

程序设计基础 (C语言)

实验与习题指导

主 编 杨永斌 丁明勇

副主编 何希平 张小莉 代小红



科学出版社

内 容 简 介

本书包括三章。第1章介绍 Turbo C 集成开发环境和 Visual C++ 集成开发环境下编写、调试 C 程序所需的基本知识。第2章精心设计了20个与 C 语言程序设计知识点紧密相关的实验项目,实验1至实验19包括多个验证性实验、填空性实验、程序改错和设计性实验,实验20提供了5个综合性实验供读者进行课程设计时采用。实验项目采用任务驱动模式,每个实验目的明确、内容丰富、可操作性强,注重训练读者解决实际问题的能力、实际动手和应用能力。第3章按照 C 语言程序设计的知识体系进行分析讲解,包括知识要点复习、典型题例解析和自测训练,方便读者自主学习和复习,巩固所学知识,掌握典型的解题方法和思路。

本书是笔者多年的教学实践经验和实际案例的总结,内容丰富,讲解翔实,突出重点、难点和考点。本书既是一本实验指导书,也是一本学习、复习指导书。

图书在版编目(CIP)数据

程序设计基础(C语言)实验与习题指导/杨永斌,丁明勇主编. —北京:科学出版社,2012

ISBN 978-7-03-035336-8

I. ①程… II. ①杨… ②丁… III. ①C语言-程序设计-高等学校-教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 192603 号

责任编辑 胡云志 石 悦 程 颢/责任校对:李 影

责任印制:阎 磊/封面设计:华路天然设计工作室

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

北京市文林印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2012年8月第 一 版 开本:787×1092 1/16

2012年8月第一次印刷 印张:15 3/4

字数:404 000

定价:31.00 元

(如有印装质量问题,我社负责调换)

前 言

C语言概念简洁,数据类型丰富、表达能力强、使用灵活方便、目标程序高、可移植性好,程序结构性和可读性好,兼具高级语言和低级语言的优点,既适合编写系统软件,也适合编写应用软件,是当今最具影响力、最流行的程序设计语言之一。

但对于程序设计的初学者来说,C语言规则较多,使用较灵活,不易掌握;另外,随着C语言的应用范围越来越大,所涉及的知识也在不断增加,学习有一定的困难。为了更好地满足课堂教学和读者自学的要求,培养读者的学习兴趣和实际应用能力,达到学以致用目的,我们撰写了本书。

本书为读者提供了内容丰富的上机实验、知识要点复习、典型题例解析和自测训练。目的是帮助读者进行目的明确的上机实验,更好地消化教材内容,有计划地对所学知识进行复习总结和自测,提高学习效果。

本书包括三章。第1章介绍C语言程序设计开发环境(Turbo C、Visual C++)。第2章精心设计了20个与C语言程序设计知识点紧密相关的实验项目,实验1至实验19包括多个验证性实验、填空性实验、程序改错和设计性实验,实验20提供了5个综合性实验供读者进行课程设计时采用。实验项目采用任务驱动模式,每个实验目的明确、内容丰富、可操作性强,注重训练读者解决实际问题的能力及实际动手和应用能力。第3章按照C语言程序设计的知识体系进行分析讲解,包括知识要点复习、典型题例解析和自测训练,方便读者自主学习和复习,巩固所学知识,掌握典型的解题方法和思路。

另外,附录1给出了25个常用算法思想及其实现过程,每个算法先讲解求解思想,然后以函数形式给出其程序实现,并用主程序给出其调用实例。附录2是一套C语言模拟试题。

读者在学习程序设计语言时,要重点解决阅读程序和编写程序的能力。本书力求将程序设计方法、读程序技巧、编程序方法等巧妙地贯穿于实用性的实例之中,让读者耳目一新。同时本书非常注重对读者的分析问题能力、逻辑思维能力、解决问题能力的培养和训练。

本书编者都是长期担任“C语言程序设计”课程及计算机专业主干课程10年以上的一线教师,他们有丰富的课程教学经验,清楚什么样的书才是老师好用、学生受用的。本书是编者多年的教学实践经验和实际案例的总结,内容丰富,讲解翔实,突出重点、难点和考点。本书既是一本实验指导书,也是一本学习、复习指导书。

本书知识体系系统完整,可以作为C语言程序设计相关教程的配套实验教程和学习指导书。

本书由重庆工商大学杨永斌、丁明勇担任主编,何希平、张小莉、代小红担任副主编,并完成全书的编写、统稿、阅校工作。参与编写的还有龚维强、李德文、李昔华、龚黔芬、陈立志等。

本书的编写工作,得到了重庆工商大学计算机科学与信息工程学院院长黄正洪教授的指导和关怀,在此表示诚挚谢意!

由于时间紧迫,加之编者水平有限,书中难免有不足之处,恳请读者批评指正!

编 者

2011年10月

目 录

前言

第 1 章 C 语言程序设计开发环境	1
1.1 Turbo C 集成开发环境	1
1.2 Visual C++ 集成开发环境简介	4
第 2 章 C 语言程序设计实验指导	12
实验 1 C 语言概述	12
实验 2 C 语言基本数据类型	15
实验 3 C 语言表达式	18
实验 4 顺序结构程序设计	21
实验 5 选择结构程序设计	23
实验 6 基本循环结构程序设计	26
实验 7 嵌套循环结构程序设计	29
实验 8 循环结构综合应用程序设计	32
实验 9 函数基本应用	36
实验 10 变量的存储属性与编译预处理	40
实验 11 一维数组	43
实验 12 二维数组	47
实验 13 字符数组	51
实验 14 指针与函数	54
实验 15 指针与数组	58
实验 16 结构体数据类型	62
实验 17 结构体数据类型的应用	67
实验 18 共同体与枚举类型	74
实验 19 文件	78
实验 20 综合程序设计	84
第 3 章 C 语言程序设计学习指南	86
3.1 C 语言程序设计概述	86
3.2 C 语言的基本数据类型与表达式	89
3.3 顺序结构程序设计	98
3.4 选择结构程序设计	102
3.5 循环结构程序设计	109
3.6 函数与编译预处理	124
3.7 数组	148
3.8 指针	164

3.9 结构体、共同体与枚举类型	180
3.10 文件	195
参考文献	208
附录 1 常用算法	209
算法 1 求最大公约数	209
算法 2 求最小公倍数	209
算法 3 求阶乘	210
算法 4 判断一个数是否为素数	211
算法 5 求 Fibonacci 数列	211
算法 6 选择排序	213
算法 7 冒泡排序	214
算法 8 插入排序	217
算法 9 顺序查找	219
算法 10 二分查找	220
算法 11 求数组的最小值	221
算法 12 将数组元素逆置	221
算法 13 矩阵转置	222
算法 14 两个字符串的连接	223
算法 15 两个字符串的比较	224
算法 16 字符串的插入	225
算法 17 字符串的删除	226
算法 18 有序数列的插入	227
算法 19 有序数列的删除	227
算法 20 单链表的建立	228
算法 21 单链表的遍历	229
算法 22 单链表的查找	230
算法 23 单链表的插入	230
算法 24 单链表的删除	231
算法 25 十进制正整数转换二至十六进制数	233
附录 2 C 语言程序设计模拟试题	235

第 1 章 C 语言程序设计开发环境

1.1 Turbo C 集成开发环境

1.1.1 Turbo C 2.0 的安装

Turbo C 2.0 是由美国研发的一个基于 DOS 平台的 C 语言程序开发软件，它集程序、编译、连接、调试为一体，具有速度快、效率高、功能强、使用方便等优点。

1. Turbo C 2.0 的系统配置

Turbo C 2.0 作为一个基于 DOS 平台的编译系统，要求的系统资源很小，仅占用 300K 左右的内存，而系统文件占用的磁盘空间不到 3MB。

2. Turbo C 2.0 的安装

在使用 Turbo C 2.0 之前，必须先熟悉其使用环境，而 Turbo C 使用环境的建立可以通过 Turbo C 系统提供的安装程序 install.exe 直接引导安装。但是也有些 Turbo C 编译系统通过直接复制运行，不需专门安装。

1.1.2 Turbo C 2.0 环境下调试程序的过程

1. 调试程序的过程

无论利用任何开发工具调试程序，都要经历编辑、编译、连接、运行等工作，在此过程中会产生相关的源程序文件、目标文件和可执行文件，如图 1-1 所示。其中编译的作用是将源程序编译成目标代码程序，连接的作用是将目标文件和库函数、相应的包含文件等连接成可执行文件（exe 文件）。

2. Turbo C 2.0 工作环境

为确保程序调试的正确性，必须对源程序文件、目标文件和可执行文件的保存位置进行设定，同时还必须明确 Turbo C 的“包含文件”（Include 子目录）和库函数（Lib 子目录）的保存位置。因此，在程序编辑之前，应先确定 Turbo C 的工作环境。要确定 Turbo C 系统的工作环境，需要利用菜单栏上的 Options 菜单中的 Directories 选择项，如图 1-2 所示。其中 Directories 选择项中各子项的含义见表 1-1。

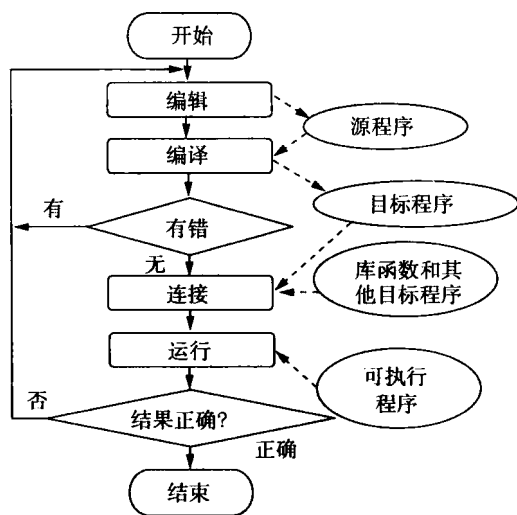


图 1-1 运行 C 程序的过程

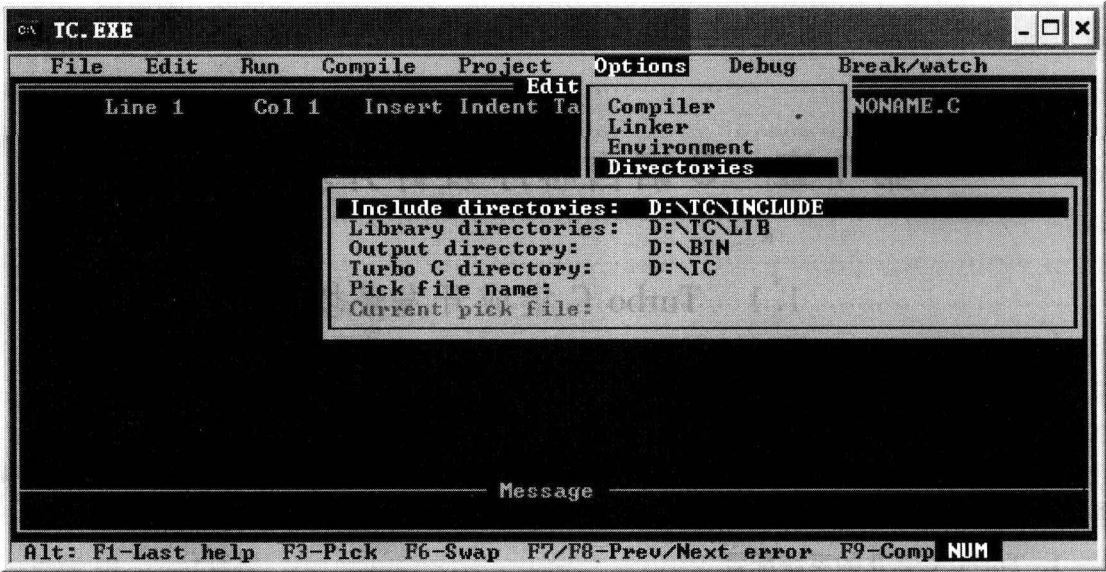


图 1-2 Turbo C 工作环境设置

表 1-1 Directories 选择项中各子项的含义

选 项	含 义
Include directories	定义包含文件所在目录
Library directories	指定库文件所在目录
Output directory	输出文件存放的目录
Turbo C directory	指定 Turbo C 系统文件所在目录

根据需要对这些选项进行设置，并保存这些设置信息，其保存的方法是通过 Options 菜单中 Save Options 选项来完成。

3. 利用 Turbo C 2.0 调试简单 C 程序

在 Turbo C 2.0 中，调试一个简单 C 程序的步骤如下。

```
#include "stdio.h"
void main()
{int k,x,y,temp;
 printf("请输入两个正整数:");
 scanf("%d,%d",&x,&y);
 temp = x*y;
 while(k = x%y)
 {x = y;y = k; }
 temp = temp/y;
 printf("最大公约数为:%d\n",y);
 printf("最小公倍数为:%d\n",y,temp);}
```

其步骤如下。

1) 新建或调入一个C程序

新建一个C程序，可以通过File菜单中的New选项来完成。在Turbo C所提供的编辑环境中，用户可将已编好的源程序逐行输入，发现错误可随时修改。

程序输入结束后，应先将程序保存起来。程序的保存可通过File菜单中的Save选项来完成，也可以用键盘上的F2键来进行。在保存源程序时，默认文件名为NONAME.C，但是建议用户自己指定文件名。

当要编辑一个已经存在的程序时，可以选择File菜单下的Load选项，在Load File对话框中，输入准备调入的文件路径和文件名。

2) 编译和连接

编辑、保存好C语言源程序后，可以通过Compile菜单下的Compile to OBJ选项来实现C源程序的编译，如图1-3所示。如果编译有错误，再修改，再编译，直到编译通过为止，此时，将生成目标代码程序。既可通过Compile菜单下的Link EXE file选项（图1-4），也可通过Compile菜单下的Make EXE file选项或F9键完成编译和连接，最终生成可行文件。

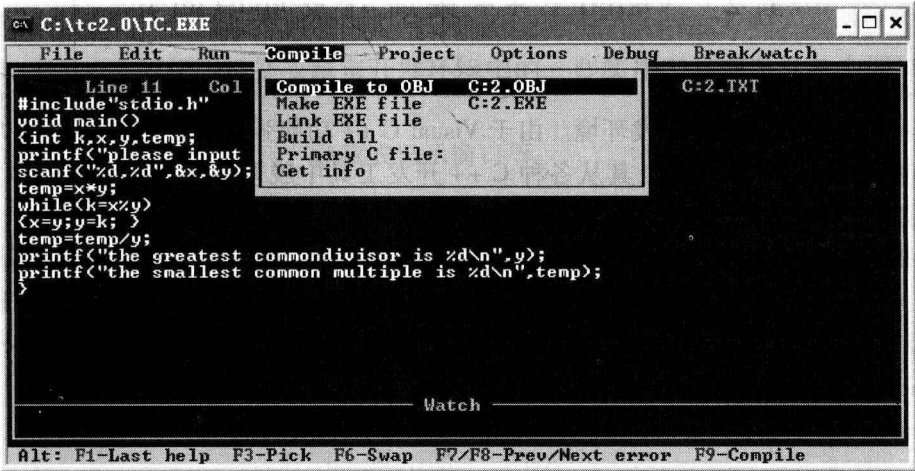


图 1-3 C 程序的编译

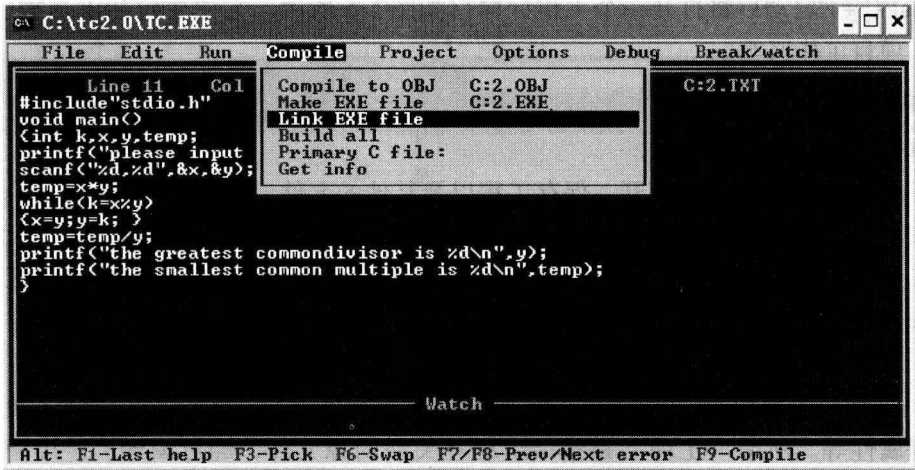


图 1-4 C 程序的连接

3) 运行

当 C 语言程序通过编译和连接后，最终将生成可执行文件。此时，可以在 Run 菜单中执行 Run 命令或使用 Ctrl + F9 执行可执行文件，可通过 Run 菜单中的 User screen 选项来查看运行结果。

4) C 语言程序的动态调试

C 语言程序的编译和连接过程中没有出现错误，不表示所得运行结果一定正确。编译系统只能检查出语法错误，而无法检查出逻辑错误。因此，当发现结果出现问题时，也需要对源程序进行调试。常见的调试方法有两种：一是单步执行法，即按一次 F7 键执行一行，用户可以逐行检查相关的变量和表达式，以便发现问题。同时可以在编辑窗口中查看变量值的变化情况；二是设置断点法，对较长的程序，可在程序中设置若干个断点，将程序划分成若干个小块，一个小块一个小块地执行检查。用 Ctrl + F8 完成断点行的设置，反复使用 Ctrl + F9 键可实现从程序的第 1 行语句到第 1 断点或从一个断点到另一个断点的执行。

1.2 Visual C++ 集成开发环境简介

Visual C++（简称 VC 或 VC++）是 Microsoft 公司推出的目前使用极其广泛的基于 Windows 平台的可视化集成开发环境。由于 Visual C++ 功能强大、使用灵活、可扩展性强，同时具有强大的 Internet 支持，其从各种 C++ 开发工具中脱颖而出，成为目前最为流行的 C++ 集成开发环境。

Visual C++ 集代码的编辑、编译、连接、调试于一体，以方便、友好的界面呈现在程序员面前。它不仅可以直接运行 C 和 C++ 程序，而且还提供了 MFC（Microsoft Foundation Class）及开发工具，用于快速地创建 Windows 应用程序框架。目前，Visual C++ 已经成为开发 Windows 应用程序的主流工具。

Visual C++ 可以单独安装，也可与 Visual Studio 同时安装。

1.2.1 Visual C++ 集成开发环境

Visual C++ 启动后将打开一个工作区窗口如图 1-5 所示。它主要包括工具栏、菜单栏、输出区和项目工作区等。

1. 菜单功能

菜单栏主要包括如下选项。

- (1) 文件：用来创建、打开、保存工程以及其他的文件。
- (2) 编辑：用来进行拷贝、粘贴、剪切、删除、查找、恢复文件等操作。
- (3) 查看：用来改变窗口和工具栏的显示方式，激活调试时所使用的输出区、调试窗口、工作区、属性区等。
- (4) 插入：主要用于项目和资源的创建和添加，如插入新建类、资源等。
- (5) 工程：主要用于设置活动工程（激活指定的项目）、向工程添加组件和外部源文件、工程属性等。所谓“工程”是指一组相互关联的源文件，经过编译、连接后产生一个可执行的 Windows 程序或动态链接库函数。
- (6) 编译：用于对应用程序进行编译、连接和运行，其命令主要包括编译、构件、调

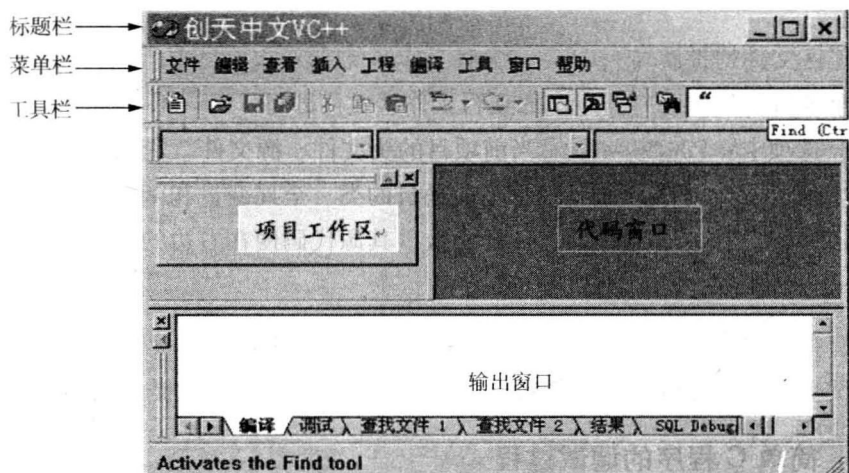


图 1-5 Visual C++ 6.0 窗口

试、运行等。

(7) 工具：主要用于选择或制定开发环境中的一些实用工具来激活各个调试窗口，改变各个窗口的显示模式。

(8) 窗口：主要用于排列、隐藏或显示窗口等。

2. 工具栏的功能

工具栏是一系列工具按钮的组合，是一种图形化的操作界面，具有直观快捷的特点。通过工具栏，可以迅速地使用常用的菜单命令。工具栏上的按钮通常与一些菜单命令相对应，提供了一种执行经常使用的快捷方法，如图 1-6 所示。

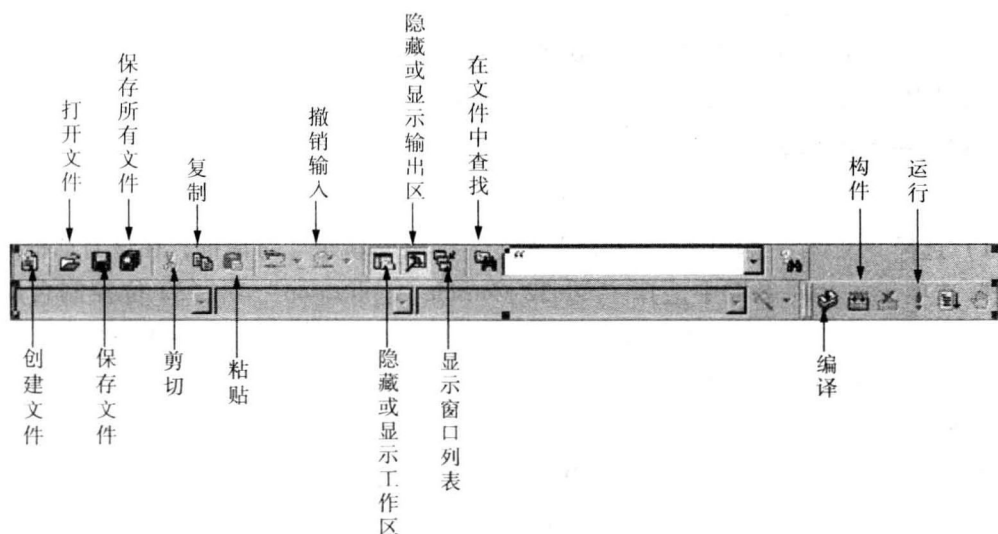


图 1-6 工具栏功能

3. 项目工作区

项目工作区有 Class View、Resource View、File View 三个选项卡，分别用来浏览当前项目所包含的类、资源和文件。如果选择“Win32 Console Application”程序，则没有 Resource

View 选项卡。

项目工作区文件的扩展名为 .dsw, 含有工作区的定义和项目中所包含文件的所有信息。若要打开一个项目, 只需要打开对应的项目工作区文件即可。

File View 选项卡: File View 显示当前项目的头文件、源文件、资源文件等。

Class View 选项卡: Class View 显示当前项目的类、全局变量和函数。

在“Class View”中, 双击 main () 函数, 其源代码便在右边的代码窗口中显示。

4. 输出区

输出区有多个选项, 如编译、调试、查找文件等。在编译、连接时, 在输出区显示相关信息, 供调试程序用; 在编译时, 将在输出窗口显示相应错误信息提示, 以使用户修改。

1.2.2 简单 C 程序的调试过程

在 Visual C++ 中, 编写、运行一个简单 C 程序一般分为三个步骤。

- (1) 创建一个空工程。
- (2) 创建一个 C 源程序文件。
- (3) 进行编译、连接、运行。

下面通过实例说明在 Visual C++ 环境中, 如何编辑、编译、连接、运行 C 语言程序。

以控制台应用程序为例介绍。所谓控制台应用程序是指那些需要与传统的 DOS 操作系统有某种程度的兼容, 同时又不需要为用户提供完善界面的程序, 也就是说, 在 Windows 环境中运行的 DOS 程序。

在 Visual C++ 中, 应用程序向导 AppWizard 能帮助快速创建一些常见的应用程序类型框架。

例 求正整数 m, n 中的最大公约数和最小公倍数。

准备工作: 先在 E 盘上建立一个文件夹 “student-c++”。

1. 创建一个空工程

其步骤如下。

(1) 选择“文件”菜单中的“新建”命令, 弹出“新建”对话框; 在该对话框中选定“工程”选项, 并在列表框中选择“Win32 Console Application”项 (32 位控制台应用程序); 并在“工程”文本框中输入工程名 exam; 在“工程位置”上, 单击下拉式按钮, 选择存放项目文件的磁盘及文件夹: E:\student-c++\exam (用户可根据需要选择创建新的工程文件的位置); 选定单选按钮“R 创建新工作区”, 按“确定”按钮, 如图 1-7 所示。

(2) 随后弹出一个询问项目类型 Win32 应用程序向导对话框, 如图 1-8 所示, 选择“An empty project”, 并选择“完成”, 显示新建工程的有关信息。

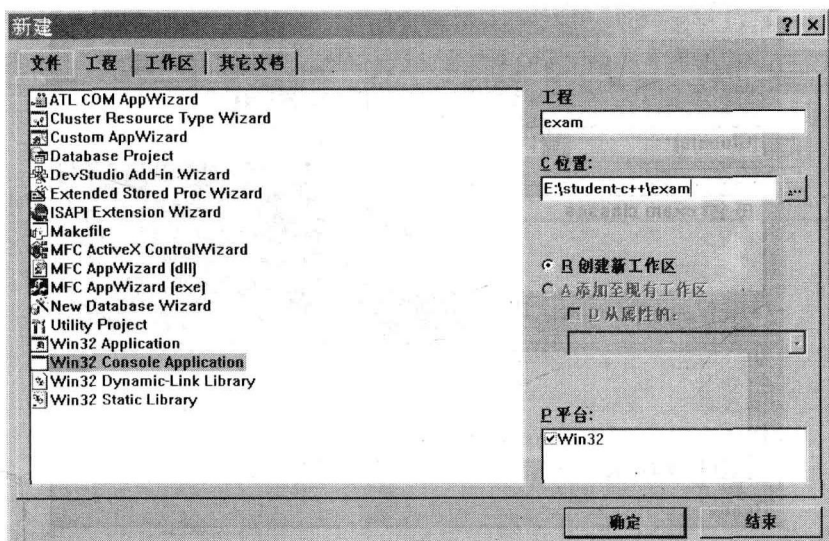


图 1-7 “新建”对话框

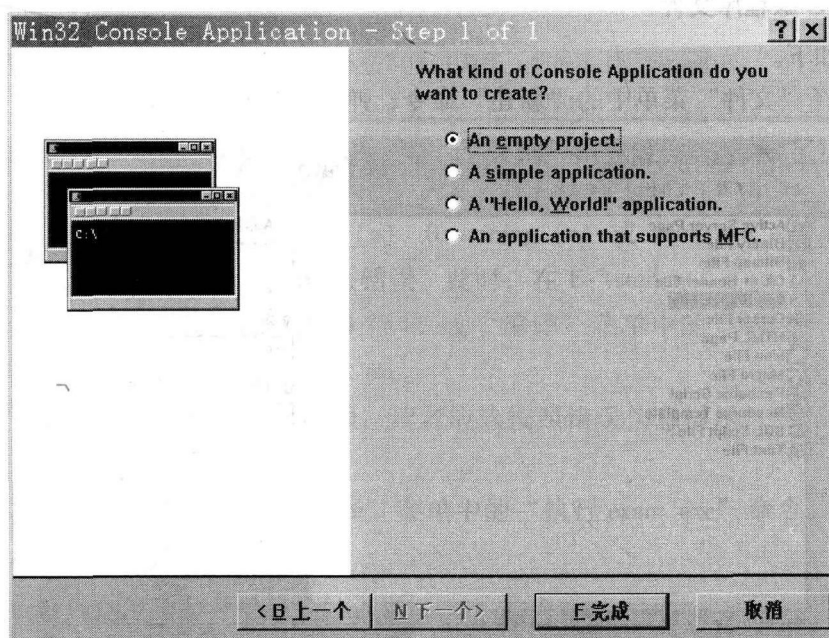


图 1-8 选择需要编写的应用程序类型

(3) 在“新建工程信息”对话框中，选择“确定”按钮后，创建空工程的任务结束，显示如图 1-9 所示的窗口。

此时为工程 exam 创建了 E:\student - c ++ \exam 文件夹，并在其中生成了项目文件 exam. dsp、工作区文件 exam. dsw 及 debug 文件夹。其中项目文件 exam. dsp 存储了当前项目的特定信息，如项目设置；工作区文件 exam. dsw 含有工作区的定义和项目中所包含文件的所有信息；debug 文件夹存放编译、连接过程中生成的中间文件及最终生成的可执行文件，主要包括 exam. obj（由编译后生成的目标代码文件）、exam. exe（由连接后生成的可执行文件）等。

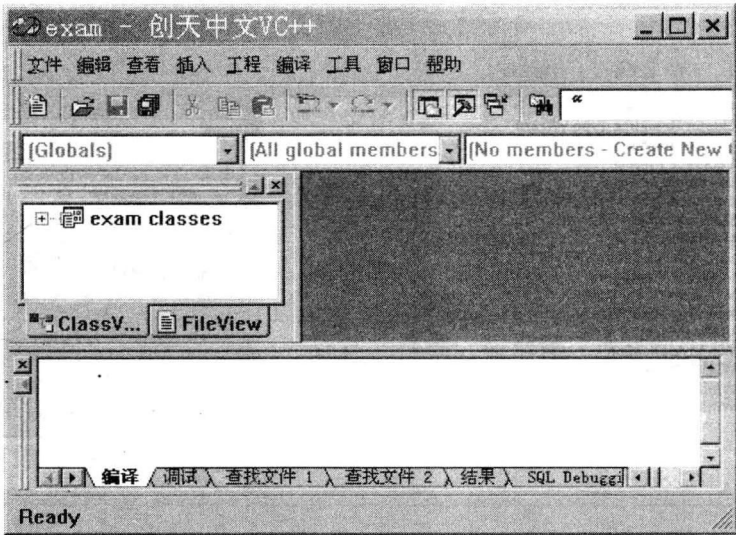


图 1-9 空工程 exam 窗口

2. 创建 C 源程序文件

其步骤如下。

(1) 选择“文件”菜单中的“新建”命令，弹出如图 1-10 所示的对话框。

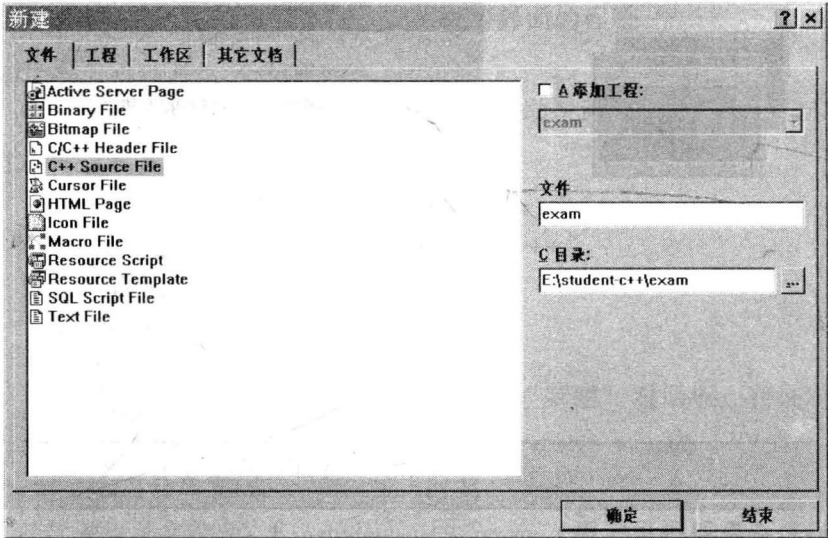


图 1-10 新建“C/C++ Source File”对话框

(2) 在“文件”选项卡中，选定“C++ Source File”，并输入源程序文件 exam，按“确定”按钮。

(3) 在代码窗口中输入、编辑源程序，并保存生成 exam.cpp 文件。

3. 编译、连接、运行

其步骤如下。

(1) 按快捷键 Ctrl + F7（生成 exam.obj）或 F7（生成 exam.exe）或选定“编译”菜单中“编译”或“构件”命令，对源程序 exam.cpp 进行编译、连接，在输出窗口中将显示编

译、连接有关的信息，如图 1-11 所示。

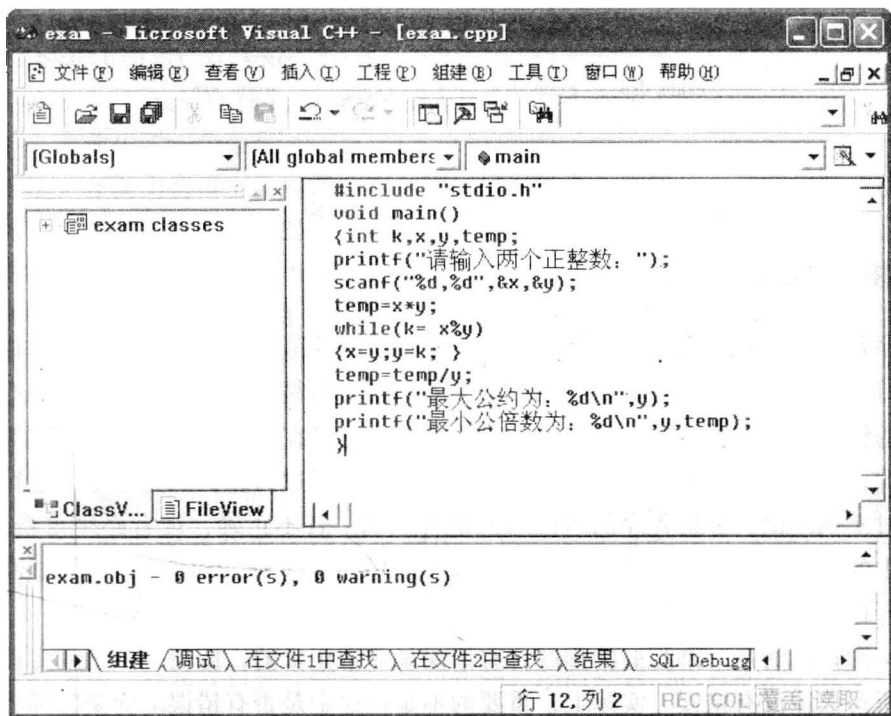


图 1-11 exam 项目工作区及输出窗口内容

若显示的信息是“exam. obj - 0 error (s), 0 warning (s) 或 exam. exe - 0 error (s), 0 warning (s)”则表示源程序无语法、词法错误。此时，在 E:\student - c ++ \exam\debug 文件夹下将生成 exam. obj 等文件信息。当执行 F7 或“编译”菜单中的“构件”命令时，在 debug 文件夹下即可生成可执行文件 exam. exe。

若有错误信息提示，表示源程序有误，读者应该按照错误信息提示，对源程序进行修改，直到编译通过为止。

(2) 按快捷键 Ctrl + F5 或选择“编译”菜单中的“执行 exam. exe”命令，即执行可执行文件 exam. exe。其结果如图 1-12 所示。

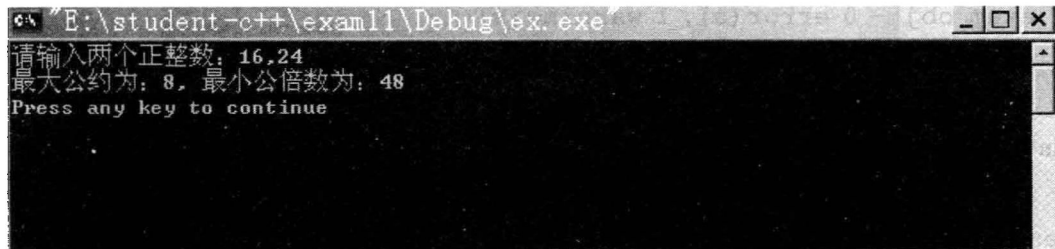


图 1-12 输出结果显示窗口

(3) 当编译通过后，读者也可对程序进行单步调试，按快捷键 F10，在输出窗口将显示每一步执行后变量值的变化情况。

通过上述工作后, E:\student-c ++ \exam 文件夹中的目录结构如图1-13 所示。

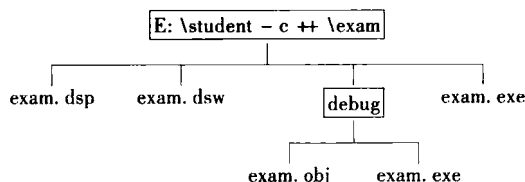


图 1-13 工程文件目录结构

1. 2. 3 调试程序中的错误

在程序开发过程中, 程序的调试是一个非常重要的环节, 特别是代码较长的程序, 其调试工作量特别大。因此, 熟悉程序调试过程中的错误信息是提高程序调试效率的有效途径。

程序的错误一般有两类。一是语法、词法错误, 它是在编译阶段中出现。如果源程序中有语法错误, 读者必须在编译阶段进行修改, 直到编译通过为止; 二是程序的逻辑错误, 读者必须通过实例测试, 分析程序的运行结果是否正确。若不正确, 读者必须重新分析算法, 检查程序中的逻辑错误, 重新修改程序。在检查程序的过程中, 读者可通过设置断点, 追踪程序的运行过程来发现错误。

需要特别指出的是, 刚写好的程序含有错误是非常正常的, 即使是熟悉的专业程序员也难以保证所写程序没有错误。实际上, 重要的不是程序中是否有错误, 而是怎样将这些错误找出来并快速进行修改。

1. 语法错误

在编译、连接过程中, 若程序有语法错误, 系统将在输出窗口中显示错误信息。错误信息的一般形式为

文件名 (行号): 错误代码: 错误内容

除语法错误信息外, 还有警告 (warning) 信息。若只有警告信息而没有错误信息, 程序可正常运行, 但可能存在某种潜在的错误。

例如:

```
E: \student - c ++ \exam \exam. cpp (12): warning C4508: 'main': function
should return a value; 'void' return type assumed
```

```
exam. obj - 0 error(s), 1 warning(s)
```

```
exam. obj - 2 error(s), 0 warning(s)
```

表示在 E:\student - c ++ \exam\exam. cpp 文件的第 12 行有一个 C4508 的警告信息: main 应该有一个返回值。

```
E: \student - c ++ \exam \exam. cpp (8) : error C2065: 'whil' : undeclared
identifier
```

```
E: \student - c ++ \exam \exam. cpp (9) : error C2143: syntax error : miss-
ing ';' before '{'
```

Error executing cl.exe.

表示在程序中将“whlie”误写为“whil”, 编译器在编译的时候, 就把它当做未定义标识符。

E:\student - c ++ \exam\exam.cpp (7) : error C2146: syntax error : missing ';' before identifier 'k'

Error executing cl.exe.

exam.obj - 1 error(s), 0 warning(s)

表示在 k 之前少了一个分号。

修改方法：在输出窗口中，用鼠标双击相应错误信息，系统将自动定位到源程序中错误所在的语句行，读者可根据错误提示修改程序。

如根据上述错误提示信息，可将上述源程序中的 main（）前加“void”；将上述源程序中的“whil”改写为“whlie”；将上述源程序中的第7行变量 k 之前加“;”即可。

值得指出的是，在编译程序的过程中，经常出现由于一个错误引起若干条错误信息，在修改时，实际上只修改一处即可。

2. 逻辑错误

程序的逻辑错误，不仅难以发现，而且这种错误是致命的，它将直接影响程序的执行结果。因此，要求读者在程序调试过程中，当程序通过编译、连接后，要多通过测试数据对程序进行测试。若程序的运行结果与实际结果不相符合，就必须检查程序中的逻辑错误。

检查逻辑错误，首先分析程序的算法是否正确，然后再通过设置断点，让程序执行到断点在 Debug 窗口中观察程序中变量的值，来分析程序中的错误。具体操作是：单击要设置为断点的行，再使用快捷键 F9 设置断点；使用快捷键 F5，使程序执行到断点，此时在窗口下增加了一个 Debug 菜单、Variables 窗口（显示的是程序中各变量的值）及 Watch 窗口（可输入变量或表达式，以观察其值），如图 1-14 所示。

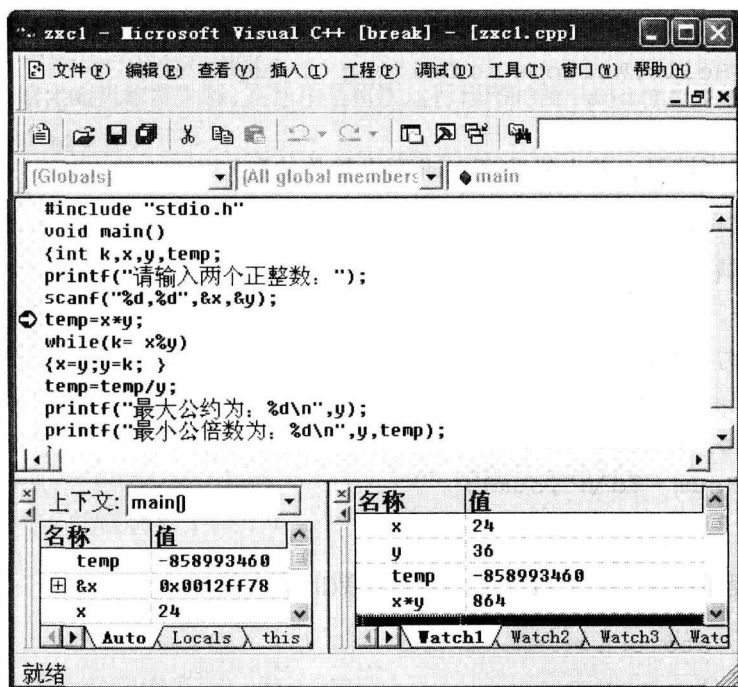


图 1-14 设置、运行断点窗口

第2章 C语言程序设计实验指导

实验1 C语言概述

【实验目的】

- (1) 通过简单的C程序的调试熟悉 Turbo C 2.0 或 Visual C ++ 环境。
- (2) 初步掌握C程序的基本构成。
- (3) 熟悉C程序的实现过程和方法。
- ①掌握编辑、编译、连接、运行程序的过程和方法。
- ②了解常见的两种出错信息提示 (Error/Warning)。

【实验内容】

1. 验证性实验

- (1) 分析下面程序结果，并上机验证（字符数据的简单输出）。

```
#include <stdio.h>
void main()
{
    printf("Hello,World!\n");
    printf("Hello,China!\n");
    printf("Hello,Welcome to Beijing Olymnpic\n");
}
```

- (2) 分析下面程序结果，并上机验证（求两个整数的和）。

```
#include <stdio.h>
void main()
{
    int a,b,sum;
    a=123;b=456;
    sum=a+b;
    printf("sum=%d\n",sum);
}
```

- (3) 分析下面程序结果，并上机验证（求两个数的最大值）。

```
#include <stdio.h>
void main()
{
```