



普通高等教育“十二五”创新型规划教材

C 语言程序设计 能力教程

C YUYAN CHENGXU SHEJI NENGLI JIAOCHENG

主 编 李小遐 刘惠梅

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

C 语言程序设计

能力教程

主 编 李小遐 刘惠梅
副主编 魏晓艳 宫护震
徐华宇 戚 斌

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书是 C 语言的入门教材。全书共有 11 章,把 C 语言的学习分为三大部分。第 1 部分(第 1,2 章)介绍了 C 程序的基本框架、C 程序的实现过程、数据类型及数据处理等程序设计的基础知识。第 2 部分(第 3,4,5,6,7 章)介绍了三种结构程序的设计、数组、函数等内容,培养程序设计的基本能力。第 3 部分(第 8,9,10,11 章)介绍了指针、结构体、文件、应用程序的开发过程等内容,培养设计应用程序的能力。

全书始终以程序案例驱动,将语法知识点同实际编程相结合,循序渐进,实例丰富有趣,便于教师讲授和学生学习。每章配有上机实训内容指导,便于学生上机实践。

本书适合作为高等院校的 C 语言课程的学习教材,也适合作其他人员学习 C 语言编程的入门书籍。

版权专有 侵权必究

图书在版编目(CIP)数据

C 语言程序设计能力教程/李小遐,刘惠梅主编. —北京:北京理工大学出版社,2011.7

ISBN 978 - 7 - 5640 - 4818 - 1

I. ①C… II. ①李… ②刘… III. ①C 语言 - 程序设计 - 教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 137357 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市文通印刷包装有限公司

开 本 / 710 毫米×1000 毫米 1/16

印 张 / 16.25

字 数 / 304 千字

版 次 / 2011 年 7 月第 1 版 2011 年 7 月第 1 次印刷

印 数 / 1~1500 册

定 价 / 36.00 元

责任校对 / 陈玉梅

责任印制 / 王美丽

图书出现印装质量问题,本社负责调换

前言

Preface <<< <<<

本书是为高等院校编写的 C 语言学习教材。

高等院校人才培养的目标是技能型、应用型，所以本书作为 C 语言的入门级教材，在教学内容的整合上以实用为主、知识够用为原则，将一些理论性强或不太实用的内容做了一定的处理，比如降低难度或适当删减。

针对高等院校学生的特点，编者特别强调在实践过程中学习 C 语言，领会程序设计的方法。本书的编写思想是以程序案例驱动，将语法知识点同实际编程相结合，避免纠缠于语法细节。按照“提出问题—分析问题—用 C 程序解决问题—分析程序中的语法现象”的路径来讲解，有利于培养学生分析问题与解决问题的能力，也使得学生对语法的理解更为容易。

教学内容安排得是否合理，直接影响学生的学习效果。因此，本书特别注意前后内容的编排和衔接，以方便教师讲授和学生学习。

本书各章按以下形式组织：

学习目标：为教师和学生规定明确的教学目标和学习目标。

学习内容：给出本章所有知识点。

授课内容：教师课堂讲授内容。为了弥补 C 语言语法的枯燥，本书配备大量的教学实例而且特别注重这些实例的合理性和趣味性。

本章小结：总结本章重要知识点，帮助学生整理复习。

实训：本章实训内容及指导。针对本章所学的语法知识，精心设计了上机实训内容指导，既方便教师布置学生上机实训作业，也便于学生上机前准备和上机后总结以及书写实训报告。

习题：对本章内容的练习和巩固。

书中还设置了“提示”和“小测验”环节。容易出错的内容或特别需要说明的内容，以提示形式给出，比较醒目容易记忆，小测验中提出的问题便于学生举一反三。

本书增加了 C 语言程序综合实训一章，从应用程序开发的角度介绍了程序设计的一般步骤，给出了课程综合实训的计划和安排，供授课教师参考。

本书提供可直接使用的电子教案（PPT），教师也可以根据需要修改后使用。

本书提供教学案例集，其中包括全书所有实例的源代码文件、各章的实训题目及习题答案。源代码均在 Visual C++ 6.0 下运行通过，所有实例的输出结果均采用屏幕拷贝后截取所得，充分体现了源代码的正确性。有需要的读者可从北京理工大学出版社网站（www.bitpress.com.cn）下载，或发邮件至邮编 yyzq_11@163.com 索取。

本书由李小遐、刘慧梅担任主编，负责全书的策划和统稿，魏晓艳、宫护震、徐华宇、戚斌担任副主编。其中李小遐编写第 1、2、3 章，魏晓艳编写第 4、10 章，宫护震编写第 5、9 章，刘慧梅编写第 6、11 章，戚斌编写第 7 章，徐华宇编写第 8 章及附录。

本书在编写过程中参考了大量的文献资料，同时也得到北京理工大学出版社的大力协助，在此对参考文献的作者和北京理工大学出版社一并表示感谢。由于编者水平有限，书中不足之处恳请读者批评指正。

编 者

目录

Contents <<< <<<

第1章 C语言程序基本框架	(1)
1.1 初识C语言程序	(1)
1.1.1 C程序实例	(1)
1.1.2 C程序的组成	(3)
1.2 C语言程序的实现	(5)
1.2.1 C程序实现过程	(5)
1.2.2 C程序开发环境	(6)
1.3 本章小结	(9)
1.4 实训	(9)
习题1	(10)
第2章 基本数据及其运算	(12)
2.1 一个C程序实例	(12)
2.2 基本数据类型	(13)
2.3 常量和变量	(14)
2.3.1 常量	(14)
2.3.2 变量	(16)
2.4 运算符和表达式	(17)
2.4.1 算术运算符和算术表达式	(18)
2.4.2 赋值运算符和赋值表达式	(20)
2.4.3 自增自减运算	(21)
2.5 本章小结	(22)
2.6 实训	(23)
习题2	(25)
第3章 顺序结构及其应用	(27)
3.1 一个顺序结构程序实例	(27)
3.2 数据的输入和输出	(28)
3.2.1 格式输出函数	(28)
3.2.2 格式输入函数	(31)
3.2.3 字符输入输出函数	(32)
3.3 顺序结构应用实例	(33)



3.4 本章小结	(35)
3.5 实训	(36)
习题 3	(38)
第 4 章 分支结构及其应用	(40)
4.1 一个分支结构程序实例	(40)
4.2 分支结构中条件的描述	(41)
4.2.1 关系运算符和关系表达式	(41)
4.2.2 逻辑运算符和逻辑表达式	(42)
4.3 实现分支结构的语句	(43)
4.3.1 if 语句	(43)
4.3.2 switch 语句	(50)
4.4 分支结构应用实例	(52)
4.5 本章小结	(56)
4.6 实训	(56)
习题 4	(60)
第 5 章 循环结构及其应用	(64)
5.1 一个循环结构程序实例	(64)
5.2 实现循环结构的语句	(65)
5.2.1 while 语句	(65)
5.2.2 do-while 语句	(68)
5.2.3 for 语句	(70)
5.3 循环嵌套	(74)
5.4 循环结构应用实例	(77)
5.5 本章小结	(82)
5.6 实训	(82)
习题 5	(88)
第 6 章 数组及其应用	(92)
6.1 一个 C 程序实例	(92)
6.2 一维数组	(94)
6.2.1 一维数组的定义和使用	(94)
6.2.2 一维数组应用举例	(97)
6.3 二维数组	(100)
6.3.1 二维数组的定义和使用	(100)

6.3.2 二维数组应用举例	(103)
6.4 字符串及其处理	(106)
6.4.1 字符串与字符数组	(106)
6.4.2 常用字符串处理函数	(108)
6.4.3 多个字符串的存储与处理	(110)
6.4.4 字符串应用举例	(112)
6.5 数组应用实例	(113)
6.6 本章小结	(117)
6.7 实训	(118)
习题 6	(124)
第 7 章 函数及其应用	(128)
7.1 一个 C 程序实例	(128)
7.2 函数的定义和调用	(130)
7.2.1 函数的定义	(130)
7.2.2 函数的调用	(131)
7.2.3 函数的声明	(134)
7.2.4 函数间的数据传递	(135)
7.3 变量的进一步讨论	(136)
7.4 函数应用实例	(139)
7.5 本章小结	(145)
7.6 实训	(145)
习题 7	(149)
第 8 章 指针及其应用	(153)
8.1 指针的基本用法	(153)
8.1.1 指针的概念	(153)
8.1.2 指针变量的定义和使用	(154)
8.2 指针与数组	(156)
8.2.1 指针与一维数组	(156)
8.2.2 指针与二维数组	(159)
8.2.3 指针与字符串	(162)
8.3 指针与函数	(164)
8.3.1 变量地址作函数参数	(164)
8.3.2 数组名作函数参数	(165)
8.4 指针应用实例	(168)



8.5 本章小结	(171)
8.6 实训	(172)
习题 8	(175)
第 9 章 结构体及其应用	(180)
9.1 一个程序实例	(180)
9.2 结构体类型的使用	(182)
9.2.1 结构体类型的定义	(182)
9.2.2 结构体变量的定义及引用	(183)
9.2.3 结构体数组	(186)
9.2.4 结构体指针	(188)
9.3 链表	(190)
9.3.1 链表的基本结构	(190)
9.3.2 链表的基本操作	(192)
9.3.3 链表综合应用	(197)
9.4 结构体应用实例	(198)
9.5 本章小结	(202)
9.6 实训	(202)
习题 9	(206)
第 10 章 文件及其应用	(209)
10.1 文件的概念	(209)
10.1.1 文件的概念及文件类型	(209)
10.1.2 文件指针	(210)
10.2 文件的基本操作	(210)
10.2.1 文件的打开与关闭	(211)
10.2.2 文本文件的读写	(212)
10.2.3 二进制文件的读写	(215)
10.3 文件应用实例	(220)
10.4 本章小结	(223)
10.5 实训	(223)
习题 10	(226)
第 11 章 C 语言程序综合实训	(228)
11.1 应用程序设计步骤	(228)
11.2 应用程序设计实例	(229)

11.3 C 语言课程实训	(236)
附录 A ASCII 码表	(242)
附录 B C 语言的库函数	(244)
附录 C C 语言运算符优先级和结合性	(248)
参考文献	(250)

学 习 目 标

初识 C 语言程序，了解 C 语言程序的基本构成及构成元素，掌握 C 语言程序的上机操作过程。

学 习 内 容

C 语言程序的组成，C 语言程序的实现过程及开发环境。

1.1 初识 C 语言程序

C 语言是一种应用广泛的计算机程序设计语言，学习程序设计语言是使用计算机和研制计算机软件的必由之路。现在就通过几个程序实例，走进 C 语言的世界。

1.1.1 C 程序实例

【例 1-1】第一个 C 程序，在显示屏上显示信息：This is my first C program!

【程序代码】

```
#include "stdio.h"
main()
{
    printf("\nThis is my first C program!\n");
}
```

程序执行后得到的输出结果为：

```
This is my first C program!
```

【程序分析】

(1) 本程序非常简单，仅由一个主函数（函数名为 main）构成，在主函数中也只有一条语句，该语句是通过标准输出函数 printf() 在显示屏上输出指定信息。

(2) 程序开头的 #include "stdio.h" 是预处理命令，其作用是包含输入输出库文件，当程序中调用标准输入或输出函数时添加此行。



【例 1-2】计算并输出一个数的平方。

【程序代码】

```
#include "stdio.h"

main()                /*主函数，程序从这里开始运行*/
{                    /*函数体开始*/
    float a,b;        /*定义语句*/
    a=2.8;            /*赋值语句*/
    b=a*a;            /*赋值语句*/
    printf("%f\n",b); /*标准输出函数*/
}                    /*函数体结束*/
```

程序执行后得到的结果为：

7.840000

【程序分析】

(1) 程序中书写在“/*”和“*/”之间的是注释，程序中添加注释的目的是帮助程序阅读者阅读理解程序。必要的注释可以增加程序的可读性，但是注释对程序的执行没有任何影响，编译时将被过滤掉，因此注释可以添加在程序的任何位置。

(2) 本程序仍然由一个主函数构成，相对例 1-1 稍复杂的是，主函数中含有 4 个不同的语句，共同完成程序要求的计算功能。

【例 1-3】从键盘任意输入两个数，输出其中的大数。

【程序代码】

```
#include "stdio.h"

int max(int x, int y) /*定义 max()函数，求两个数中的大数*/
{
    int z;
    if (x>y) z = x; else z = y;
    return (z);        /*返回函数值*/
}

main()                /*主函数，程序从这里开始执行*/
{
    int a,b,c;
    printf("请输入两个整数: ");
    scanf("%d, %d",&a,&b); /*标准输入函数*/
    c = max(a,b);        /*调用 max()函数*/
    printf("%d, %d 中的大数为: %d\n\n",a,b,c); /*标准输出函数*/
}
```

该程序执行时，如果从键盘提供两个整数 12 和 21，得到的输出结果为：

```
请输入两个整数: 12,21
12,21中的大数为: 21
```

该程序执行时，如果从键盘提供的两个整数是 31 和 13，那么得到的输出结果则为：

```
请输入两个整数: 31,13
31,13中的大数为: 31
```

【程序分析】

(1) 本程序由两个函数模块组成：主函数 `main()` 和子函数 `max()`。子函数 `max()` 称为自定义函数，功能是找出两个数中的大数。

(2) 标准库函数由 C 系统定义，用户在程序中只需直接调用即可，比如程序中的输入输出函数 `scanf()` 和 `printf()`。与标准库函数不同，自定义函数由用户定义，一旦定义好后，就可以像标准库函数一样使用。有关自定义函数的内容，本书将在第 7 章介绍。

(3) 程序从 `main()` 函数开始执行，执行到语句 `c = max(a,b);` 时转到 `max()` 函数，遇到 `return` 语句返回主函数继续执行，直到程序结束。

1.1.2 C 程序的组成

以上实例及实例分析已经反应出了 C 程序的基本框架及其组成要素。

1. C 程序是由函数构成的

C 程序通常由包括 `main()` 在内的一个或多个函数组成，函数是构成 C 程序的基本单位。其中，主函数必须有且只能有一个，被调用的其他函数可以是系统提供的库函数，也可以是用户自定义的函数。C 程序的全部工作都是由函数来完成的，因此 C 语言被称为函数式语言。

2. 函数的构成

C 的函数由函数首部和函数体两大部分组成。以主函数为例：

<code>main()</code>	← 函数首部，指定函数名、函数参数等信息
<code>{</code>	← 函数体从这里开始
<code>...</code>	← 函数体内的语句
<code>}</code>	← 函数体到这里结束

3. 语句

语句是构成函数的基本单位，函数功能的实现由若干条语句序列完成。程序中的语句有说明语句和执行语句，说明语句完成数据的描述，执行语句完成指定的操作功能。每个语句必须以“;”结束，这是 C 语言的一个特色。

4. 标识符

标识符是程序中函数、变量、语句及数据类型等对象的名称。C 语言的标识符可分为两类：

(1) 关键字。关键字是 C 语言系统规定的、具有特定含义和专门用途的一些字符序列，如前面程序中出现的 `main`、`int`、`printf`、`scanf`、`include`、`return` 等。

程序中使用关键字时，不能用错也不能挪作他用。

(2) 用户标识符。用户标识符是设计程序时用户自己定义的名字，这类标识符命名时要遵守以下规则：由字母（26 个大、小写字母）、数字（0~9）和下划线（“_”）组成，不能以数字开头，区分大小写。为了便于使用，命名应简洁、实用。

小测验

区分以下标识符中，哪些是合法标识符，哪些是不合法标识符？

<code>int</code>	<code>define</code>	<code>double</code>	<code>swicth</code>	<code>while</code>
<code>m+y</code>	<code>a#</code>	<code>b-4</code>	<code>π</code>	<code>2x</code>
<code>_1</code>	<code>Abc</code>	<code>_max</code>	<code>b_1</code>	<code>a123</code>
<code>name</code>	<code>a</code>	<code>al</code>	<code>sum</code>	<code>aver</code>

提示：（1）标识符中不能含有除字母、数字和下划线外的其他字符。

（2）用户标识符不能与保留字重名。

（3）标识符命名以“顾名思义”为佳。

5. C 程序的书写

C 程序的书写没有格式要求，因此可以自由书写。但是为了程序的易读性，长期以来人们还是形成了一定的规范。

（1）C 语言没有行的概念，但是通常一行写一条语句，当然一行可以写多条语句，一条语句也可以写在多行上。

（2）整个程序采用缩进格式书写，表示同一层次的语句行对齐，缩进同样多的字符位置。比如选择体和循环体中的语句要缩进对齐。

（3）程序代码习惯用小写字母，只是在特定的时候才使用大写字母。

（4）在程序中恰当地使用空行，以便分隔程序中的语句块，增加程序的可读性。

一个好的程序员应该养成合理使用缩进和对齐的好习惯，从而使程序的结构更加清晰。例如，以下两个程序都能编译成功，但你更愿意读哪个程序？

```
#include "stdio.h"
main()
{
    printf("有趣的程序");
}
```

```
#include "stdio.h"
main()
{
    printf("有趣的程序");}
```

提示：C 程序中花括号“{}”用得比较多，书写程序时要注意“{”和“}”的成对。

1.2 C 语言程序的实现

1.2.1 C 程序实现过程

按照 C 语言语法规则编写的 C 程序称为源程序。设计好 C 的源程序后，要将其输入到计算机并得到最终结果，必须经过编辑、编译、连接和运行这几个主要环节，其实现过程如图 1-1 所示。

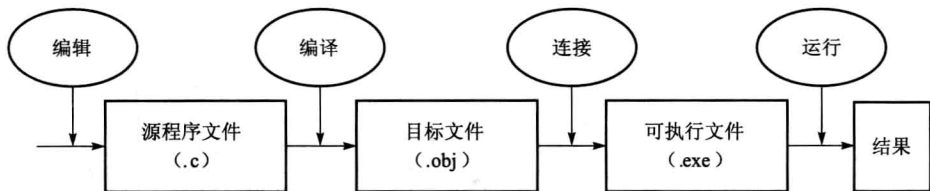


图 1-1 C 程序实现过程流程

1. 编辑源程序

设计好的源程序要利用程序编辑器输入到计算机中，输入的程序一般以文本文件的形式存放在磁盘上，文件的扩展名为.c。所用的编辑器可以是任何一种文本编辑软件，比如像 Turbo C 和 VC++ 这样的专用编辑系统，或者是 Windows 系统提供的写字板或字处理软件等都可以用来编辑源程序。

2. 编译源程序

源程序是无法直接被计算机执行的，因为计算机只能执行二进制的机器指令，这就需要将源程序先翻译成机器指令，然后计算机才能执行翻译好的程序，这个过程是由 C 语言的编译系统完成的。源程序编译之后生成的机器指令程序叫目标程序，其扩展名为.obj。

3. 连接程序

在源程序中，输入输出等标准函数不是用户自己编写的，而是直接调用系统函数库中的库函数。因此，必须把目标程序与库函数进行连接，才能生成扩展名为.exe 的可执行文件。

4. 运行程序

执行.exe 文件，得到最终结果。

在编译、连接和运行程序过程中，都有可能出现问题，此时可根据系统给出的错误信息对源程序进行修改，并重复以上环节，直到得出正确的结果为止。

1.2.2 C 程序开发环境

C 语言的标准已被大多数 C 和 C++ 的开发环境所兼容，我们可以使用很多工具开发自己的 C 语言程序。下面以 Microsoft Visual C++ 6.0 为上机平台，介绍 C 程序的实现过程。

VC++ 集成环境不仅支持 C++ 程序的编译和运行，而且也支持 C 语言程序的编译和运行。通常 C++ 集成环境约定：当源程序文件的扩展名为 .c 时，则为 C 程序，而当源程序文件的扩展名为 .cpp 时，则为 C++ 程序。

1. 启动 VC++

启动“Microsoft Visual C++ 6.0”，屏幕将显示图 1-2 所示的窗口。

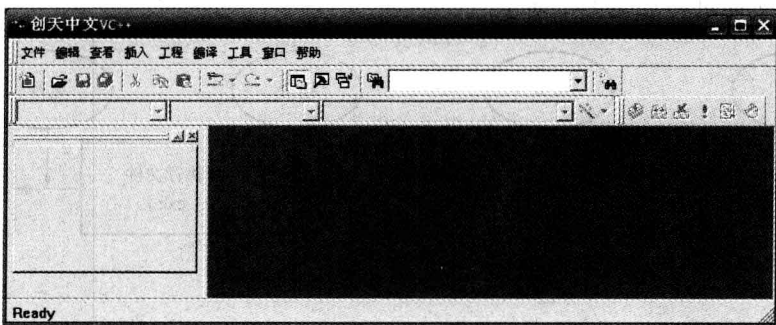


图 1-2 VC++ 6.0 界面

2. 新建 C 程序文件

在图 1-2 中选择“文件”菜单的“新建”菜单项，会弹出“新建”窗口，如图 1-3 所示。单击“文件”标签，选中“C++ Source File”，同时在右边文件输入框中输入自己的文件名，如“a01.c”，在目录框中选择或输入文件路径，然后按“确定”。

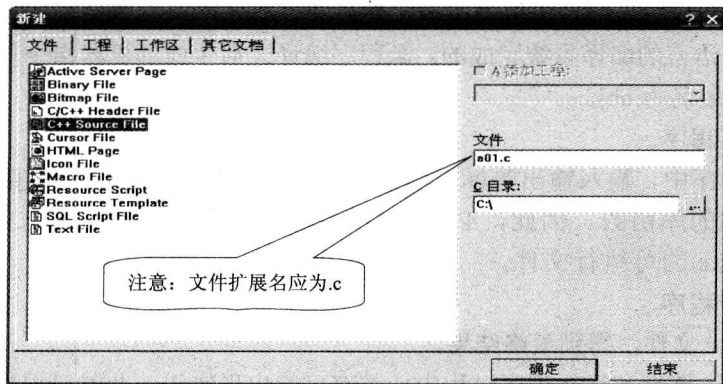


图 1-3 新建文件窗口

3. 编辑源程序

在图 1-4 所示的编辑窗口输入程序代码。由于完全是 Windows 界面，输入及修改可借助鼠标和菜单进行，十分方便。

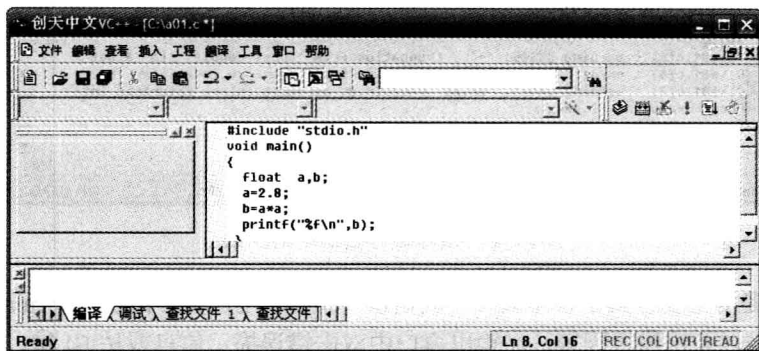


图 1-4 源程序编辑界面

4. 保存程序

在图 1-4 所示的编辑窗口中，选择“文件”菜单中的“保存”命令，将源程序保存到指定的文件中。

5. 编译程序

在图 1-4 所示的编辑窗口中，选择“编译”菜单中的“编译 a01.c”命令。

6. 运行程序

在图 1-4 所示的编辑窗口中，选择“编译”菜单中的“执行 a01.exe”命令，立即可以看到程序的运行结果。

对于编译连接和运行操作，VC++还提供了一组快捷工具按钮，如图 1-5 所示。

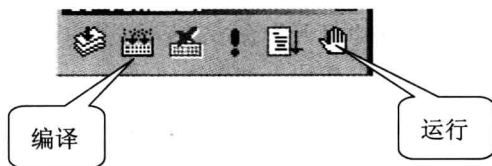


图 1-5 编译运行工具按钮

7. 调试程序

调试程序是程序设计中一个很重要的环节，一个程序很难保证一次就能运行通过，一般都要经过多次调试。

程序中的错误一般分为源程序语法错误和程序设计上的逻辑错误，编译时只能找出语法错误，而逻辑错误需要靠程序员手工查找。

如果程序中存在语法错误，那么编译时会在输出窗口中给出错误提示，如图 1-6 所示。