

Table of contents

- Preface to the Third Edition
- Preface
- Concept Directory
- History, Motivation, and Evaluation
- Design and Implementation
- Principles
- Implementation
- Introduction
- 1 The Beginning: Pseudo-Code Interpreters
 - 1.1 History and Motivation
 - 1.2 Design of a Pseudo-Code
 - 1.3 Implementation
 - 1.4 Phenomenology of Programming Languages
 - 1.5 Evaluation and Epilog
- 2 Emphasis on Efficiency: Fortran
 - 2.1 History and Motivation
 - 2.2 Design: Structural Organization
 - 2.3 Design: Control Structures
 - 2.4 Design: Data Structures
 - 2.5 Design: Name Structures
 - 2.6 Design: Syntactic Structures
 - 2.7 Evaluation and Epilog
- 3 Generality and Hierarchy: Algol-60
 - 3.1 History and Motivation
 - 3.2 Design: Structural Organization
 - 3.3 Design: Name Structures
 - 3.4 Design: Data Structures
 - 3.5 Design: Control Structures
- 4 Syntax and Elegance: Algol-60
 - 4.1 Design: Syntactic Structures
 - 4.2 Descriptive Tools: BNF
 - 4.3 Design: Elegance
 - 4.4 Evaluation and Epilog
- 5 Return to Simplicity: Pascal
 - 5.1 History and Motivation
 - 5.2 Design: Structural Organization
 - 5.3 Design: Data Structures
 - 5.4 Design: Name Structures
 - 5.5 Design: Control Structures
 - 5.6 Evaluation and Epilog
- 6 Implementation of Block-Structured Languages
 - 6.1 Activation Records and Context
 - 6.2 Procedure Call and Return
 - 6.3 Display Method

- 6.4 Blocks
- 6.5 Summary
- 7 Modularity and Data Abstraction: ADA
- 7.1 History and Motivation
- 7.2 Design: Structural Organization
- 7.3 Design: Data Structures and Typing
- 7.4 Design: Name Structures
- 8 Procedures and Concurrency: ADA
- 8.1 Design: Control Structures
- 8.2 Design: Syntactic Structures
- 8.3 Evaluation and Epilog
- 9 List Processing: LISP
- 9.1 History and Motivation
- 9.2 Design: Structural Organization
- 9.3 Design: Data Structures
- 10 Functional Programming: LISP
- 10.1 Design: Control Structures
- 10.2 Design: Name Structures
- 10.3 Design: Syntactic Structures
- 11 Implementation of Recursive List-Processors: LISP
- 11.1 Recursive Interpreters
- 11.2 Storage Reclamation
- 11.3 Evaluation and Epilog
- 12 Object-Oriented Programming: Smalltalk
- 12.1 History and Motivation
- 12.2 Design: Structural Organization
- 12.3 Design: Classes and Subclasses
- 12.4 Design: Objects and Message Sending
- 12.5 Implementation: Classes and Objects
- 12.6 Design: Object-Oriented Extensions
- 12.7 Evaluation and Epilog
- 13 Logic Programming: Prolog
- 13.1 History and Motivation
- 13.2 Design: Structural Organization
- 13.3 Design: Data Structures
- 13.4 Design: Control Structures
- 13.5 Evaluation and Epilog
- 14 Principles of Language Design
- 14.1 General Remarks
- 14.2 Principles
- Each chapter ends with Exercises
- Bibliography
- Index