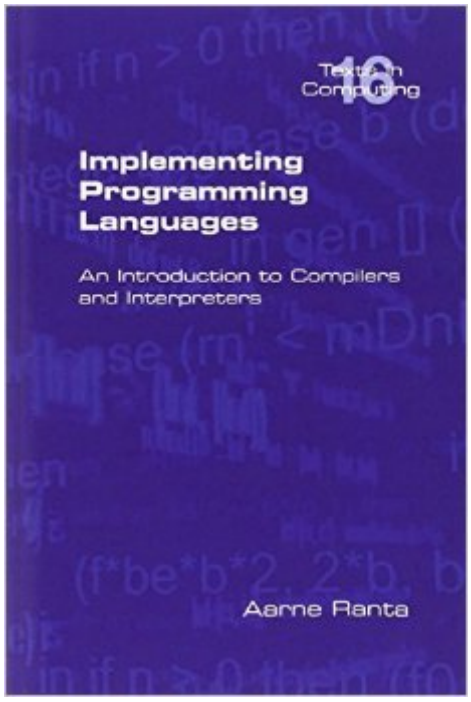
Implementing Programming Languages. An Introduction To Compilers And Interpreters (Texts In Computing)



Synopsis

Implementing a programming language means bridging the gap from the programmer's high-level thinking to the machine's zeros and ones. If this is done in an efficient and reliable way, programmers can concentrate on the actual problems they have to solve, rather than on the details of machines. But understanding the whole chain from languages to machines is still an essential part of the training of any serious programmer. It will result in a more competent programmer, who will moreover be able to develop new languages. A new language is often the best way to solve a problem, and less difficult than it may sound. This book follows a theory-based practical approach, where theoretical models serve as blueprint for actual coding. The reader is guided to build compilers and interpreters in a well-understood and scalable way. The solutions are moreover portable to different implementation languages. Much of the actual code is automatically generated from a grammar of the language, by using the BNF Converter tool. The rest can be written in Haskell or Java, for which the book gives detailed guidance, but with some adaptation also in C, C++, C#, or OCaml, which are supported by the BNF Converter. The main focus of the book is on standard imperative and functional languages: a subset of C++ and a subset of Haskell are the source languages, and Java Virtual Machine is the main target. Simple Intel x86 native code compilation is shown to complete the chain from language to machine. The last chapter leaves the standard paths and explores the space of language design ranging from minimal Turing-complete languages to human-computer interaction in natural language.

Book Information

Series: Texts in Computing
Paperback: 224 pages
Publisher: College Publications (May 9, 2012)
Language: English
ISBN-10: 1848900643
ISBN-13: 978-1848900646
Product Dimensions: 6.1 x 0.5 x 9.2 inches
Shipping Weight: 11.2 ounces (View shipping rates and policies)
Average Customer Review: 4.3 out of 5 starsÂ Â See all reviewsÂ (3 customer reviews)
Best Sellers Rank: #486,451 in Books (See Top 100 in Books) #30 inÂ Books > Computers & Technology > Programming > Languages & Tools > Compiler Design #77 inÂ Books > Computers & Technology > Programming > Languages & Tools > Compilers #982 inÂ Books > Computers &

Technology > Programming > Introductory & Beginning

Customer Reviews

I felt like I needed to write a review on this one and take it down a couple notches unfortunately. I think if this book is used in the classroom, or by someone with at least some experience implementing a programming language, it could be a very good book. However, as a beginner who is trying to use this book for self study I find it very difficult. I'm at the end of chapter 4, at the exercise portion, and I feel that there was simply not enough information presented in the chapter for me to understand how to implement the type checker. Frankly, I feel lost. I emailed the author with a question in the previous exercise and received no response, so if you are using this book, and have trouble understanding something, there are really very few resources available to look for answers. Furthermore, I would absolutely recommend that the reader understand Haskell before purchasing this book. Overall, I can only recommend this book if you already have some experience, or have a person, such as a teacher, who you can talk to when you have questions, otherwise I would recommend looking somewhere else for a book on implementing programming languages.

Good introduction to the major concepts. I like the fact that it actually shows real code (in Haskell no less) implementations, and not just all theory. Was very helpful getting started with the topic.

This book taught me more about grammars, parsing, and compiling than any other resource I've had. It's great for beginners. Thank you Mr. Ranta!!

1620-1860: A Handbook for Interpreters (American Association for State and Local History) The Interpreters Training Manual for Museums Masterminds of Programming: Conversations with the Creators of Major Programming Languages (Theory in Practice (O'Reilly)) Strategic Computing: DARPA and the Quest for Machine Intelligence, 1983-1993 (History of Computing) Dependable Computing for Critical Applications 5 (Dependable Computing and Fault-Tolerant Systems) Wireless Computing in Medicine: From Nano to Cloud with Ethical and Legal Implications (Nature-Inspired Computing Series) Java: The Simple Guide to Learn Java Programming In No Time (Programming,Database, Java for dummies, coding books, java programming) (HTML,Javascript,Programming,Developers,Coding,CSS,PHP) (Volume 2) Introduction to Functional Programming (Prentice Hall International Series in Computing Science) Comparing and Assessing Programming Languages: Ada, C and Pascal (Prentice-Hall software series) Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX (4th Edition) (International Computer Science Series) Inside ATL (Programming Languages/C)