

<<C/C++程序设计技术>>

图书基本信息

书名：<<C/C++程序设计技术>>

13位ISBN编号：9787111243397

10位ISBN编号：7111243390

出版时间：2008-7

出版时间：机械工业出版社

作者：陈卫卫，王庆瑞 编著

页数：327

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>

<<C/C++程序设计技术>>

内容概要

本书是用于学习C / C++语言程序设计技术的教科书。

全书共7章。主要内容包括：c语言基础，分支和循环，构造类型，函数，指针，类和对象的概念和设计方法，Visualc++6.0上机操作方法，以及用附录形式给出的ASCII码表、常用库函数表、部分习题参考答案等。

本书以基本语法规则为线索，通过126个例题和460多道习题（连同一题多问的小题，共700多道），向读者传授程序设计技术。

本书可作为普通高校计算机科学与技术专业语言课程教材，也可作为广大电脑爱好者学习程序设计方法的参考书。

<<C/C++程序设计技术>>

书籍目录

出版说明前言第1章 基础知识 1.1 程序设计语言的发展和分类 1.2 程序的基本结构和流程 1.2.1 程序的基本结构 1.2.2 程序设计的基本步骤 1.2.3 流程图 1.2.4 程序设计风格 1.3 Visual C++6.0的基本用法 1.3.1 主界面 1.3.2 工作间操作 1.3.3 编译和运行 1.3.4 程序测试和调试 1.4 习题第2章 C语言基础 2.1 概述 2.1.1 C语言的产生和发展 2.1.2 源程序的结构形式 2.2 字词和数据 2.2.1 字和词 2.2.2 数据及其类型 2.3 运算符和表达式 2.3.1 运算符和表达式的一般概念 2.3.2 算术运算 2.3.3 关系运算 2.3.4 逻辑运算 2.3.5 条件运算 2.3.6 赋值运算 2.3.7 逗号运算 2.3.8 位运算 2.4 数据的输入和输出 2.4.1 cin和cout 2.4.2 printf 2.4.3 scanf 2.5 编译预处理 2.6 习题第3章 分支和循环 3.1 分支结构 3.1.1 if语句 3.1.2 复合语句和if语句的嵌套 3.1.3 switch语句 3.1.4 分支程序设计示例 3.2 循环结构 3.2.1 while语句 3.2.2 for语句 3.2.3 do-while语句 3.2.4 多重循环 3.2.5 break语句和continue语句 3.2.6 goto语句 3.3 循环程序设计示例 3.4 习题第4章 构造类型 4.1 数组类型 4.1.1 一维数组 4.1.2 二维数组 4.1.3 字符数组 4.1.4 程序设计示例 4.2 结构类型 4.2.1 定义方式和引用方式 4.2.2 typedef的用法 4.2.3 结构的嵌套和位域 4.2.4 程序设计示例 4.3 联合类型和枚举类型 4.3.1 联合类型 4.3.2 枚举类型 4.4 文件类型 4.4.1 文件的概念和操作步骤 4.4.2 文本文件的读写 4.4.3 二进制文件的读写 4.5 习题第5章 函数 5.1 函数定义和调用 5.1.1 函数定义 5.1.2 函数调用 5.1.3 函数的返回值 5.2 参数传递方式 5.2.1 传值 5.2.2 传地址 5.2.3 传引用 5.2.4 数组作为参数 5.2.5 参数选择的基本原则和选择方法 5.3 变量的作用域和存储属性 5.3.1 作用域 5.3.2 存储属性 5.4 函数的嵌套调用和递归调用 5.4.1 嵌套调用 5.4.2 递归调用 5.5 函数设计示例 5.6 习题第6章 指针类型 6.1 指向普通变量的指针 6.1.1 指针的定义和引用 6.1.2 指向结构类型的指针 6.2 指向数组和函数的指针 6.2.1 指向一维数组的指针 6.2.2 指向字符串的指针 6.2.3 指向二维数组的指针 6.2.4 指向函数的指针 6.2.5 程序设计示例 6.3 动态变量和链表 6.3.1 动态管理函数的用法 6.3.2 new和delete的用法 6.3.3 链表简介 6.4 习题第7章 类和对象 7.1 基本用法 7.1.1 定义方式 7.1.2 引用方式 7.1.3 构造函数和析构函数 7.1.4 程序设计示例 7.2 重载、组合和继承 7.2.1 重载 7.2.2 组合 7.2.3 继承 7.3 虚拟和友元 7.3.1 虚拟函数 7.3.2 虚拟基类 7.3.3 友元 7.4 模板 7.4.1 函数模板 7.4.2 类模板 7.5 习题附录 附录A 数制和码制 附录B ASCII码表 附录C 常用库函数 附录D 部分习题参考答案参考文献

章节摘录

第1章 基础知识 1.1 程序设计语言的发展和分类 众所周知,计算机是在程序的控制下自动工作的,要让计算机完成某项任务,必须为其设计相应的计算机程序。

编写计算机程序(简称编程)必须使用程序设计语言。

程序设计语言则是人和机器都能“懂得”(理解)的一种语言,是人与计算机交流,并指挥计算机工作的工具。

由于计算机中直接参与计算的部件——运算器和控制器等,都是由逻辑电路构成的,而逻辑部件只“认识”0和1,所以程序的最终形式都是由0和1组成的二进制代码形式(指令序列)。

这种二进制代码形式的语言称为机器语言。

早在计算机诞生之初,人们就是用机器语言编程的。

但是,这种在计算机看来十分明了的机器语言程序,在人看来却是一部“天书”。

后来,人们又将3个二进制位合并在一起,这就形成了八进制,再后来,为了与字节对应,又将4个二进制位合并在一起,就变成了十六进制。

将机器语言程序写成八进制或十六进制形式,要比二进制形式“好看”多了。

不管二进制、八进制,还是十六进制,用数字表示程序都不直观,不仅专业性极强,且非常难读难用,编程工作效率低,且极易出错。

好在当初计算机应用面很窄,编程工作量不大,矛盾并不十分突出。

随着计算机应用面不断地扩大,程序需求量大增,编程工作量也越来越大,人们便产生了用符号(通常选用英文字词的缩写)代表机器指令(称为硬指令)的想法,设计出汇编语言(Assemble Language,又称符号语言)。

比如,用ADD表示加法指令,用SUB表示减法指令等,要比形如“00111011”表示某条指令直观得多。

人们将汇编语言编写的程序(称汇编源程序)送入计算机,再由计算机中的汇编程序将源程序自动翻译成计算机能够直接执行的二进制程序(目标程序,可执行程序)。

汇编程序(Assembler,又称汇编器)是专门用来将汇编源程序翻译成机器指令程序的软件。

当然,它也是人们事先编写好,并安装在计算机系统中供反复使用的。

一台计算机配上了汇编程序就相当于人们“教会”计算机认识汇编语言了。

汇编程序把人容易理解的汇编源程序转变成了计算机可直接执行的目标程序。

再后来,人们又设计出反汇编程序,它能将机器语言程序反过来翻译成汇编语言程序。

通过反汇编,人们就可以读懂安装在计算机中的可执行程序。

使用汇编语言减轻了人们不少的编程工作量,但是,汇编语言仍然十分原始,一条汇编语句(也称汇编指令)对应一条机器指令,易读性仍然很差。

编制一个程序,哪怕只是用来完成简单计算任务的程序,通常需要成百上千条汇编指令。

不仅编程效率低,程序不易调试,且容易出错。

更为麻烦的是,这种语言是完全按照计算机硬件设计的,不同种类的计算机都有自己特有的机器语言和汇编语言,一种类型的机器无法识别另一种类型机器的机器语言,所以,汇编源程序缺乏可移植性。

<<C/C++程序设计技术>>

编辑推荐

作者将记、读、仿、练、操五个环节的教学理念，融入到了《高等院校规划教材·计算机科学与技术系列：C/C++程序设计技术》内容的组织与编排上，是作者多年编程语言教学经验的总结。书中提供的各种题型和大量的习题，为读者学好、学通《高等院校规划教材·计算机科学与技术系列：C/C++程序设计技术》提供了坚实的基础；《高等院校规划教材·计算机科学与技术系列：C/C++程序设计技术》配套的电子教案以及可为教师提供的全部习题答案，可以为教师教学提供最大的方便，节省教师宝贵的时间和精力。

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>