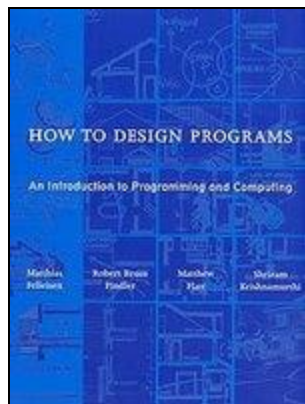


《程序设计方法》



分类	IT 类 / 软件开发 / 编程语言 / Lisp
作者	(美) Matthias Felleisen / 马蒂亚斯·费雷森 (美) Robert Bruce Findler / 罗伯特·布鲁斯·芬德勒 (美) Matthew Flatt / 马修·弗拉特 (美) Shriram Krishnamurthi / 施拉姆·克里斯纳默西
英文书名	《How to Design Programs》
知名缩写	HtDP
出版年份	2001(第1版, 原著) 2003(第1版, 简体中译本) 2018(第2版, 原著) 2020(第2版, 简体中译本)
相关链接	本书官网 How to Design Programs @ Wikipedia 程序设计方法 @ 维基百科 英文在线阅读(第2版) 英文在线阅读(第1版)

简介

本书的4个作者都是美国高校的教授。第一作者 Felleisen 也是《The Little Schemer》的作者。

此书在编程界颇有名气，其知名缩写是【HtDP】。

它通常被拿来与 [SICP](#) 作对比——这两本书都堪称经典，都基于 [Scheme](#) (Lisp 的方言) 进行编程教学；都被众多欧美高校采用。

HtDP 的名气不如 SICP——主要因为问世较晚(第1版完成于2001年)。

虽然名气不如, 但 HtDP 的门槛(学习曲线) 比 SICP 要【低很多】——更容易上手。
友情提示: 如果你读 SICP 觉得吃力, 可以改看这本。

中文目录(第1版)

前言

第一部分 简单数据的处理

- 第1章 学生、教师和计算机
- 第2章 数、表达式和简单程序
- 第3章 程序就是函数加上变量定义
- 第4章 条件表达式和函数
- 第5章 符号信息
- 第6章 复合数据之一——结构体
- 第7章 数据的多样性
- 第8章 语法和语义

第二部分 任意数目数据的处理

- 第9章 复合数据类型之二——表
- 第10章 表的进一步处理
- 第11章 自然数
- 第12章 三论函数复合
- 第13章 用list构造表

第三部分 再论任意大数据的处理

- 第14章 再论自引用数据定义
- 第15章 相互引用的数据定义
- 第16章 反复精化设计
- 第17章 处理两种复杂数据片段
- 第18章 局部定义和辖域

第四部分 抽象设计

- 第19章 定义的相似性
- 第20章 函数也是值
- 第21章 抽象设计的例子
- 第22章 使用函数进行抽象设计
- 第23章 数学方面的例子
- 第24章 定义匿名函数

第五部分 生成递归

- 第25章 一种新的递归形式
- 第26章 设计算法
- 第27章 主题的变更

第28章 回溯算法

第29章 计算的代价和向量

第六部分 知识累积

第30章 知识的丢失

第31章 设计带累积器的函数

第32章 使用累积器的更多例子

第33章 非精确数的本质

第七部分 改变变量的状态

第34章 函数的记忆

第35章 对变量赋值

第36章 设计有记忆的函数

第37章 使用记忆的例子

第38章 最终的语法和语义

第八部分 复合值的改变

第39章 封装

第40章 可改变的结构体

第41章 设计改变结构体的函数

第42章 相等

第43章 修改结构体、向量和对象

结束语

英文目录(第1版)

Preface

Part I Processing Simple Forms of Data

Chapter 1 Students, Teachers, and Computers

Chapter 2 Numbers, Expressions, Simple Programs

Chapter 3 Programs are Function Plus Variable Definitions

Chapter 4 Conditional Expressions and Functions

Chapter 5 Symbolic Information

Chapter 6 Compound Data, Part 1: Structures

Chapter 7 The Varieties of Data

Chapter 8 Intermezzo 1: Syntax and Semantics

Part II Processing Arbitrarily Large Data

Chapter 9 Compound Data, Part 2: Lists

Chapter 10 More on Processing Lists

Chapter 11 Natural Numbers

Chapter 12 Composing Functions, Revisited Again

Chapter 13 Intermezzo 2: List Abbreviations

Part III More on Processing Arbitrarily Large Data

Chapter 14 More Self-referential Data Definitions

Chapter 15 Mutually Referential Data Definitions

Chapter 16 Development through Iterative Refinement

Chapter 17 Processing Two Complex Pieces of Data

Chapter 18 Intermezzo 3: Local Definitions and Lexical Scope

Chapter IV Abstracting Designs

Chapter 19 Similarities in Definitions

Chapter 20 Functions are Values

Chapter 21 Designing Abstractions from Examples

Chapter 22 Designing Abstractions with First-Class Functions

Chapter 23 Mathematical Examples

Chapter 24 Intermezzo 4: Defining Functions on the Fly

Chapter 25 A New Form of Recursion

Chapter 26 Designing Algorithms

Chapter 27 Variations on a Theme

Chapter 28 Algorithms that Backtrack

Chapter 29 Intermezzo 5: The Cost of Computing and Vectors

Chapter 30 The Loss of Knowledge

Chapter 31 Designing Accumulator-Style Functions

Chapter 32 More Uses of Accumulation

Chapter 33 Intermezzo 6: The Nature of Inexact Numbers

Chapter 34 Memory for Functions

Chapter 35 Assignment to Variables

Chapter 36 Designing Functions with Memory

Chapter 37 Examples of Memory Usage

Chapter 38 Intermezzo 7: The Final Syntax and Semantics

Chapter 39 Encapsulation

Chapter 40 Mutable Structures

Chapter 41 Designing Functions that Change Structures

Chapter 42 Equality

Chapter 43 Changing Structures, Vectors, and Objects

Epilogue

Index

[【编程随想】收藏的电子书清单](#)

[【编程随想】的博客](#)