languages, Compilers, Interpreters Data Storage Representation
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Introduction -- Interpreters and Compilers -- From Concrete Syntax to Abstract Syntax -- A First-Order Functional Language -- Higher-Order Functions -- Polymorphic Types -- Imperative Languages -- Compiling Micro-C -- Real-World Abstract Machines -- Garbage Collection -- Continuations -- A Locally Optimizing Compiler -- Compiling Micro-SML -- Real Machine Code -- A Crash Course in F#.
Sommario/riassunto	This book uses a functional programming language (F#) as a metalanguage to present all concepts and examples, and thus has an operational flavour, enabling practical experiments and exercises. It includes basic concepts such as abstract syntax, interpretation, stack machines, compilation, type checking, garbage collection, and real machine code. Also included are more advanced topics on polymorphic types, type inference using unification, co- and contravariant types, continuations, and backwards code generation with on-the-fly peephole optimization. This second edition includes two new chapters. One describes compilation and type checking of a full functional language, tying together the previous chapters. The other describes how to compile a C subset to real (x86) hardware, as a smooth extension of the previously presented compilers.The examples present several interpreters and compilers for toy languages, including

compilers for a small but usable subset of C, abstract machines, a garbage collector, and ML-style polymorphic type inference. Each chapter has exercises. Programming Language Concepts covers practical construction of lexers and parsers, but not regular expressions, automata and grammars, which are well covered already. It discusses the design and technology of Java and C# to strengthen students' understanding of these widely used languages.