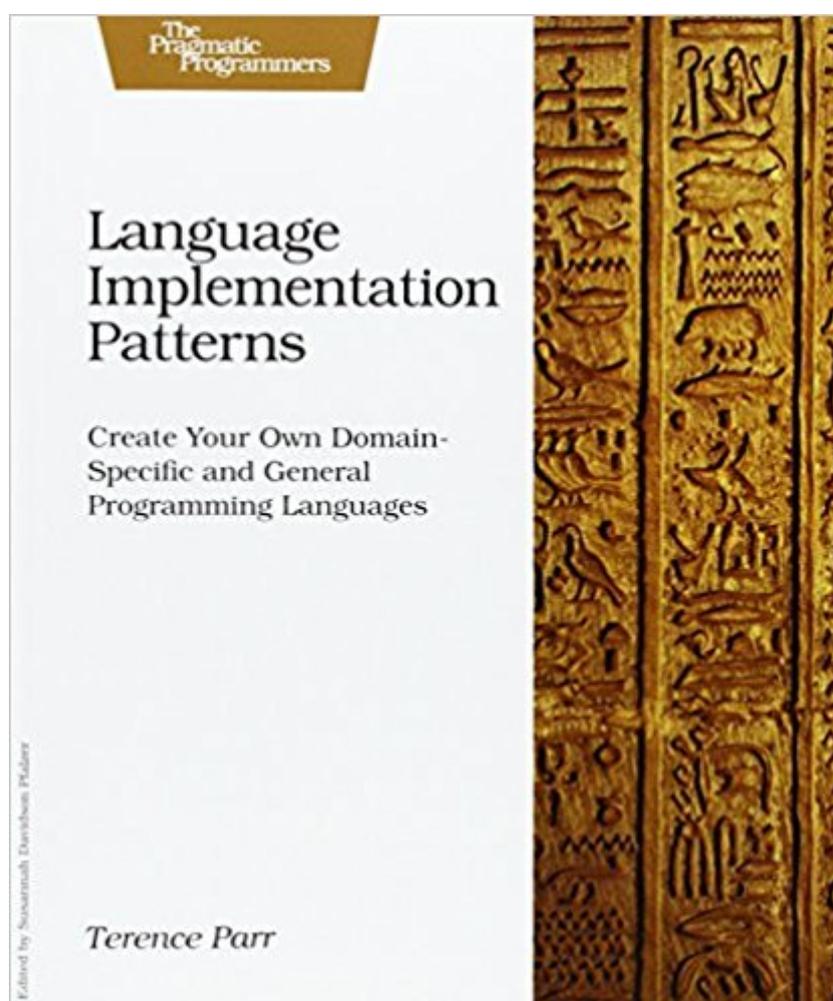


The book was found

Language Implementation Patterns: Create Your Own Domain-Specific And General Programming Languages (Pragmatic Programmers)



Synopsis

Learn to build configuration file readers, data readers, model-driven code generators, source-to-source translators, source analyzers, and interpreters. You don't need a background in computer science--ANTLR creator Terence Parr demystifies language implementation by breaking it down into the most common design patterns. Pattern by pattern, you'll learn the key skills you need to implement your own computer languages. Knowing how to create domain-specific languages (DSLs) can give you a huge productivity boost. Instead of writing code in a general-purpose programming language, you can first build a custom language tailored to make you efficient in a particular domain. The key is understanding the common patterns found across language implementations. Language Design Patterns identifies and condenses the most common design patterns, providing sample implementations of each. The pattern implementations use Java, but the patterns themselves are completely general. Some of the implementations use the well-known ANTLR parser generator, so readers will find this book an excellent source of ANTLR examples as well. But this book will benefit anyone interested in implementing languages, regardless of their tool of choice. Other language implementation books focus on compilers, which you rarely need in your daily life. Instead, Language Design Patterns shows you patterns you can use for all kinds of language applications. You'll learn to create configuration file readers, data readers, model-driven code generators, source-to-source translators, source analyzers, and interpreters. Each chapter groups related design patterns and, in each pattern, you'll get hands-on experience by building a complete sample implementation. By the time you finish the book, you'll know how to solve most common language implementation problems.

Book Information

Series: Pragmatic Programmers

Paperback: 374 pages

Publisher: Pragmatic Bookshelf; 1 edition (January 10, 2010)

Language: English

ISBN-10: 193435645X

ISBN-13: 978-1934356456

Product Dimensions: 7.5 x 0.8 x 9.2 inches

Shipping Weight: 1.4 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars 33 customer reviews

Best Sellers Rank: #176,791 in Books (See Top 100 in Books) #5 in [Books > Computers &](#)

Technology > Programming > Languages & Tools > Compiler Design #15 in [Books](#) >
Computers & Technology > Programming > Languages & Tools > Compilers #36 in [Books](#) >
Computers & Technology > Programming > Microsoft Programming > .NET

Customer Reviews

"Throw away your compiler theory book! Terence Parr shows how to write practical parsers, translators, interpreters, and other language applications using modern tools and design patterns. Whether you're designing your own DSL or mining existing code for bugs or gems, you'll find example code and suggested patterns in this clearly written book about all aspects of parsing technology." • Guido van Rossum, Creator of the Python language "This text is excellent. The exposition plus the examples makes otherwise complex ideas very clear and accessible. Well done!" • Tom Nurkkala, Associate Professor, Computer Science and Engineering, Taylor University

Terence Parr is a professor of computer science and graduate program director at the University of San Francisco, where he continues to work on his ANTLR parser generator (<http://www.antlr.org>) and template engine (<http://www.stringtemplate.org>). Terence has consulted for and held various technical positions at companies such as IBM, Lockheed Missiles and Space, NeXT, and Renault Automation. Terence holds a Ph.D. in computer engineering from Purdue University and was a postdoctoral fellow at the Army High-Performance Computing Research Center at the University of Minnesota, where he built parallelizing FORTRAN source-to-source translators. He is the author of "The Definitive ANTLR Reference": <http://pragprog.com/titles/tpantlr>.

This is a great book for those who want to learn how to write and exploit parsing technology to create DSL (domain specific languages) and transform programs using parsing technology when regular expressions won't do the job or are too complex to be coded easily. The book discusses many aspects of compiler technology and interpreter technology. It will help you write a compiler or interpreter using ANTLR3. It goes in depth into the many things you need to know such as - LL(1), LL(k), and LL(*) parsers and symbol tools. The book instructs one in how to use ANTLR - which is the lexer/parser generator which the author of the book makes freely available. ANTLR is the most powerful generator on the market today and used by many companies such as Oracle. To get the most from this informative book, you should also buy "The Definitive ANLTR 4 Reference" which also sells. Parr is a genius. He has produced the definitive parser generator and the definitive books

that enable one to use this generator. Even if one will not use ANTLR, the books are valuable for their overview and in depth discussion of writing compilers and parsers. Note that the book does not discuss some things necessary for writing a world class compiler such as register allocation. I'm using it to write translators from one language to another and in that application its advice is spot on and extremely helpful.

Very informative, useful, and not math intensive. This is a great addition to my library

This is one of best book on the topic I have read. Not as complete as others, but it's the one I'd recommed to a person interested in the topic as the entry point.

It's an important book for ones who wanna learn more in programming language. Easy to read, and easy to understand

Excellent book. I have been teaching compilers during several years and this book is one of the top modern books. Antlrworks is missed. Hope it will be working with the last version of Antlr.

Good, clear, didactic, complete, useful

Very practical attitude, concisely written :)

When I bought this book I expected a completely theoretical content. It was a pleasant surprise to find a theoretical book, but based on practice.

[Download to continue reading...](#)

Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages (Pragmatic Programmers) 3D Game Programming for Kids: Create Interactive Worlds with JavaScript (Pragmatic Programmers) Practical Programming: An Introduction to Computer Science Using Python 3 (Pragmatic Programmers) Good Math: A Geek's Guide to the Beauty of Numbers, Logic, and Computation (Pragmatic Programmers) Python Programming: Python Programming for Beginners, Python Programming for Intermediates, Python Programming for Advanced C++: The Ultimate Crash Course to Learning the Basics of C++ (C programming, C++ in easy steps, C++ programming, Start coding today) (CSS,C Programming, ... Programming,PHP, Coding, Java Book 1) Ace General Chemistry I and II (The EASY Guide to Ace General Chemistry I

and II): General Chemistry Study Guide, General Chemistry Review The Complete Software Developer's Career Guide: How to Learn Your Next Programming Language, Ace Your Programming Interview, and Land The Coding Job Of Your Dreams Felix Gonzalez-Torres: Specific Objects Without Specific Form Java for COBOL Programmers (2nd Edition) (Programming Series) A Photographer's Guide to Body Language: Harness the power of body language to create stronger, more meaningful portraits and create an experience your clients will rave about The Atlas of Languages: The Origin and Development of Languages Throughout the World (Facts on File Library of Language and Literature)**OUT OF PRINT** WordPress To Go - How To Build A WordPress Website On Your Own Domain, From Scratch, Even If You Are A Complete Beginner C++ and Python Programming: 2 Manuscript Bundle: Introductory Beginners Guide to Learn C++ Programming and Python Programming C++ and Python Programming 2 Bundle Manuscript. Introductory Beginners Guide to Learn C++ Programming and Python Programming Python Programming: The Complete Step By Step Guide to Master Python Programming and Start Coding Today! (Computer Programming Book 4) Internet Empire Profits: Create an Internet Business from Absolute Scratch with Domain Flipping & Associate Marketing Patterns, Principles, and Practices of Domain-Driven Design Raspberry Pi 3: The Ultimate Guide on how to design and build your own projects with Raspberry Pi 3 (Computer Programming, Raspberry Pi 3) (Raspberry Pi ... general,all,new, 2017 updated user guide) Python: Programming: Your Step By Step Guide To Easily Learn Python in 7 Days (Python for Beginners, Python Programming for Beginners, Learn Python, Python Language)

[Contact Us](#)

[DMCA](#)

[Privacy](#)

[FAQ & Help](#)