

<<计算机英语>>

图书基本信息

书名：<<计算机英语>>

13位ISBN编号：9787811372830

10位ISBN编号：7811372835

出版时间：2009-7

出版时间：苏州大学出版社

作者：张国安 编

页数：288

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>

<<计算机英语>>

内容概要

本书力图向读者介绍计算机科学的全貌，即在计算机环境下如何解决问题和有效地完成任务。由于计算机知识的多样性，本书无法只介绍其中某一方面的问题，而是尽可能地描述计算机科学的发展，读者从中可以看清计算机发展的脉络。

本书覆盖了离散数学、算法、编程方法学、面向对象程序设计、操作系统、科学计算这样一些传统意义上的计算机科学理论，也涉及到计算机图形学、多媒体、人工智能、计算机网络这些新议题，更加进了述录系统、语音技术、网络安全、应用程序、微控制器这样一些应用实例。

这样的编排是为了拓展读者的视野，从中发现自己感兴趣的问题。

与其他计算机英语教科书不同的是，本书是按照简单的全反应式模型来编写的，与其说这是一本英语书，不如说这是用英语编写的介绍计算机科学的引论。

本书不再介绍英语语法，但教师还是应当对复杂的语法现象做深入分析，这既是理解上下文的需要，也是提高英语阅读能力所必须的。

从学习者的角度来看，这使解释更清晰，读者把握内容更全面。

从教学的角度看，这样的做法为教师提供了一种自然的循序渐进的教学方法，从而讲授那些难于掌握的材料。

本书分成20个单元，每个单元包含3篇文章，Text A和Text B给出了一些关键词汇和短语，对较为深入的问题作了注解，并安排少量问题。

读者除了用英文完成那些问题外，还应自主查找相关资料。

这些问题有些涉及计算机的专门知识，有些是为了提高英语应用水平。

使用本书并不需要计算机科学基本知识以外的其他预先知识。

当然，已经学了一些计算机基础知识会有助于理解本书中一些实际应用方面的材料。

书籍目录

Unit 1 Computer Science Unit 2 Discrete Mathematics for Computer Science Unit 3 Algorithms in the “ Real World ” Unit 4 Dictation Systems Unit 5 Programming Methodology Unit 6 Coding Guidelines Unit 7 Object-Oriented Systems Design Unit 8 Operating Systems and Systems Pmgmmim Unit 9 introduction to ScientiDc Computing Unit 10 lutmduction to Computer System Unit 11 Speech Technology Research, Development, and Oeplotmlem Unit 12 Computer Language and Programming Unit 13 Computer Graphics and image Unit 14 Multimedia Technology Unit 15 Research Project in Artificial intelligence Unit 16 SoRwm That Inables You to, fhor ImelliUem Tm~r Behavior Unit 17 Computer Network TechnologY Unit 18 COmPuter and NetWO Secuy Unit 19 Compmor Application Unit 20 Foundation and Application of Microcontroller

章节摘录

So far, we've been looking at various CPU instructions and how to represent them in memory. Basically, we've been looking at the machine language of the 68000. That is, the language that the 68000 inherently understands. This is all fine and dandy if you're actually setting the bits in a computer's memory by hand. However, more often than not, you don't end up programming this way. In order to make it easier to program computers, someone came up with the bright idea of creating an assembler. An assembler is able to take machine instructions in a human-readable form and translate them into binary automatically. That's about a million times better than machine language. However, it's quite easy to do. An assembly language is simply a syntax for specifying machine instructions. That means that you can convert from assembly language to machine language with just a simple encoding of the machine language instructions. In fact, by definition all assembly language instructions have a one-to-one mapping to machine language. In other words, the assembler doesn't need any smarts—it just needs to know the translation. Having assembly languages is great, but there is still something missing. It's still difficult to get things done in assembly, plus every processor has a different assembly language (they have to because they all have different machine languages). That means that if you write some code for one processor, you have to completely re-write it if you change processors. Over time, people developed higher-level languages, such as Fortran, C, Pascal, Java, Visual Basic. You should be familiar with at least one of these languages (namely C, since it's a prerequisite for the class). These higher-level languages are great because they make programming easier and also allow for some portability between processors. One interesting point is that in higher-level languages, one statement can translate into numerous assembly language instructions.

编辑推荐

中国大学英语教学改革的方向，中国社会经济发展的必然要求。

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>