<<编程语言>>

图书基本信息

书名：<<编程语言>>

13位ISBN编号：9787302198062

10位ISBN编号：7302198063

出版时间：2009-5

出版时间：塔克、 努南 清华大学出版社 (2009-05出版)

作者：Allen Tucker,Robert Noonan

页数：590

前言

自本书第1版于1999年出版以来，编程语言的研究已得到迅猛发展。
例如，从CSl开始，Java已经成为计算机科学课程的一门重要语言。
敏捷编程与软件设计如影随形，且其语言习惯有别于传统的程序。
用正规的方法进行软件设计已渐入主流，且作用显著。
有鉴于此，新版希望能够适应在当前和未来编程语言设计过程中所伴随的激励和新挑战。
例如，新版对全部4种程序设计范例及其所用的语言在广度和深度上都有进一步的提高，如表O一1所示。
新版第二个主要的变化是大大丰富了第2-11章关于语言设计原理的内容。
我们使用不太正式的描述风格，增加了新的来自流行编程语言（如Python和Perl）的例子。
此外，对于已经不再广泛应用的语言（如Pascal和Modula），新版多已略去。
第2、4、5、7、9章主要讲述编程语言的核心原理——语法、名称、类型、语义和函数。
这几章与第1版相比较，为上机学习这些原理提供了更广泛、更深入的语言和范例。
如果您希望了解语法、类型系统、语义、函数和存储管理的具体实现原理，可以在第3、6、8、10和11章中找到这些材料。
读者可以有选择地学习这几章，以丰富相关的核心原理。
例如，通过学习第3章中编译器的词汇和语法段，可加深和巩固对第2童的学习。
注意.可以跳过部分或全部的章节。

<<编程语言>>

内容概要

《编程语言：原理与范型（第2版）》以C、Java、Perl和Python编程语言为范型，介绍了编程语言的原理与设计。
全书共有18章，第2～11章介绍了关于语言设计原理的内容，主要讲述编程语言的核心原理：语法、名称、类型、语义和函数等。
第12～15章介绍了命令式编程、面向对象编程、函数式编程和逻辑式编程等。
第16～18章详细介绍了事件处理、并发性和程序正确性。
为了提高读者的上机学习能力，《编程语言：原理与范型（第2版）》为这些原理提供了丰富的应用范例。

作者简介

作者：(美国)塔克 努南

书籍目录

章节摘录

插图：Definition：A context-free grammar has a set of productions P，a set of terminal symbols T and a set of nonterminal symbols N，one of which，S，is distinguished as the start symbo1.Definition：AgrammarproductionhastheformA一∞whereAisanonterminal symbol and cois a stringofnonterminalandterminal symbols.One form of context-free grammar,called Backus-Naur Form（BNF for short）hasbeen widely used to define the syntax of programming languages.2.1.1 Backus-Naur Form（BNF）GrammarsIn 1960，BNF was adapted from Chomsky,s theory by John Backus and Peter Naur to express a formal syntactic definition for the programming language Algol[Naur（ed.），1963].Like many texts, we use the term BNF grammar as a synonym for context free grammar.For a more complete discussion of the differences between the two，see Observation（P.26）.Aproductionis aruleforrewritingthatcanbeappliedtoa stringofsymbolscalleda sententialform.Aproduction is interpreted as follows：the nonterminalA can be replaced by in a sententialform.The symbol A is often called the left-handside，whilethe string is called the right.hand side of the production.In BNF grammars，the sets of terminaland nonterminal symbols are disjoint. When a BNF grammar is used for defining programming language syntax，the nonterminals N identify the language ’s grammatical categories like Identifier，Inte。
ger，Expression，Statement，and Program.The start symbol S identifies the principal grammatical category being defined by the grammar（typically Program），and is usually defined in the first production.The terminal symbols T form the basic alphabet from which programs are constructed.To illustrate these ideas，here is a pair of productions that defines the syntax of the grammatical category binaryDigit：binaryDigit-0binaryDigit-1This pair defines a binaryDigit as either 0 or 1，but nothing else.The nonterminal sym-bols are all the symbols that appear on the left-hand side of at least one production.For the above grammar,binaryDigit is the only nonterminal.The terminal symbols are a11 the other symbols that appear in the productions；for the above grammar,0 and 1 arethe terminal symbols.When a series ofproductions all have the same nonterminal symbol on their left-hand sides，they may be combined into a single production.For example，the above two productions can be abbreviated by：binaryDigit一0In this case.the alternatives are separated by a vertical bar（I），which literally means “ or，”so the interpretation remains the same as the original pair of productions.In this example，both the fight arrow and vertical bar are metasymbols，whichare symbols that are part of the metalanguage and are not part of the language being defined.

编辑推荐

《编程语言:原理与范型(第2版)》是由清华大学出版社出版的。

版权说明