

<<实用C语言程序设计教程>>

图书基本信息

书名：<<实用C语言程序设计教程>>

13位ISBN编号：9787302207696

10位ISBN编号：7302207690

出版时间：2009-12

出版时间：清华大学出版社

作者：孟朝霞 编

页数：413

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>

<<实用C语言程序设计教程>>

前言

随着计算机产业的迅速发展, 各行各业对计算机应用人才需求日益迫切。如何使程序设计学习者顺利地进入程序设计的大门, 如何熟悉和精通程序设计, 是非计算机专业教学的难题。

程序设计既是一门科学, 也是一门艺术。

要掌握程序设计的开发艺术, 必须掌握程序设计语言。

C语言以其灵活性和实用性受到了广大计算机程序设计人员的喜爱, 它可以开发系统软件和应用软件, 是软件开发领域中广泛应用的语言, 也是高校计算机语言类课程的首选。

根据高等学校本科教学工作的指导思想, 着眼于国家发展和人的全面发展需要, 本书注重能力培养, 着力于提高学习者的学习能力、实践能力和创新能力, 全面推进素质教育, 以人才培养为根本任务, 致力于培养开拓创新、适应社会发展的合格人才。

本书旨在讲授程序设计基础和C语言基础, 突出C语言课程本身实践性强的特点, 通过应用案例和项目案例讲解, 以倡导启发式教学和研究性学习为核心, 激发学习者的兴趣和潜能, 注重学习者思考能力和创新能力的培养, 从重视知识目标转向重视智能目标。

本书“从零开始”, 在内容组织上循序渐进, 在结构上做了精心安排。

全书共8章, 分为初级篇、中级篇和高级篇, 应用内容嵌入各章。

初级篇介绍了C语言基本数据、基本结构以及解决实际问题的基本步骤, 引入了数据文件; 中级篇介绍了用函数进行模块化程序设计的方法、变量作用域和存储特性及编译预处理; 高级篇系统阐述了C语言构造数据类型, 描述了动态数据结构。

本书由作者在多年教学与程序实践的基础上, 结合多次编写相关讲义和教材的经验总结而成。

本书由孟朝霞主编, 其中第1章, 第5章由杨立编写, 第2章, 第6章由李霞编写, 第3章, 第4章由王琴竹编写, 第7章, 第8章由孟朝霞编写。

为了便于教学, 每章基本上按照以下结构进行安排。

<<实用C语言程序设计教程>>

内容概要

本书旨在培养非计算机专业学生使用计算机解决问题的能力，这些问题包括从计算简单函数到解非线性方程，再至较复杂的事务处理。

本书避免过分研究语法细节，注重通过案例和项目培养程序设计能力，内容侧重点安排恰当，强调程序设计及风格，充分体现应用型人才培养的目标要求，本书分为初级篇（必选）、中级篇（有选择安排）和高级篇（建议辅导、自学结合）。

本书基于Visual C++ 6.0集成开发环境，每章配有编程练习和实验，教学中最好配合以小组学习法。教材结合各种实际工程问题，精心设计应用案例和项目开发任务，把“语句(Statement)、代码(Code)、编程(Programming)、应用(Application)”教学目标融入课程，使编程成为经验体验和创新乐趣的结合。

本书强调用计算机求解问题的方法论，注重对工程和科学问题的求解，重点介绍如何结合现实工程和科学应用的示例与问题。

本书可作为高等院校和职业技术学校非计算机专业的计算机程序设计教学用书，也可作为从事计算机应用的科技人员的参考书或培训教材（目录中标注“*”的为选修章节）。

<<实用C语言程序设计教程>>

书籍目录

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|----------------------|---------------|---------------|-------------|-----------|----------------|--------------|---------------|--------------|---------------|---------------|-------------|--------------------|--------------------|------------|-------------------|----------------|-------------|----------|-------------|----------|----------------|---------------|--------------|---------------|---------------|---------------------|-----------------|----------------------|-------------------|-------------------|-------------------|---------------------|-------------|------------|-------------|--------------|----------------|--------------|-----------------|-----------------------|-------------------|---------------------|----------------------|---------------------|-------------------|------------------------------|-----------|---------------|----------|-------------------|-----------------|--------------|-------------------|-------------------|------------------|--------------|--------------------------------------|---------------|-------------------------------|------------|------------------------------|------------|-----------------|---------------|------------|----------|-------------|---------------|---------------|--------------|------------------|----------|-----------------|--------------------|--------|---------------------|--------------|--------|
| 初级篇 第1章 C语言学习基础 | 1.1 预备知识：计算机系统的硬件与软件 | 1.1.1 计算机硬件系统 | 1.1.2 计算机软件系统 | 1.1.3 计算机语言 | 1.2 C语言简介 | 1.2.1 C语言的发展历史 | 1.2.2 C语言的特点 | 1.3 工程问题求解方法论 | 1.4 算法及其描述方法 | 1.4.1 算法的基本概念 | 1.4.2 算法的描述方法 | 1.5 C语言程序初探 | 1.5.1 亲身经历第一个C语言程序 | 1.5.2 认识C语言程序的基本结构 | 1.5.3 项目任务 | 1.5.4 要记住的C语言基本词汇 | 1.5.5 C语言的语句分类 | 1.6 对初学者的忠告 | 1.7 本章小结 | 第2章 C程序设计初步 | 2.1 程序入门 | 2.2 C语言的数据及其类型 | 2.2.1 C语言中的数据 | 2.2.2 数据类型概念 | 2.2.3 C语言数据类型 | 2.3 程序中的常量和变量 | 2.3.1 程序中的直接常量和符号常量 | 2.3.2 C语言程序中的变量 | 2.3.3 确定问题领域的的数据及其类型 | 2.4 程序中的常用运算符和表达式 | 2.4.1 算术运算符和算术表达式 | 2.4.2 赋值运算符和赋值表达式 | 2.4.3 自增(减)运算的进一步理解 | 2.4.4 逗号运算符 | 2.4.5 位运算符 | 2.4.6 指针运算符 | 2.4.7 数据类型转换 | 2.5 程序中的人机交互设计 | 2.6 C语言的基本语句 | 2.7 C语言中的输入输出语句 | 2.7.1 标准格式输入输出语句的简单应用 | 2.7.2 输入输出的复杂格式控制 | 2.7.3 字符/字符串数据的输入输出 | 2.8 工程计算问题求解与数学函数的应用 | 2.8.1 工程应用案例：数值问题求解 | 2.8.2 常用数学函数及其头文件 | 2.9 项目任务：“学生信息管理系统”软件界面的优化设计 | 2.10 本章小结 | 第3章 控制结构和数据文件 | 3.1 算法开发 | 3.1.1 自顶向下设计的算法思想 | 3.1.2 结构化程序设计思想 | 3.2 构造程序中的条件 | 3.2.1 关系运算符与关系表达式 | 3.2.2 逻辑运算符与逻辑表达式 | 3.2.3 控制条件的描述与表示 | 3.3 选择结构程序设计 | 3.3.1 选择结构语句(Conditional Statements) | 3.3.2 选择结构的嵌套 | 3.3.3 switch语句、break语句和goto语句 | 3.3.4 项目案例 | 3.4 循环结构(Loop Structure)程序设计 | 3.4.1 循环机制 | 3.4.2 实现循环的三种语句 | 3.4.3 三种循环的比较 | 3.4.4 循环嵌套 | 3.5 数据文件 | 3.5.1 I/O语句 | 3.5.2 数据文件的读取 | 3.6 应用案例：数值技术 | 3.6.1 定积分的计算 | 3.6.2 一元方程根的数值计算 | 3.7 本章小结 | 中级篇 第4章 函数模块化编程 |高级篇 第5章 数组和矩阵 | 第6章 指针 | 第7章 结构体、联合体及用户自定义类型 | 第8章 项目案例综合实现 | 附录参考文献 |
|-----------------|----------------------|---------------|---------------|-------------|-----------|----------------|--------------|---------------|--------------|---------------|---------------|-------------|--------------------|--------------------|------------|-------------------|----------------|-------------|----------|-------------|----------|----------------|---------------|--------------|---------------|---------------|---------------------|-----------------|----------------------|-------------------|-------------------|-------------------|---------------------|-------------|------------|-------------|--------------|----------------|--------------|-----------------|-----------------------|-------------------|---------------------|----------------------|---------------------|-------------------|------------------------------|-----------|---------------|----------|-------------------|-----------------|--------------|-------------------|-------------------|------------------|--------------|--------------------------------------|---------------|-------------------------------|------------|------------------------------|------------|-----------------|---------------|------------|----------|-------------|---------------|---------------|--------------|------------------|----------|-----------------|--------------------|--------|---------------------|--------------|--------|

<<实用C语言程序设计教程>>

章节摘录

对于要先做判断再选择的问题就要使用选择结构。

选择结构的执行是依据一定的条件选择执行路径，而不是严格按照语句出现的物理顺序。

选择结构程序设计方法的关键在于构造合适的选择条件和分析程序流程，根据不同的程序流程选择适当的选择语句。

循环结构可以减少源程序重复书写的工作量，用来描述重复执行某段算法的问题，这是程序设计中最能发挥计算机特长的程序结构。

三种结构并不彼此孤立，在循环中可以有选择、顺序结构，选择中也可以有循环、顺序结构，其实不管哪种结构，均可广义地把它们看成一条语句。

实际编程过程中常将这3种结构相互结合以实现各种算法，设计出相应程序，但是如果编程的问题较大，编写出的程序就往往很长、结构重复多，造成可读性差，难以理解，解决这个问题的方法是将C语言程序设计成模块化结构。

(3) 通过上机实践理解所学知识 学习程序设计语言，不仅要会阅读程序，更要通过上机实践来学习程序设计，尽量养成独立编写调试程序的习惯。

2.程序设计风格 C语言的书写格式非常灵活，但这容易使程序结构不清晰、表达式难于理解。因此为了提高程序设计的效率、提高程序的可读性、减少程序设计中的错误，要注意程序设计的风格。

程序设计风格是指编写程序的风格。

编写程序时应该遵循下面的一些原则。

(1) 适当的宏定义。

程序中，将一些常用的、值相对确定的数据定义为符号常量。

使用宏定义可增加程序的可读性。

将某些数据集中在宏定义部分便于查找，便于适当的时候对常量的值进行修改。

另外，符号常量最好用大写字母，便于与变量区分，并可避免逻辑错误。

(2) 合适的标识符。

程序中变量、数组、函数和文件等的命名应该尽可能地做到“见名知义”，以增加程序的可读性。

如用radius、area表示圆的半径和面积，用studem - name来表示学生姓名等。

C语言中，变量名一般用小写字母。

(3) 程序书写的缩进规则。

根据语句的并列关系及包含关系，将包含关系中的被包含语句缩进书写。

缩进一般使用Tab键来完成。

在程序中同一层次的{}应该对齐，与该结构语句的第一个字母对齐，并单独占一行。

按缩进格式书写的程序中语句间的逻辑结构清晰，便于阅读程序。

(4) 适当的注释。

注释是一种便于阅读和理解程序的信息，在程序中加入适当注释对于提高程序的可读性、可调性、可维护性都是非常必要的。

<<实用C语言程序设计教程>>

编辑推荐

《实用C语言程序设计教程》内容基于Visual C++6.0集成开发环境，每章配有编程练习和实验，教学中最好配合以小组学习。

结合各种实际工程问题，精心设计应用案例和项目开发任务，把“语句、代码、编程、应用”教学目标融入课程，使编程成为经验体验和创新乐趣的结合。

《实用C语言程序设计教程》配有习题、课件和实验教材。

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>