

## <<编程语言实现模式>>

### 图书基本信息

书名：<<编程语言实现模式>>

13位ISBN编号：9787560977003

10位ISBN编号：7560977006

出版时间：2012-3-20

出版时间：华中科技大学出版社

作者：Terence Parr

页数：388

译者：李袁奎,尧飘海

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<编程语言实现模式>>

### 前言

随着你不断编写语言应用，这个过程中所蕴涵的模式就会逐渐变得清晰而明朗。其实，大多数的语言应用在架构上都是相似的。每次编写语言应用的时候，我都不断告诉自己：“先建立解析器，用它在内存中把数据结构建立起来。然后从中抽取信息，必要时还要改变其结构。最后再写一个能根据这些信息自动输出代码或者报告的工具”。

看吧，这不就是模式？在这些任务中总能发现一些相似的算法和数据结构。一旦掌握了这些语言实现的设计模式或者架构，编写起语言应用来就得心应手了。如果你想快速掌握编写语言应用的能力，这本书正适合你。本书奉行实用主义，从本质上挖掘并提炼语言应用中的设计模式。你会了解模式的重要性，学习如何实现这些模式，如何组合这些模式。很快你就能成为开发语言应用的行家里手！

创造新的语言其实不需要深厚的理论知识做铺垫。你可能不信，毕竟所有语言应用方面的书都会占用大量的篇幅讲解编译器知识。我承认，为通用编程语言编写编译器确实需要扎实的计算机科学知识。然而，大多数程序员并不需要编写这种编译器。因此本书的重心是解决程序员平时最可能遇到的问题：配置文件读取、数据读取、模型驱动的代码生成、源代码之间的翻译、源代码分析和解释器的实现。同理，我们没有使用Scheme等学术界推崇的语言，而是跟随业界的发展采用Java编写所有的示例，以便你能快速地在实际项目中大显身手。

## <<编程语言实现模式>>

### 内容概要

本书旨在传授开发语言应用（工具）的经验和理念，帮助读者构建自己的语言应用。这里的语言应用并非特指用编译器或解释器实现编程语言，而是泛指任何处理、分析、翻译输入文件的程序，比如配置文件读取器、数据读取器、模型驱动的代码生成器、源码到源码的翻译器、源码分析工具、解释器，以及诸如此类的工具。

为此，作者举例讲解已有语言应用的工作机制，拆解、归纳出31种易于理解且常用的设计模式（每种都包括通用数据结构、算法、策略）。

虽然示例是用Java编写的，但相信读者可以触类旁通，利用这些设计模式构建针对其他编程语言（既包括特定领域语言，也包括通用编程语言）的应用。

## <<编程语言实现模式>>

### 作者简介

Terence

Parr是美国旧金山大学的计算机教授、研究生导师，他一直致力于从事ANTLR项目（[antlr.org](http://antlr.org)）和模板引擎（[stringtemplate.org](http://stringtemplate.org)）的设计和开发工作。

Terence曾担任IBM、洛克希德马丁、NeXT、雷诺汽车等公司的技术顾问，另著有《ANTLR权威指南》。

## <<编程语言实现模式>>

### 书籍目录

#### 第1部分 读取输入

##### 第1章 初探语言应用

- 1.1 大局观
- 1.2 模式概览
- 1.3 深入浅出语言应用
- 1.4 为语言应用选择合适的模式

##### 第2章 基本解析模式

- 2.1 识别式子的结构
- 2.2 构建递归下降语法解析器
- 2.3 使用文法DSL来构建语法解析器
- 2.4 词法单元和句子

##### 第3章 高阶解析模式

- 3.1 利用任意多的向前看符号进行解析
- 3.2 记忆式解析
- 3.3 采用语义信息指导解析过程

#### 第2部分 分析输入

##### 第4章 从语法树构建中间表示

- 4.1 为什么要构建树
- 4.2 构建抽象语法树
- 4.3 简要介绍ANTLR
- 4.4 使用ANTLR文法构建AST

##### 第5章 遍历并改写树形结构

- 5.1 遍历树以及访问顺序
- 5.2 封装访问节点的代码
- 5.3 根据文法自动生成访问者
- 5.4 将遍历与匹配解耦

##### 第6章 记录并识别程序中的符号

- 6.1 收集程序实体的信息
- 6.2 根据作用域划分符号
- 6.3 解析符号

##### 第7章 管理数据聚集的符号表

- 7.1 为结构体构建作用域树
- 7.2 为类构建作用域树

##### 第8章 静态类型检查

#### 第3部分 解释执行

##### 第9章 构建高级解释器

- 9.1 高级解释器存储系统的设计
- 9.2 高级解释器中的符号记录
- 9.3 处理指令

##### 第10章 构建字节码解释器

- 10.1 设计字节码解释器
- 10.2 定义汇编语言语法
- 10.3 字节码机器的架构
- 10.4 如何深入

#### 第4部分 生成输出

## <<编程语言实现模式>>

### 第11章 语言的翻译

- 11.1 语法制导的翻译
- 11.2 基于规则的翻译
- 11.3 模型驱动的翻译
- 11.4 创建嵌套的输出模型

### 第12章 使用模板生成DSL

- 12.1 熟悉StringTemplate
- 12.2 StringTemplate的性质
- 12.2 从一个简单的输入模型生成模板
- 12.4 在输入模型不同的情况下复用模板
- 12.5 使用树文法来创建模板
- 12.6 对数据列表使用模板
- 12.7 编写可改变输出结果的翻译器

### 第13章 知识汇总

- 13.1 在蛋白质结构中查找模式
- 13.2 使用脚本构建三维场景
- 13.3 处理XML
- 13.4 读取通用的配置文件
- 13.5 对代码进行微调
- 13.6 为Java添加新的类型
- 13.7 美化源代码
- 13.8 编译为机器码

### 参考文献

### 索引

## <<编程语言实现模式>>

### 媒体关注与评论

别看那些编译原理的书了！

这本书教你编写真正实用的解析器、翻译器、解释器等语言应用，Terence Parr 在书中细致地讲解了先进的语言工具和语言应用中设计模式的用法。

无论是编写自己的领域专用语言（DSL），还是挖掘已有代码、查错或是寻宝，都能从这本简单易懂的书中找到示例和模式，因为它基本上覆盖了解析技术的方方面面。

——Python语言之父Guido Van Rossum；我的“龙书”被打入冷宫了！

——Android平台Dalvik虚拟机的设计者Dan Bornstein 本书对每个语言设计者来说都是一笔无价的财富。

——泰勒大学计算机科学系副教授Tom Nurkkala博士 Terence清晰地阐释了语言设计中的概念。

如果你想独创一门语言却又无从下手，或者觉得它高不可攀，那么，从这本书开始吧。

——Adam Keys 这本书行文风格浅显却又不失韵味，以这个经久不衰的热门话题为中心，娓娓道来，颇有大师风范。

《编程语言实现模式》不光讲述创造语言的方法，还指引我们在这个过程中该思考些什么。

要想创建一个强壮的、可维护的专用语言，这本书是无价之宝。

——Breaulty研究机构科学软件开发部门主管Kyle Ferrio博士

## <<编程语言实现模式>>

### 编辑推荐

虽然《编程语言实现模式》不专门讨论如何设计编程语言，但读者在阅读的过程中将吸收丰富的相关知识。

比较不同编程语言的特点、了解编程语言的发展历史是学习设计编程语言的好途径。

<<编程语言实现模式>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>