



高职高专计算机
系列规划教材

PUP6

高职高专计算机系列规划教材

全国高职高专计算机 **立体化** 系列规划教材

嵌入式C程序设计

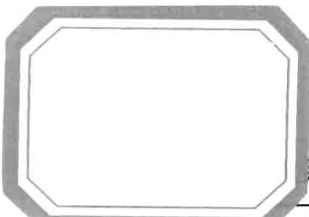


主 编 冯 刚 蔡冬玲
万 兵 周继松

- 按企业工作流程组织教学
- 8个问题学会C语言
- 9个项目掌握智能车编程



北京大学出版社
PEKING UNIVERSITY PRESS



系列规划教材

全国高职高专计算机立体化系列规划教材

嵌入式 C 程序设计

主 编 冯 刚 蔡冬玲
万 兵 周继松
副主编 沈 洋 朱 磊 李天祥



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书是重庆市级精品课程的配套教材。本书针对高职高专和应用型本科教育的特点,以独特的教学设计和先进的开发平台展开C程序设计的教学与实践,将企业软件项目开发流程、规范、岗位技能传授给学生,培养岗位工作能力。全书分为基础知识、基础项目、应用项目、拓展项目4个部分,共18章。基础知识部分(第1~8章)将C程序设计的主要知识及基本算法贯穿于8个学习情境中,并在知识讲解过程中强调编程规范。从基础项目部分开始,以可编程智能小车及其软硬件开发平台为载体展开实践教学,以嵌入式软件开发的工作过程来组织教学内容。基础项目部分(第9~13章)包含5个项目,重在培养编程思维和基本的软件项目开发能力。应用项目部分(第14~17章)包含4个应用开发项目,重在培养软件项目开发能力、软件文档撰写能力、产品测试能力和岗位协同工作能力。拓展项目部分(第18章)以市场需求为导向,指导学生提出改进或创新的产品设计方案并加以实施。

本书可作为计算机类、电子信息类专业的高职高专及应用型本科的教材,也可作为从事计算机应用开发、电子产品设计的工程技术人员的自学参考书。

图书在版编目(CIP)数据

嵌入式C程序设计/冯刚等主编. —北京:北京大学出版社,2012.1

(全国高职高专计算机立体化系列规划教材)

ISBN 978-7-301-19890-2

I. ①嵌… II. ①冯… III. ①C语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第257627号

书 名: 嵌入式C程序设计

著作责任者: 冯 刚 蔡冬玲 万 兵 周继松 主编

策 划 编 辑: 李彦红 刘国明

责 任 编 辑: 李彦红

标 准 书 号: ISBN 978-7-301-19890-2/TP·1202

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路205号 100871

网 址: <http://www.pup.cn> <http://www.pup6.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: pup_6@163.com

印 刷 者: 河北滦县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16开本 15印张 348千字

2012年1月第1版 2012年1月第1次印刷

定 价: 29.00元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

前 言

高职高专和应用型本科的教学,在传授知识技能的同时,更要注重岗位工作流程、规范、行业标准及职业素养方面的训练。

本书不仅介绍 C 程序设计的一般方法,还将企业软件项目开发的流程、规范、岗位技能传授给学生,培养岗位工作能力。本书面向嵌入式软件项目开发,以解决实际问题为目标,按项目需求分析、系统设计、模块划分、模块设计、算法设计与编程调试、软件测试分析与故障排除、项目验收的流程展开教学,通过需求管理员、系统设计员、模块设计员、程序员和测试员等岗位角色扮演和团队式训练来组织教学,培养学生的产品开发能力、团队协作能力和就业上岗能力。

本书以可编程智能小车为 C 程序设计实践教学平台,将 C 语言编程技术与智能电子产品软件设计紧密结合。

本书提出的教学模式和教学手段,在国内现有的 C 程序设计类课程教学中是一个创新。

本书共分 18 章,其主要内容如下:

第 1~8 章为基础知识部分,包括物品寄存问题、计算问题、启箱择器问题、条件判断问题、累计问题、模块化问题、毒酒测试问题和访存问题 8 个学习情境。每个学习情境的教学内容都由趣味案例来引导,按照问题引入、解决问题的方法描述、问题拓展、知识扩充的顺序,将同类问题的程序设计知识贯穿其中,使学生能进行程序设计和较简单的算法设计,能具备程序员岗位的基础能力。

第 9~13 章为基础项目部分,包括灯光控制、行驶控制、光感控制、里程控制、触碰控制 5 个项目。每个项目都按照嵌入式软件模块开发流程组织教学,包括任务下达、必要知识讲解、硬件测试、模块设计、编程与调试、模块测试等,使学生具备软件模块开发能力、基本的文档写作能力和模块测试能力,以满足后续项目开发中模块设计员、程序员和测试员的岗位技能要求。

第 14~17 章为应用项目部分,包括音乐彩灯、小车舞蹈、迷宫机器人、智能清障 4 个应用项目。每个项目都按照企业嵌入式软件开发流程来组织教学,包括需求搜集与分析、需求评审与确认、概要设计、硬件测试、详细设计、编程与调试、软件测试等,使学生具备一定的嵌入式软件完整项目开发能力。

第 18 章为拓展项目部分,由教师指导学生根据市场和用户的需求,提出改进或创新的产品设计方案并完成项目开发。

本书由重庆科创职业学院冯刚、蔡冬玲、万兵、周继松担任主编,大连职业技术学院沈洋、河南推拿职业学院朱磊和四川科技职业学院李天祥担任副主编。

由于作者水平有限,书中疏漏之处在所难免,恳请读者批评和指正。

编 者

2011 年 8 月

目 录

第 1 章 物品寄存问题	1	4.6 习题	63
1.1 主函数结构	2	第 5 章 累计问题	66
1.2 容器命名及物品放入	3	5.1 while 结构	67
1.3 常用容器及刻度	4	5.2 领取津贴的过程	68
1.4 容器的使用规定	7	5.3 领取 Q 币的过程	69
1.5 容器箱	7	5.4 for 结构	70
1.6 变量命名规范(一)	9	5.5 循环嵌套	71
1.7 实作: C-Free5 安装与使用	10	5.6 循环终止	74
1.8 习题	17	5.7 实作: 打字游戏	76
第 2 章 计算问题	20	5.8 习题	81
2.1 四则运算符	21	第 6 章 模块化问题	84
2.2 表单实例	21	6.1 流程图绘制	85
2.3 运算顺序控制及程序注释	24	6.2 函数	88
2.4 if 结构	25	6.3 宏定义	95
2.5 求余	26	6.4 头文件定义	97
2.6 复合运算符	27	6.5 实作: 小月亮餐厅点餐系统	98
2.7 自增与自减	27	6.6 习题	102
2.8 实作: Visual C++ 6.0 安装、配置与 使用	29	第 7 章 毒酒测试问题	105
2.9 习题	36	7.1 初识内存数据	107
第 3 章 启箱择器问题	38	7.2 二进制编码	108
3.1 打开工具箱	39	7.3 进位与溢出	109
3.2 常用工具	39	7.4 位运算	110
3.3 无限循环	47	7.5 实作: 趣味测试	113
3.4 基本注释规范	47	7.6 习题	117
3.5 实作: 时钟	48	第 8 章 访存问题	118
3.6 习题	50	8.1 不同机器中的数据长度	119
第 4 章 条件判断问题	51	8.2 一维数组	120
4.1 if-else 结构	52	8.3 二维数组	123
4.2 if-else 嵌套与 if-else if 结构	54	8.4 字符数组	126
4.3 条件表达式	56	8.5 指针	130
4.4 switch 结构与 break 语句	57	8.6 实作: 打地鼠游戏	134
4.5 实作: 简单选播系统	59	8.7 习题	138

第 9 章 灯光控制基础项目	141	13.6 模块测试	194
9.1 基础项目教学概述	142	第 14 章 音乐彩灯应用项目	196
9.2 项目团队	142	14.1 应用项目教学概述	196
9.3 预备知识	142	14.2 项目团队	196
9.4 任务下达	148	14.3 需求搜集	197
9.5 相关函数介绍	148	14.4 需求确认	198
9.6 模块设计	150	14.5 概要设计	198
9.7 LED 硬件测试	152	14.6 详细设计	199
9.8 程序设计	153	14.7 测试	200
9.9 模块测试	157	第 15 章 小车舞蹈应用项目	202
第 10 章 行驶控制基础项目	159	15.1 需求搜集与确认	202
10.1 任务下达	160	15.2 概要设计	203
10.2 相关函数	160	15.3 详细设计	206
10.3 模块设计	161	15.4 测试	207
10.4 硬件测试	163	第 16 章 迷宫机器人应用项目	208
10.5 程序设计	165	16.1 需求搜集与确认	208
10.6 模块测试	167	16.2 概要设计	209
第 11 章 光感控制基础项目	169	16.3 测试	212
11.1 任务下达	170	第 17 章 智能清障应用项目	214
11.2 光强感知函数	170	17.1 需求搜集	214
11.3 模块设计	170	17.2 概要设计	215
11.4 光感临界值参数测试	173	17.3 详细设计	217
11.5 程序设计	175	17.4 测试	218
11.6 模块测试	177	第 18 章 拓展项目	220
第 12 章 里程控制基础项目	179	18.1 拓展项目教学概述	220
12.1 任务下达	180	18.2 拓展项目开发流程	220
12.2 红外光强感知函数	180	18.3 企业实训	221
12.3 模块设计	180	附录 A ASCII 码表	222
12.4 红外光感临界值参数测试	183	附录 B C 语言中的运算符及其 优先级	224
12.5 程序设计	184	附录 C 匈牙利命名法	226
12.6 模块测试	187	附录 D 输入输出格式符	227
第 13 章 触碰控制基础项目	188	附录 E 常用的 C 库函数	229
13.1 任务下达	189	参考文献	231
13.2 相关函数	189		
13.3 模块设计	190		
13.4 硬件测试	191		
13.5 程序设计	192		

第1章

物品寄存问题



教学目标

通过本章的学习，使学生能使用变量和结构体进行最简单的顺序结构程序设计。



教学要求

知识要点	能力要求	关联知识
主函数	(1) 掌握主函数基本结构 (2) 能在 20 秒以内完成主函数基本结构输入	主函数名 main 空类型 void return 语句
数据类型	掌握 char、unsigned char、short、unsigned short、int、unsigned int、long、unsigned long、bool 9 种数据类型	9 种数据类型的取值范围
变量	掌握变量的用法	变量命名基本规则 变量命名规范 变量类型声明 赋值运算符
结构体	了解结构体的用法	结构体类型定义 结构体变量定义 结构体成员赋值
C-Free5 软件	掌握 C-Free5 编程软件的安装与使用	C-Free5 安装 建立控制台工程



重点难点

- ◇ 数据类型取值范围
- ◇ 变量声明与赋值
- ◇ 结构体类型定义

请思考一个简单的问题，如图 1.1 所示。

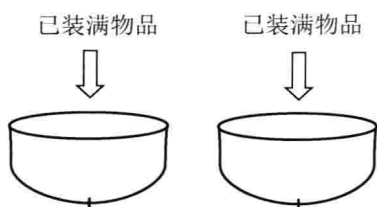


图 1.1 导入案例

假设有两个同样大小的容器，两个容器中都已装满了物品(两个容器中的物品不同，你可以将它们想象成是一碗酱油和一碗醋)。要求人只能碰容器而不能碰容器中的物品，将两个容器中的物品相互交换(原本盛酱油的容器盛醋，原本盛醋的容器盛酱油)。

一般会采用这种解决方法：另取一个新容器，首先把左边容器中的物品放入新容器中，再次把右边容器中物品放入左边容器中，最后把新容器中物品放入右边容器中。

以上方法是正确的，但计算机却不能照此方法去交换容器内物品，因为计算机不能理解人的意图。要让计算机按照人的意图去处理问题，必须先将人的意图转换成计算机认识的语言。C 语言就是计算机认识的一种语言。本章主要介绍如何用 C 语言来描述物品交换的方法。

1.1 主函数结构

1. 认识 4 个英文词语

integer 译为“整数”，在 C 程序中缩写为 int，称为整数类型(简称整型)。

main 译为“主体部分”，C 程序的主体部分称为主函数，main 被用做主函数的名称。

void 译为“空白”，C 程序中称为空类型。

return 译为“返回”，写在主函数的末尾，用于表示程序结束。

2. 主函数基本结构

C 程序主函数的基本结构为：

1	int main(void)
2	{
3	
4	return 0;
5	}

主函数第 1 行由 int、main 和(void)3 部分组成，int 和 main 之间用空格分开。main 是主函数的名称。函数名称左边的文字用于表示函数结束时返回值的类型。因主函数内最后一个语句 return 0 要返回的数是 0，0 是整数，所以 main 左边写为 int。函数名称 main 右边的部分必须用一对圆括号括起来，括号内写参数的类型和名称(关于参数的知识，将在第 6 章介绍，初学者不必提前阅读)。因主函数不需要参数，因此写为 void。

第 2 行和第 5 行是一对花括号，它们将函数体(函数的内容)括起来。

第 3 行是程序的核心，用于书写程序的功能语句。编程者需要计算机做什么事，就把自己的想法写成 C 语言语句，置于此处。此处的篇幅根据程序需要而变化。

第 4 行 return 0 语句意为“将 0 返回给操作系统”，表示程序结束并正常退出。

程序书写规范：C 程序严格区分英文大小写字母；主函数的第 1 行顶格，int 和 main 两部分之间用空格隔开；第 2 行和第 5 行的括号顶格(同一对花括号竖直对齐)；花括号内的语句以花

括号为准右缩进 4 个空格；程序语句以分号结尾；一行仅书写一条语句；return 0 语句中，return 与 0 之间有一个空格。

课堂练习 1：在计算机的记事本中进行输入练习，20 秒内完成主函数基本结构的输入。

1.2 容器命名及物品放入

认识了 C 程序主函数基本结构之后，本节继续分析图 1.1 中物品交换的问题。

前文提到的解决方法是“另取一个新容器，先把左边容器中的物品放入新容器中，再把右边容器中物品放入左边容器中，最后把新容器中物品放入右边容器中”。方法虽正确，但“左边容器”、“右边容器”等词语会让听者感到不明确。若将此方法传达给计算机，计算机也不知道孰左孰右。为了使得方法的描述更清晰，应该给容器命名。

1. 容器命名

给每一个容器命名，能使描述语言或文字更容易被人理解，也能被计算机接受。

容器的名称可以由英文字母、数字和下划线组成，不能以数字开头，不能与表 1-1 中的关键字同名。

表 1-1 C 语言关键字

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	unsigned	union	void	volatile	while

例如，将图 1.1 中左侧容器命名为 iCup1，右侧容器取名为 iCup2，如图 1.2 所示。

容器命名后，交换容器内物品的方法可描述为：另取一个新容器，命名为 iCup3(其他符合命名规则的名称也可)，先将 iCup1 内物品放入 iCup3 中，再将 iCup2 中物品放入 iCup1 中，最后将 iCup3 中物品放入 iCup2 中。

学习怎样描述一个问题，最终目的是让计算机能理解人的意图。现在，计算机已经能明白 iCup1、iCup2 等文字的含义了，但怎么将 iCup2 中物品放入 iCup1 中呢，动作“放入”如何表达？

2. “放入”的表达

在 C 程序中，动作“放入”用等号“=”来表示。其含义为将“=”右侧的物品(或右侧容器中的物品)放入“=”左侧的容器中，如图 1.3 所示。

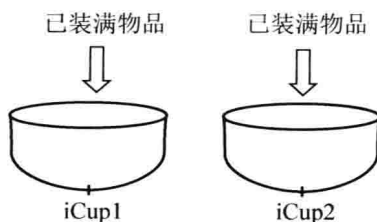


图 1.2 给容器命名

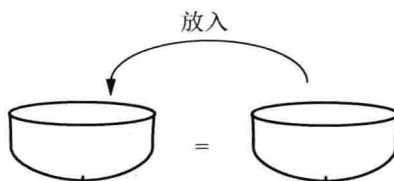


图 1.3 等号的作用

程序书写规范：等号左右各空一格。

【例 1.1】将容器 iY 中的物品放入容器 iX 中，用语句描述为：

```
iX = iY;
```

至此，交换容器中物品的方法可描述为以下 3 条语句：

```
iCup3 = iCup1;
iCup1 = iCup2;
iCup2 = iCup3;
```

将这 3 条语句置于主函数中 return 语句前，就形成了 C 程序(但还不完整)：

```
int main (void)
{
    iCup3 = iCup1;
    iCup1 = iCup2;
    iCup2 = iCup3;
    return 0;
}
```

课堂练习 2：有 5 个容器，如图 1.4 所示，容器名分别为 nBowl_A、nBowl_B、nBowl_C、nBowl_D、nBowl_E，均已装满物品，将容器中物品按箭头所示进行交换。

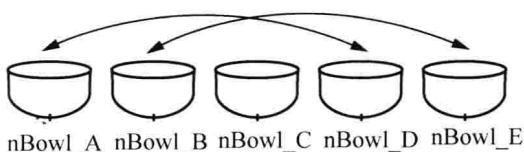


图 1.4 交换 5 个容器

前文多次提到容器中装有物品，计算机的容器中究竟装着什么物品呢？答案：数据。

【例 1.2】把数据 98 放入容器 cTemp 中，用语句描述为以下句式。

```
cTemp = 98;
```

课堂练习 3：先将数据 10、20、30、40 分别放入容器 nBowl_A、nBowl_B、nBowl_C、nBowl_D 中，再按图 1.5 箭头所示进行数据交换，写出程序。

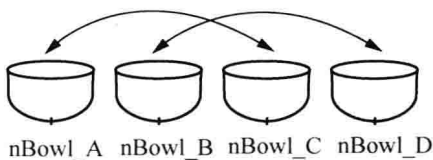


图 1.5 数据交换

1.3 常用容器及刻度

在介绍常用容器之前，先介绍两个英文词语：

character 译为“字符”，是字母与符号的总称，C 程序中缩写为 char，称为字符类型(简称字符型)。

unsigned 译为“无符号的”，表示没有正负符号。

另一个词语 integer 曾在 1.1.1 节介绍，缩写为 int，称为整型。

1. 一对小号容器

有一对小号容器，如图 1.6 所示。

第一个容器的类型为“无符号字符型”，第二个容器的类型为“有符号字符型”（简称“字符型”）。两个容器的容量相等。

第二个容器看似沙漏，沙漏的上下两部分容量相等，都能存放数据，第一个容器看似半个沙漏。

许多读者见过烧杯、试管等带有刻度线的容器。刻度线能反映容器的容量。图 1.6 中的两种容器也有刻度线，其中零刻度线在容器内径最小的位置（即容器上最细的位置）。

unsigned char 类型的容器，刻度最低为 0，最高为 255，该容器可以装入范围为 0~255 的整数，包括 0 和 255，共 256 个可能的数值。

char 类型的容器，零刻度线在容器中部，零刻度线以上的刻度为正数，零刻度线以下的刻度为负数。该类型容器上下两部分容量相等，并且与 unsigned char 类型一样能包含 256 种数值，故刻度最低为 -128，最高为 +127。容器可以装入范围为 -128~+127 的整数。

2. 一对中号容器

有一对中号容器，如图 1.7 所示。

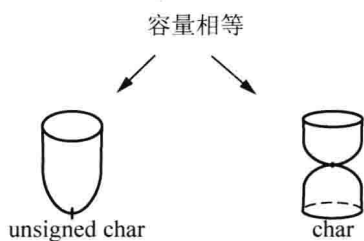


图 1.6 小号容器

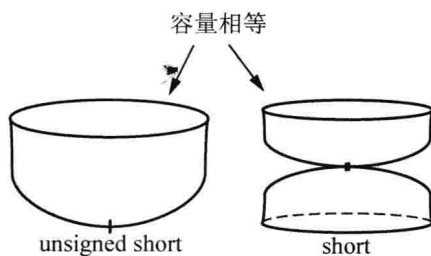


图 1.7 中号容器

第一个容器的类型为“无符号短型”，第二个容器的类型为“有符号短型”（简称“短型”）。两个容器的容量相等。

unsigned short 类型的容器，刻度最低为 0，最高为 65535，该容器可以装入范围为 0~65535 的整数，包括 0 和 65535，共 65536 个可能的数值。

short 类型的容器，零刻度线在容器中部，零刻度线以上的刻度为正数，零刻度线以下的刻度为负数。该类型容器上下两部分容量相等，并且与 unsigned short 类型一样能包含 65536 种取值，故刻度最低为 -32768，最高为 +32767。容器可以装入范围为 -32768~+32767 的整数。

3. 两对大号容器

有两对大号容器，如图 1.8 所示。

第一个容器的类型为“无符号整型”，第二个容器的类型为“有符号整型”（简称“整型”），第三个容器的类型为“无符号长型”，第四个容器的类型为“有符号长型”（简称“长型”）。这四个容器的容量相等。

unsigned int 类型和 unsigned long 类型的容器，刻度最低为 0，最高为 4294967295，该容器

可以装入范围为 0~4294976295 的整数, 包括 0 和 4294976295, 共 4294976296 个可能的数值。

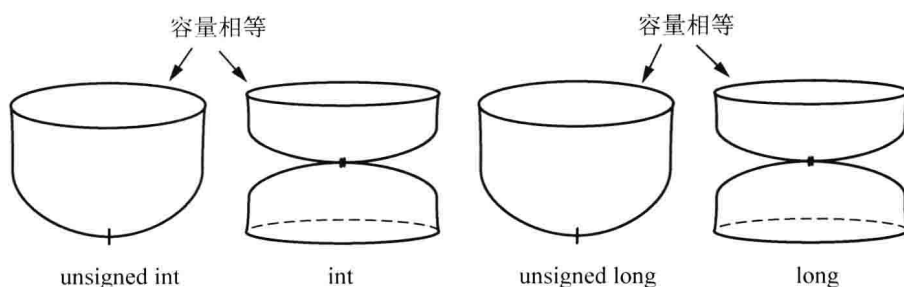


图 1.8 大号容器

int 类型和 long 类型的容器, 零刻度线在容器中部, 上下两部分容量相等, 并且与 unsigned int 及 unsigned long 类型一样能包含 4294976296 种取值, 故刻度最低为 -2147483648, 最高为 +2147483647, 可以装入范围为 -2147483648~+2147483647 的整数。

4. 一种最小号容器

C 语言中最小号的容器, 如图 1.9 所示。

这种容器的类型为“布尔型”, 只能装入 0 和 1 两个数。它允许将 1 表示为 true(真), 将 0 表示为 false(假)。



图 1.9 最小号容器

【例 1.3】若 bCup 是一个布尔类型的容器, 以下两条语句都是将 1 装入 bCup 中:

```
bCup = 1;
bCup = true;
```

以下两条语句都是将 0 装入 bCup 中:

```
bCup = 0;
bCup = false;
```

表 1-2 列出了本节所介绍的 9 种容器类型占用内存空间的情况和取值范围。

表 1-2 各类整数数据的长度及取值范围

数据类型	占用的存储空间	取值范围
unsigned char	1 个字节	0~255
char	1 个字节	-128~+127
unsigned short	2 个字节	0~65535
short	2 个字节	-32768~+32767
unsigned int	4 个字节	0~4294976295
int	4 个字节	-2147483648~+2147483647
unsigned long	4 个字节	0~4294976295
long	4 个字节	-2147483648~+2147483647
bool	1 个字节	0、1

思考 1: 要将数据 500 放入容器 h 中, h 应为哪种类型?

A. unsigned char

B. char

C. unsigned int

D. int

1.4 容器的使用规定

C 语言规定，容器在使用前，必须先说明(声明)容器的类型。形式为：

类型 容器名

【例 1.4】把数据 100 放入容器 iNum 中。iNum 可以是除 bool 类型外其他任意一种类型，若选用 int 类型，语句为：

```
int iNum;
iNum = 100;
```

将这两个语句置于主函数 return 语句前，则形成了一个完整的 C 程序：

```
int main (void)
{
    int iNum ;
    iNum = 100;
    return 0;
}
```

注意：

程序中涉及多个容器时，先声明每一个容器的类型，再对容器进行操作。

【例 1.5】把数据 100 和 150 分别放入容器 iNum1 和 iNum2 中。

因容量限制，容器 iNum1 不能选用 bool 类型，iNum2 不能选用 bool 和 char 类型，程序为：

```
int main (void)
{
    int iNum1;
    int iNum2;
    iNum1 = 100;
    iNum2 = 150;
    return 0;
}
```

课堂练习 4：将数据 100、300 分别放入两个变量中，再把两个变量内数据互换，写出程序。

1.5 容器箱

C 语言还允许使用一种特殊的容器：容器箱。它是“装容器的容器”，任何容器都可以放在容器箱中，如图 1.10 所示。

容器箱可装容器，容器并不是必须放在容器箱中，容器可单独使用。数据只能放在容器中而不能直接放在容器箱中。

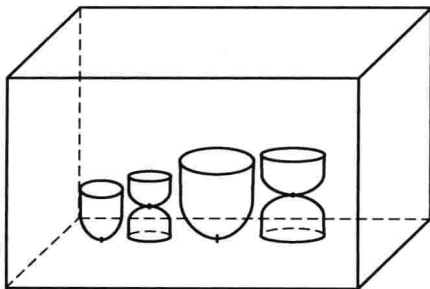


图 1.10 容器箱

1. 设计容器箱类型

前面讲到的 9 种容器，容器的类型是 C 语言规定的，我们只能根据需要选用某种类型的容器，而容器箱的类型却可以由我们自己设计。设计容器箱的类型时，可以规定这种类型容器箱内的容器个数和每个容器的类型。

容器箱的类型，按这种形式设计：

1	struct 容器箱类型名
2	{
3	箱内容器
4	};

设计容器箱的类型，以 **struct** 开头，它与容器箱类型名之间用空格隔开，容器箱类型名由编程者规定。容器箱内各个容器的类型和名称写在花括号内。容器箱内的容器可以是相同类型，但不能是相同名称。

注意：

第 4 行的分号不能省略。

【例 1.6】设计一种容器箱类型，使这种类型的容器箱能容纳 2 个容器，这两个容器都能装入 0~100 的整数。

```
struct Box
{
    char cNum1;
    unsigned char byNum2;
};
```

语句设计了容器箱类型 **Box**，该类型的容器箱内装有两个容器，一个是 **char** 类型的容器，容器名为 **cNum1**，另一个是 **unsigned char** 类型的容器，容器名为 **byNum2**。

课堂练习 5：设计一种容器箱类型，使这种类型的容器箱能容纳 3 个容器，这 3 个容器都能装入 -500~-1000 的整数。

2. 定义容器箱

编程人员设计容器箱类型，就像服装设计师设计服装款式一样，是概念上的行为。服装设计师用图来设计服装，设计图完成了并不等于就做出了服装，必须按照设计图生产加工才能得到服装实体。同理，容器箱类型设计完成了并不等于就有了容器箱，必须按照容器箱类型定义容器箱才能得到实体。

定义容器箱的常用形式为：

```
struct 容器箱类型 容器箱名；
```

【例 1.7】设计一种容器箱类型，使这种类型的容器箱能容纳 2 个容器，这两个容器都能装入 0~100 的整数，按照该类型定义两个容器箱实体。

```
struct Box
{
    char cNum1;
    unsigned char byNum2;
```

```
};
struct Box tEntity1;
struct Box tEntity2;
```

在设计了容器箱类型后，按照这种类型，定义了两个容器箱，容器箱名称分别是 tEntity1 和 tEntity2。

课堂练习 6：设计一种容器箱类型，使这种类型的容器箱能容纳 3 个容器，这 3 个容器都能装入 -1000 ~ -500 的整数，按照该类型定义两个容器箱实体。

3. 使用容器箱

使用容器箱，实际是使用容器箱内的容器，将数据装入箱内容器中。

使用独立存在的容器，可以直接写出容器名称；而使用容器箱内的容器，则采用以下形式。

容器箱名.箱内容器名

【例 1.8】将数据 50 放入容器箱 tEntity1 内容器 cNum1 中，语句为：

```
tEntity1.cNum1 = 50;
```

【例 1.9】编程，设计容器箱类型，该类型容器箱可容纳 2 个容器。定义一个容器箱，给容器箱内的两个容器分别装入数据 50 和 100，再将这两个容器内数据互换。

```
int main (void)
{
    char cTemp;                // 声明容器 cTemp 的类型
    struct Box                 // 设计容器箱类型
    {
        char cx;
        char cy;
    };
    struct Box tEntity;        // 定义容器箱，容器箱名称 tEntity
    tEntity.cx = 50;            // 将 50 放入容器箱 tEntity 内的容器 cx 中
    tEntity.cy = 100;          // 将 100 放入容器箱 tEntity 内的容器 cy 中
    cTemp = tEntity.cx;        // 交换容器箱 tEntity 内两个容器中的数据
    tEntity.cx = tEntity.cy;
    tEntity.cy = cTemp;
    return 0;
}
```

程序中，cx 和 cy 是容器箱内的容器，使用它们时必须加上容器箱名作为前缀。而 cTemp 是一个独立的容器，在声明了它的类型后便可进行使用。

课堂练习 7：编程，设计容器箱类型，该类型容器箱可容纳 2 个容器。定义两个容器箱，给每个容器箱内的任意一个容器装入数据，再将这两个装有数据的容器内容互换。

1.6 变量命名规范(一)

本章为便于初学者理解，多次使用“容器”和“容器箱”两个词语。实际上，在 C 语言程序中，容器被称为“变量”，例如，容器 iNum 称为“变量 iNum”；容器箱被称为“结构体”，例如，容器箱 tEntity 称为“结构体 tEntity”；容器箱内的容器，往往被称为“成员”，例如，

结构体 tEntity 内的变量 cx 称为“结构体 tEntity 的成员 cx”。前文多次提到等号“=”的作用为“放入”或“装入”，实际在 C 语言中，等号“=”的作用通常被称为“赋值”，例如“iNum = 5”称为“将 iNum 赋值为 5”或“将 5 赋值给 iNum”。

匈牙利命名法是一种编程时的命名规范。这种命名法是一位叫 Charles Simonyi 的匈牙利程序员发明的，后来他在微软工作了几年，于是这种命名法就通过微软的各种产品和文档资料向世界传播开了。现在，大部分程序员不管自己使用什么软件进行开发，或多或少都使用了这种命名法。匈牙利命名法的出发点是：根据类型加前缀，用英文词语或词语的缩写描述变量作用，词语首字母大写。该命名法使程序阅读者对变量的类型和其他属性有直观的了解。表 1-3 仅列举出本章所介绍的 9 种类型变量和结构体的命名规范，更多命名规范见附录 C。

表 1-3 匈牙利命名法(一)

数据类型	前 缀	举 例
unsigned char	uc 或 by	unsigned char ucFlag; unsigned char byFlag;
char	c	char cCh;
unsigned short	w	unsigned short wYear;
short	n	short nStepCount;
unsigned int	u	unsigned int uNum;
int	i	int iTemp;
unsigned long	ul 或 dw	unsigned long ulResult; unsigned long dwResult;
long	l	long lSum;
bool	b	bool blsAlpha;
结构体	t	struct Date tMyBirthday;

1.7 实操：C-Free5 安装与使用

C-Free5 是一款基于 Windows 的 C/C++ 集成化开发软件。利用该软件，使用者可以轻松编辑、编译、连接、运行、调试 C/C++ 程序，这款软件操作简单，非常适合 C/C++ 的初学者，容易使用。读者可从网站 <http://www.programarts.com> 下载软件。本节介绍 C-Free5 的安装和使用步骤，请读者按步骤进行操作。

1. C-Free5 安装

(1) 打开 C-Free5 安装程序，会出现安装向导欢迎界面如图 1.11 所示，单击【下一步】按钮。

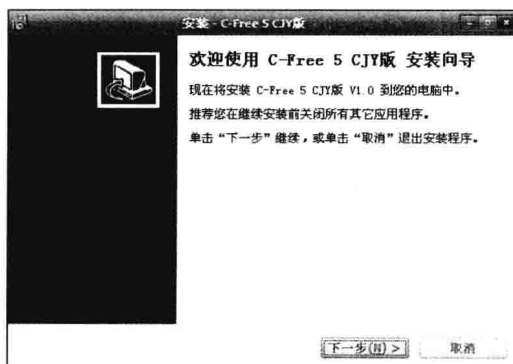


图 1.11 C-Free5 安装向导欢迎界面

(2) 如图 1.12 所示, 在许可协议下方选中【我同意此协议】单选按钮, 单击【下一步】按钮。

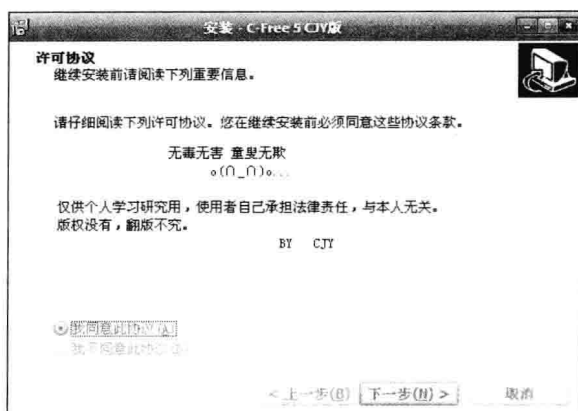


图 1.12 C-Free5 使用许可协议界面

(3) 如图 1.13 所示, 在新增功能信息下方, 单击【下一步】按钮。

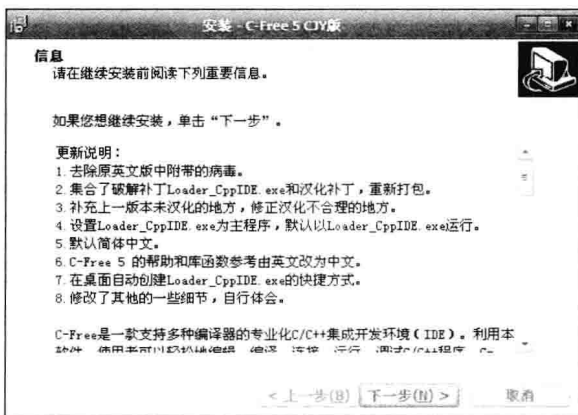


图 1.13 软件新增功能信息提示界面

(4) 如图 1.14 所示, 可通过单击【浏览】按钮设定 C-Free5 的安装路径。设定后, 单击【下一步】按钮。

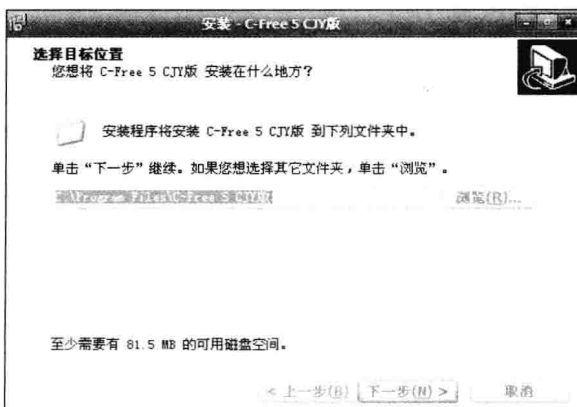


图 1.14 软件安装路径设置界面