

基础教育规划教材·精品系列

C++程序设计实验教程

C++ CHENGXU SHEJI SHIYAN JIAOCHENG



史巧硕 刘洪普 主编



◆ 免费提供教学资源

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

基础教育规划教材·精品系列

C++程序设计实验教程

主 编 史巧硕 刘洪普

副主编 朱怀忠 毕晓博 金 迪 刘晓星

参 编 郭迎春 李建晶 梁艳红 路 静

内 容 简 介

本书是与主教材《C++ 程序设计教程》(史巧硕 朱怀忠主编)配套使用的上机实验指导用书,是编者多年教学实践经验的总结。全书包括 21 个实验,实验的例题和内容与主教材相应章节呼应,可以方便教师有计划有目的地安排学生上机操作,达到事半功倍的学习效果。本书内容丰富,例题详尽,部分程序的案例取自实际应用。

本书适合作为高等院校“C++ 程序设计课程”的辅助教材,也可作为计算机培训班的培训教材,还可作为广大软件开发人员和自学者学习 C++ 程序设计语言的参考用书。

图书在版编目(CIP)数据

C++ 程序设计实验教程/史巧硕,刘洪普主编. —北京:中国铁道出版社, 2017.8

高等院校计算机基础教育规划教材-精品系列

ISBN 978-7-113-23248-1

I. ①C… II. ①史… ②刘… III. ①C 语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 182639 号

书 名: C++ 程序设计实验教程

作 者: 史巧硕 刘洪普 主编

策 划: 魏 娜 周海燕

读者热线: (010) 63550836

责任编辑: 周海燕 李学敏

封面设计: 付 巍

封面制作: 刘 颖

责任校对: 张玉华

责任印制: 郭向伟

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 北京鑫正大印刷有限公司

版 次: 2017 年 8 月第 1 版 2017 年 8 月第 1 次印刷

开 本: 880 mm×1 230 mm 1/16 印张: 11.75 字数: 357 千

书 号: ISBN 978-7-113-23248-1

定 价: 32.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

前言 *Preface*

上机实验是学习计算机程序设计语言的重要环节。学生通过实际上机编程的演练，可以加深对编程规则及理论知识的理解，同时对培养自学能力、锻炼实际的编程能力也起着极为重要的作用。为此，我们编写了本书。本书是与主教材《C++程序设计教程》（史巧硕 朱怀忠主编）配套使用的实验教材，同时也可以与其他C++程序设计教科书配合使用。

本书共有21个实验，每个实验包括实验目的、范例分析、实验内容和问题讨论等内容。实验一介绍Visual C++ 6.0的开发环境，并通过简单的例子介绍了上机操作的步骤及在Visual C++ 6.0中调试C++程序的一般方法；实验二~实验十四涵盖Visual C++的数据类型、程序的基本结构与流程控制语句、数组和指针的操作、函数与预处理、结构体和联合体的操作，这些内容也是构成C++程序设计的基础内容；实验十五~实验十七介绍C++面向对象方面的知识，包括类与对象的操作、继承与虚函数、运算符重载等；实验十八~实验二十一介绍Visual C++ 6.0的Windows编程的基础知识。

本书的作者长期从事C++语言程序设计课程的教学工作，并曾利用C++、Visual C++语言开发了多个软件项目，因此有着丰富的教学经验和较强的科研能力，对C++有着较深入的理解。为了实现理论联系实际，达到良好的教学效果，作者精心选择了实验的例题和内容，并与教材各章相呼应，以方便教师有计划、有目的地安排学生上机操作，从而达到事半功倍的教学效果。另外，在实验中，还有针对性地提供了一些接近实际要求或直接取自实际应用的较为完整的程序案例，教师可以以这些程序为范本，进行综合练习或组织课程设计的题目。教师若能配合C++程序设计教材，有计划地按本书要求安排实验上机，可迅速提高学生的实际操作能力。

本书由史巧硕、刘洪普主编，并负责全书的总体策划与统稿、定稿工作，朱怀忠、毕晓博、金迪、刘晓星任副主编，各章编写分工如下：实验一由李建晶编写，实验二、实验三、实验四、实验五由史巧硕编写，实验六、实验七、实验八由朱怀忠编写，实验九、实验十、实验十一、十二由刘洪普编写，实验十三、实验十四由毕晓博编写，实验十五、实验十六由郭迎春编写，实验十七由路静编写，实验十八、实验十九由金迪编写，实验二十由刘晓星编写，实验二十一由梁艳红编写。

在本书编写过程中，参考了大量文献资料，在此向这些文献资料的作者深表感谢。

由于时间仓促，编者水平有限，书中难免有不当和欠妥之处，敬请各位专家、读者不吝批评指正。

编 者

2017年5月

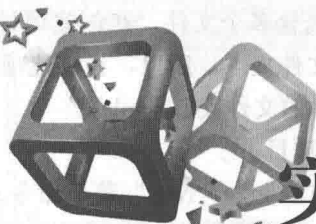
目录 Contents

实验一 Visual C++ 6.0开发环境及简单应用程序的创建	1
一、实验目的	1
二、相关知识	1
三、实验内容	9
四、问题讨论	9
实验二 输入/输出与顺序结构	11
一、实验目的	11
二、范例分析	11
三、实验内容	14
四、问题讨论	15
实验三 选择结构程序设计	16
一、实验目的	16
二、范例分析	16
三、实验内容	23
四、问题讨论	26
实验四 循环结构程序设计	27
一、实验目的	27
二、范例分析	27
三、实验内容	32
四、问题讨论	35
实验五 典型程序设计	36
一、实验目的	36
二、范例分析	36
三、实验内容	43
四、问题讨论	45
实验六 一维数组	46
一、实验目的	46
二、范例分析	46
三、实验内容	52
四、问题讨论	55
实验七 二维数组与字符数组	56
一、实验目的	56
二、范例分析	56
三、实验内容	63
四、问题讨论	66

实验八 指针	67
一、实验目的	67
二、范例分析	67
三、实验内容	70
四、问题讨论	76
实验九 函数及其调用	77
一、实验目的	77
二、范例分析	77
三、实验内容	80
四、问题讨论	82
实验十 函数与指针	83
一、实验目的	83
二、范例分析	83
三、实验内容	90
四、问题讨论	92
实验十一 函数嵌套调用及函数重载与带默认参数的函数	93
一、实验目的	93
二、范例分析	93
三、实验内容	99
四、问题讨论	101
实验十二 作用域和预处理	102
一、实验目的	102
二、范例分析	102
三、实验内容	106
四、问题讨论	111
实验十三 结构体与共用体	112
一、实验目的	112
二、范例分析	112
三、实验内容	115
四、问题讨论	118
实验十四 结构体数组和结构体指针变量	119
一、实验目的	119
二、范例分析	119
三、实验内容	123
四、问题讨论	128
实验十五 类与对象	129
一、实验目的	129
二、范例分析	129
三、实验内容	135
四、问题讨论	136
实验十六 继承与虚函数	137
一、实验目的	137
二、范例分析	137



三、实验内容	140
四、问题讨论	142
实验十七 运算符重载	143
一、实验目的	143
二、范例分析	143
三、实验内容	146
四、问题讨论	146
实验十八 创建基于对话框的MFC应用程序	147
一、实验目的	147
二、范例分析	147
三、实验内容	161
四、问题讨论	161
实验十九 多对话框应用程序	162
一、实验目的	162
二、范例分析	162
三、实验内容	170
四、问题讨论	170
实验二十 菜单	171
一、实验目的	171
二、范例分析	171
三、实验内容	174
四、问题讨论	174
实验二十一 创建单文档应用程序	175
一、实验目的	175
二、范例分析	175
三、实验内容	179
四、问题讨论	179
参考文献	180



实验一

Visual C++ 6.0开发环境 及简单应用程序的创建

一、实验目的

- (1) 了解Visual C++ 6.0开发环境。
- (2) 掌握在Visual C++ 6.0中编写控制台应用程序的过程。
- (3) 熟悉Visual C++ 6.0开发环境中的一些常用操作。
- (4) 学习如何在Visual C++ 6.0中调试C++程序。

二、相关知识

1. Visual C++ 6.0集成开发环境简介

- (1) 启动Microsoft Visual C++ 6.0。

选择“开始”→“所有程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”菜单命令，或双击桌面上名为“Microsoft Visual C++ 6.0”的快捷方式图标，从而进入Microsoft Visual C++ 6.0集成开发环境。

- (2) Visual C++ 6.0主窗口。

Visual C++ 6.0主窗口包括标题栏、菜单栏、工具栏、工作区窗口、编辑窗口区、输出窗口和Build MiniBar工具栏，如图1.1所示。其中，Build MiniBar工具栏中的工具，在编译和连接程序时会经常使用。在工具栏区右击，可从快捷菜单中选择或取消工具栏、工作区窗口Workspace和输出窗口Output的显示。

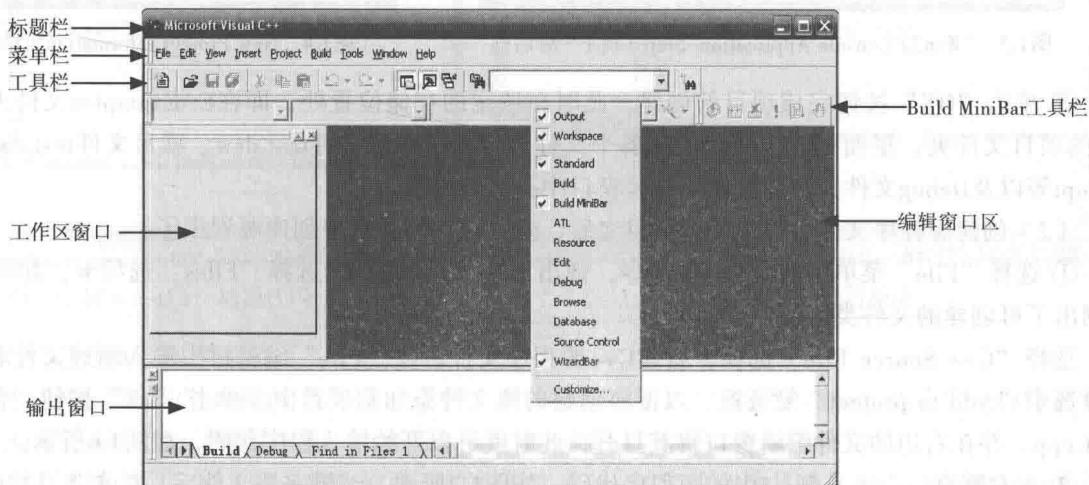


图1.1 主窗口

2. 在Visual C++中编写一个基于控制台的应用程序

一个项目包含多个文件，在创建项目时系统会在指定的文件夹中创建项目文件夹，并将项目中包含的文件存放在该文件夹下。所以，在创建项目前，最好创建一个文件夹存放编写的C++程序，这里预先在E盘建立了存放项目的文件夹Examples。

(1) 创建项目。

① 选择“File”→“New”菜单命令，弹出“New”对话框。在“New”对话框中，选择“Projects”选项卡（默认），如图1.2所示，从列出的项目类型清单中选择“Win32 Console Application”选项，创建一个基于控制台的应用程序；在“Project name”下的文本框中输入新建项目名，如test；在“Location”下的文本框中显示将要生成的项目文件夹的存放位置，可单击右侧的“...”按钮，修改项目文件夹的保存位置，否则默认存放位置为Visual C++的安装目录下的MyProjects文件夹。最后单击“OK”按钮，打开“Win32 Console Application-Step 1 of 1”对话框，如图1.3所示。

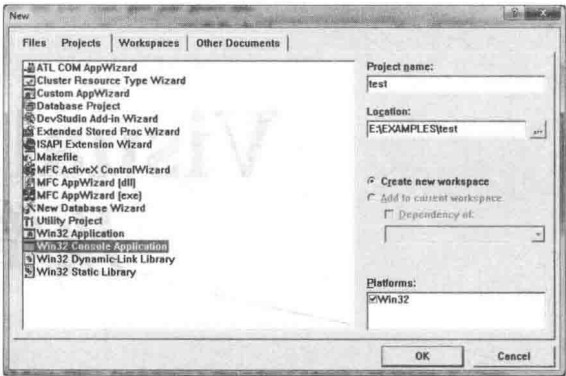


图1.2 “New”对话框

② 该对话框提供了四种项目的类型，选择不同的选项，意味着系统会自动生成一些程序代码，为项目增加相应的功能。这里选择“An empty project”选项，则生成一个空白的项目。单击“Finish”按钮，弹出“New Project Information”对话框，显示出将要创建项目的有关信息，如图1.4所示。

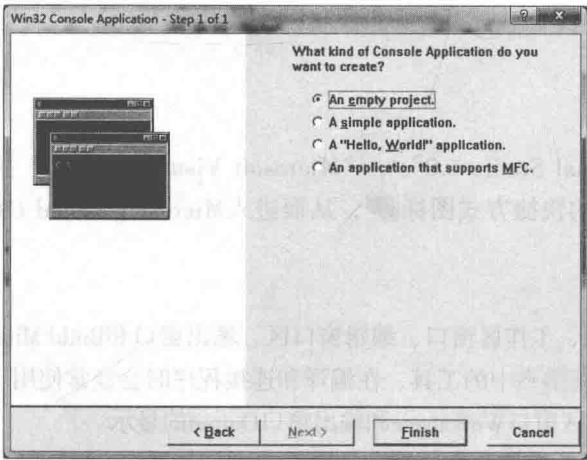


图1.3 “Win32 Console Application-Step 1 of 1”对话框

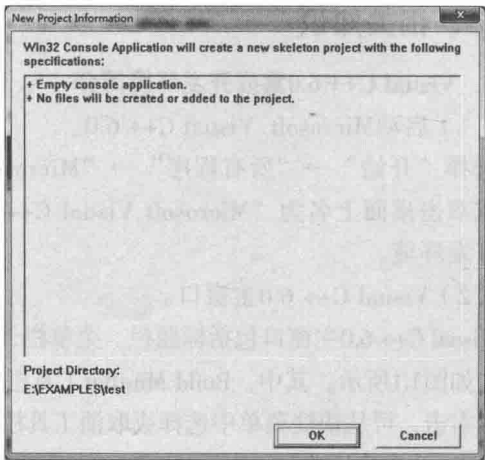


图1.4 New Project Information对话框

③ 单击“OK”按钮完成项目的创建，此时在选定的存储位置处，即在E:\Examples文件夹下创建名为test的项目文件夹，里面存放了项目中的各个文件，包括工作区文件test.dsw、项目文件test.dsp、参数文件test.opt等以及Debug文件夹，并且在工作区窗口中打开该项目。

(2) 创建源程序文件。创建一个项目之后，就可以在该项目中创建源程序了。

① 选择“File”菜单中的“New”命令，弹出“New”对话框，选择“Files”选项卡，如图1.5所示，其中列出了可创建的文件类型。

选择“C++ Source File”选项，新建C++源程序文件；在“File”编辑栏中输入新建文件名，如test1；注意选中“Add to project”复选框，以便将创建的源文件添加到项目中。单击“OK”按钮，创建了源文件test1.cpp，并在右边的文件编辑窗口将其打开，此时就可以开始输入程序代码，如图1.6所示。

② 在右侧窗口中输入例1.1中的源程序代码，如图1.7所示。编辑完源文件后，单击工具栏中的“Save”按钮保存源文件。

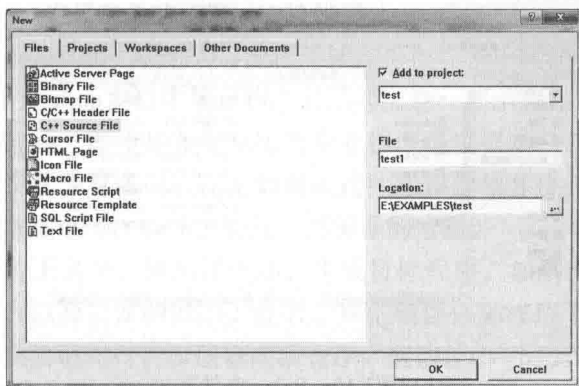


图1.5 “New”对话框中的“Files”标签

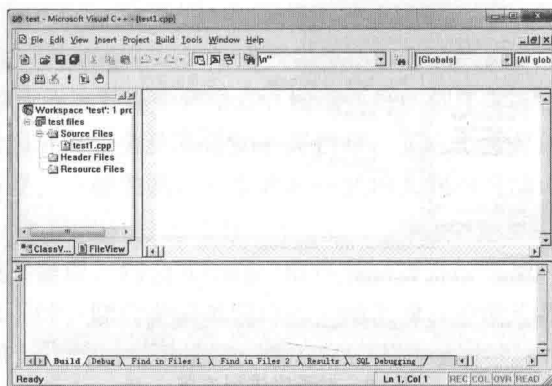


图1.6 在项目中创建了源程序文件test1.cpp

【例1.1】编写程序，该程序在控制台上显示信息“Hello world”。

源程序：

```
#include <iostream>           //包含头文件
using namespace std;         //使用std命名空间
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

注意：

单击文本编辑窗口的关闭按钮 ，可将正编辑的源文件test1.cpp关闭。若需要再次打开，可在工作区窗口选择“FileView”选项卡，将文件夹展开，双击“Source Files”，可看到源文件test1.cpp，双击该文件，即可将其在右面的编辑窗口打开。

(3) 编译源程序文件。

单击工具栏上的“Compile”按钮  或按【Ctrl+F7】组合键，则在Output窗口显示编译结果。此时源程序中没有错误，显示无错误，成功生成目标程序test1.obj，如图1.8所示。

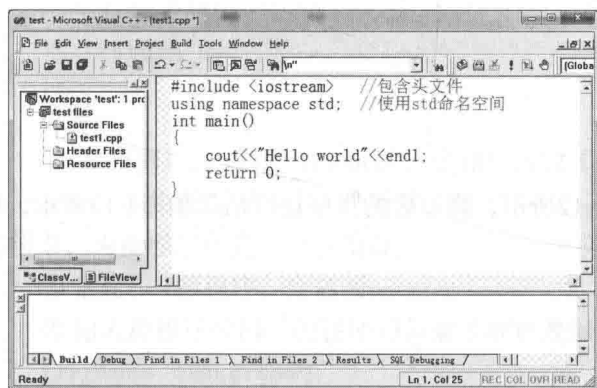


图1.7 输入test1.cpp的源代码

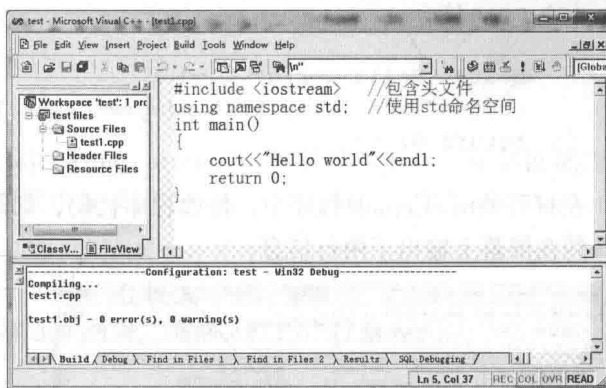


图1.8 对程序进行编译

(4) 连接生成可执行程序。

单击工具栏上的“Build”按钮  或按【F7】键，则连接生成可执行程序test.exe，可在Output窗口看到成功生成可执行程序的信息，如图1.9所示。如果连接出错，则在输出窗口显示错误信息。

编译生成的目标程序(.obj)和连接生成的可执行程序(.exe)存放在项目文件夹test下的Debug文件夹中，如图1.10所示。



图1.9 对程序进行连接

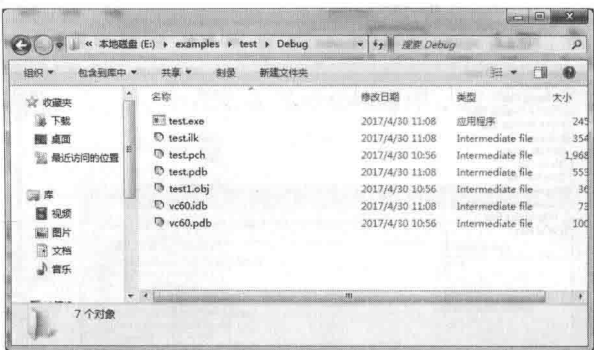


图1.10 Debug文件夹

(5) 运行程序。

生成可执行程序后，单击工具栏上的“Execute Program”按钮或按【Ctrl+F5】组合键，运行该程序。程序的运行结果会显示在一个DOS窗口，如图1.11所示。

(6) 打开和修改已有程序。

若要打开和修改以前编写好的程序，需要用Visual C++重新打开，在完成修改后，再编译、连接、运行。一个程序对应一个项目，要打开一个程序，就是要打开它对应的项目的工作区文件（.dsw），并不是打开它的源程序文件（.cpp），请初学者一定要注意。

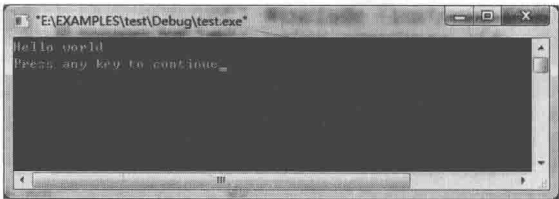


图1.11 程序运行结果

选择“File”菜单中的“Open Workspace”命令，打开“Open Workspace”对话框，选择驱动器、文件夹和项目工作区文件，单击“打开”按钮即可打开程序。或在项目文件夹中双击相应的工作区文件（扩展名为.dsw，例test.dsw），也可打开程序。

【例1.2】打开例1.1中的test.dsw，修改test1.cpp，在控制台上显示两行信息“Hello world”和“This is my first program”。

源程序：

```
#include <iostream> //包含头文件
using namespace std; //使用std命名空间
int main()
{
    cout<<\"Hello world\"<<endl;
    cout<<\"This is my first program\"<<endl;
    return 0;
}
```

在打开的test1.cpp源程序中，修改程序代码，如图1.12所示，修改后的程序运行结果如图1.13所示，可以看到在屏幕上输出了两行信息。

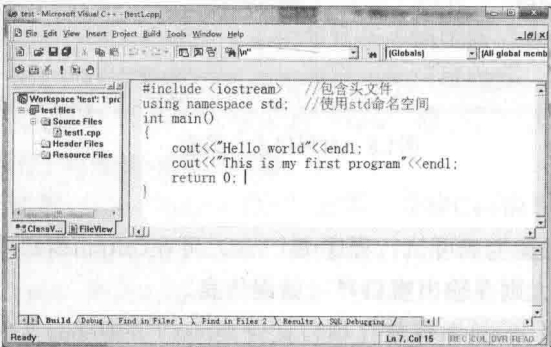


图1.12 修改后的源程序

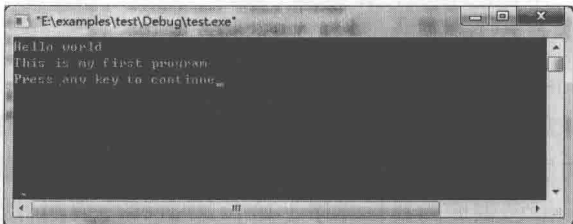


图1.13 程序运行结果



3. 查看和修改编译、连接错误

编译的目的是将C++源程序转换为机器指令代码。在编译过程中,如果遇到程序中有语法错误,则会在底部的Output窗口中显示相应的错误信息,提示程序员修改程序。作为初学者,刚编写好的程序含有错误是很正常的,即使是非常熟练的专业程序员也很难一次就编写出完全没有错误的源程序。实际上重要的不是程序中是否有错误,而是如何将这些错误找到并进行修改。一般来说,一个源程序从输入到通过编译,往往要重复很多次编译、修改、再编译的过程。

若无错误,则编译成功,生成目标程序。如果出现了错误,则会在Output窗口显示错误的类型、错误所在行以及错误的原因。此时,双击错误信息或按【F4】键,则会在编辑窗口的左侧出现一个箭头指示发现错误的语句行,以便修改源程序。应该说C++的编译器虽然可以查出错误,但是对错误的说明可能并不十分准确,而且一个实际错误往往会引出若干条错误说明,因此在检查错误时,应首先查看第一个错误出现的位置,在改正了该错误后,可以先进行编译,此时往往会发现错误的数目已经大大减少。重复此过程直到所有的错误均已修改,然后再连接、运行程序。

在连接阶段也可能出现一些错误提示,与编译错误提示信息不同的是连接错误不指出错误发生的详细位置,这是因为连接的对象是目标程序,与源程序格式有很大差别,不容易确定错误的准确位置。

连接阶段出现的错误一般比较少,大多数是因为在程序中调用了某个函数,而连接程序却找不到该函数的定义。初学者经常发生的连接错误是程序中没有主函数或有两个主函数,前者是因为把主函数名main拼错了,如写成了mian;后者是因为编制下一个新程序时没有新建项目,而只新建了源程序文件,使得一个项目中有了两个主函数。在找到连接错误的原因并修改以后,一定要重新编译后才能再次进行连接。

【例1.3】编写程序,计算并输出边长为2和3的长方形的面积。

源程序:

```
//计算矩形的面积
#include <iostream>
using namespace std;
int main()
{
    int a, b, s;
    a = 2;
    b = 3;
    s = a * b;
    cout<<"area = "<<s<<endl;
    return 0;
}
```

步骤:

① 参照例1.1,创建C++项目,选择Win32 Console Application,项目名称为area,保存位置为E:\Examples。如果还打开着上一个程序,在“New”对话框中默认选项卡是“Files”,一定选择“Projects”选项卡,为新程序创建一个新项目。

② 创建C++源程序,命名为rec.cpp。

③ 输入源程序代码,为说明如何查看和修改编译和连接问题,按如下代码进行输入:

```
#include <iostream>
using namespace td;
int mian()
{
    int a, b, s
    a = 2;
    b = 3;
    s = a * b;
    cout<<"area = "<<s<<endl;
    return 0;
}
```

④ 编译。单击工具栏上的“Compile”按钮，编译源程序文件。从底部的输出窗口可看到有5条编译错误。双击输出窗口中的第一条错误，或按【F4】键，在源程序编辑窗口中出现一个箭头，指向该错误对应的代码行，如图1.14所示。该错误信息为：‘td’：does not exist or is not a namespace，因此需将该行中的td修改为std，而后再单击“Compile”按钮，显示只有一条错误，再次双击此错误信息，箭头指向a = 2;那一行，如图1.15所示。错误信息missing ‘;’ before identifier ‘a’，表示在标识符a之前缺少分号，因此需在上一行的int a, b, s后添加分号。修改完毕后再单击“Compile”按钮，编译成功，生成目标文件rec.obj。



图1.14 第一次编译后的结果



图1.15 第二次编译后的结果

⑤ 连接。单击工具栏上的“Build”按钮或使用【F7】键进行连接，此时在Output窗口中显示有两条连接错误，如图1.16所示。错误信息显示unresolved external symbol _main，检查源程序中的main函数的定义发现误将main写成了mian，由于C++程序中必须有且只能有一个主函数main()，在连接过程中未找到main()函数，从而造成了连接错误。修改完成后再次进行连接，如图1.17所示，连接成功，生成了area.exe文件。



图1.16 第一次连接后的结果



图1.17 第二次连接后的结果

⑥ 运行。单击工具栏上的“Execute Program”按钮或使用【Ctrl+F5】组合键，运行程序，程序运行结果如图1.18所示。

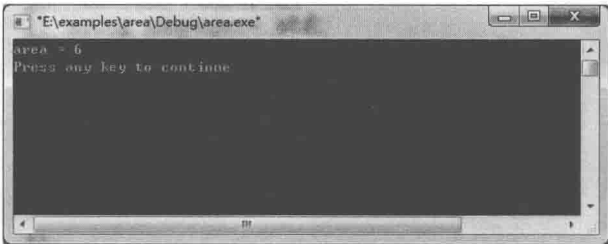


图1.18 程序运行结果

说明:

在完成一个程序的编译、连接和运行后,若需要再创建一个新程序,此时需要重新创建项目、创建源文件,不能在同一个项目中添加两个包含main()主函数的源文件,否则会出现连接错误。

如果源程序文件没有经过编译,而直接单击工具栏上的“Build”按钮进行连接,系统会自动进行编译,编译无误后才能连接。同样的,如果没有编译、连接而运行程序,系统也要依次进行编译、连接后才能运行程序。

4. 查看和修改运行错误

有些程序在编译、连接时都没有错误,能够运行,但是得到的运行结果却和预期的结果不同,例如,计算结果错误、死循环(进入循环后无法终止循环)、对只读的内存空间进行写操作等,这类错误就是运行错误,运行错误也称为逻辑错误。通常导致运行错误的原因有如下几种:

- ① 程序书写错误,例如,在循环结构中修改循环控制变量的表达式应该写作i++,却写作了i--。
- ② 程序的算法有误,对各种情况考虑不周。
- ③ 数组元素的下标越界,对未分配的内存空间进行写数据的操作等。

运行时的错误通常比较隐蔽,不容易发现,需要检查程序的算法、逻辑以及执行过程是否正确等。此时,可使用调试工具跟踪程序的执行,以便找出程序中的真正错误。

常用的调试手段为:

(1) 设置断点(Breakpoint)。

所谓断点,实际上就是告诉调试器在何处暂时中断程序运行,以便查看程序的状态以及浏览和修改变量的值等。在程序中设置和清除断点可使用如下方式:

- ① 按快捷键【F9】。
- ② 单击Build工具栏中的按钮。
- ③ 在需要设置(或清除)断点的位置上右击,在弹出的快捷菜单中选择“Insert/Remove Breakpoint”命令。

利用上述方式可以将断点设置在程序源代码中指定的一行上,或者某函数的开始处,或指定的内存地址上。一旦断点设置成功,则断点所在代码行的最前面的窗口页边距上将显示一个红色的实心圆。

(2) 单步跟踪。

当程序在断点处暂停时,就进入了单步跟踪状态。通过单步跟踪,可以逐条语句或逐个函数地执行程序,每执行完一条语句或一个函数,程序就暂停,因此可逐条语句或逐个函数地检查它们的执行结果。

对程序的单步跟踪执行有以下几个选择:

- ① Step Into (【F11】): 执行一条语句,如果此语句中有函数调用,则进入该函数内部,在该函数的第一行代码处暂停。
- ② Step Over (【F10】): 执行一条语句,如果此语句中有函数调用,则把整个函数视为单步一次执行。
- ③ Step Out (【Shift+F11】): 继续执行程序,当遇到断点或返回函数调用者时暂停。如果当前执行位置是在某一函数内,可使程序快速跳出该函数。
- ④ Run to Cursor (【Ctrl+F10】): 将光标定位在某行代码上并使用这个命令,程序会执行到断点或光标定位的那行代码暂停。
- ⑤ Stop Debugging (【Shift+F5】): 程序运行终止,回到编辑状态。

(3) Watch窗口。

当程序暂停时,除了可以控制它的执行,还可以通过监视窗口来查看和修改各个变量的值。为了更好地进行程序调试,调试器提供一系列的窗口,用来显示各种不同的调试信息。借助View菜单下的Debug Windows子菜单可以访问它们。事实上,当用户启动调试器后,Visual C++ 6.0的开发环境会自动显示出

Watch和Variables调试窗口，且Output窗口自动切换到Debug页面。综合使用监视窗口和单步跟踪功能，可找出程序中隐藏的逻辑错误。

以上是程序调试的一些基本方法。当然，Visual C++ 6.0功能强大的调试器还能调试断点、异常、线程、OLE以及远程调试等，且支持多平台和平台间的开发，有兴趣的读者可进一步查阅相关资料。

5. 熟悉Visual C++ 6.0集成环境其他常用操作

(1) 程序的打开与关闭。

要打开一个程序，只要打开其项目的工作区文件（扩展名为.dsw）即可。选择File→Open Workspace菜单命令，打开“Open Workspace”对话框，选择驱动器、文件夹和项目工作区文件，单击“打开”按钮打开程序；或者，从File→Recent Workspaces级联菜单中选择最近操作过的工作区文件；还可以从Windows资源管理器窗口找到存放项目的文件夹，双击其中的工作区文件（扩展名为.dsw，例test.dsw），可打开程序。

注意：

一个程序对应一个项目，要打开一个程序，就是要打开它对应的项目的工作区文件（.dsw），并不是打开它的源程序文件（.cpp），请初学者一定要注意。

使用“File”→“Save Workspace”菜单命令，可保存当前打开的项目。

选择“File”→“Close Workspace”菜单命令，可将当前打开的项目关闭。

(2) 工作区窗口（Workspace）。

工作区窗口一般位于Visual C++ 6.0主窗口的左侧，如果没有显示，可在菜单栏或工具栏区的空白处右击，从快捷菜单中选择“Workspace”命令将其打开，也可以选择“View”菜单下的“Workspace”命令。使用同样的方法可控制输出窗口Output的显示。

打开例1.1的工作区文件test.dsw，在工作区窗口中选择“FileView”选项卡，单击顶层“test files”前的“+”或双击其前的图标，将其展开；再单击“Source Files”前的“+”，显示出项目中已经创建的源程序文件。对项目的很多操作都可在工作区窗口中进行，例如：

① 打开文件。双击test1.cpp源程序文件，可在右边的文本编辑窗口将其打开。

② 删除文件。单击test1.cpp文件选中它，选择“Edit”→“Delete”菜单命令或直接按【Del】键，将该文件从项目中删除，但文件依然存放在硬盘上。

③ 将已有的文件添加到项目中。选择“Project”→“Add To Project”→“Files…”菜单命令，或者在工作区窗口中的“Source Files”文件夹上右击，从快捷菜单中选择“Add Files to Folder”命令；在弹出的“Insert Files into Project”对话框中，选择test1.cpp文件，单击“OK”按钮，将文件添加到项目中。

④ 在工作区窗口右击源程序文件test1.cpp，从快捷菜单中选择“Compile test.cpp”命令进行编译。

(3) 熟悉Build MiniBar工具栏。

在操作中，使用工具栏比菜单更为方便。在编制程序过程中，经常要进行程序的编译、连接和运行，可使用Build MiniBar工具栏中的按钮来操作，还可以使用更快捷的热键，如图1.19所示。



图1.19 Build MiniBar工具栏

如果Visual C++ 6.0中没有显示Build MiniBar工具栏，可在工具栏区右击，从快捷菜单中选择“Build MiniBar”命令将其显示出来。



三、实验内容

1. 单选题

(1) 在一个C/C++程序中, _____。

A. main()函数出现在所有函数之前

B. main()函数可以在任何地方出现

C. main()函数出现在所有函数之后

D. main()函数出现在固定位置

(2) 以下叙述中正确的是: _____。

A. 一个控制台类型的项目只能有一个源文件

B. 一个控制台类型的项目只能有一个头文件

C. 一个控制台类型的项目只能有一个main()函数

D. 一个控制台类型的项目只能有一个函数

(3) 一个C++程序的执行是从_____。

A. main()函数开始, 直到main()函数结束

B. 第一个函数开始, 直到最后一个函数结束

C. 第一个语句开始, 直到最后一个语句结束

D. main()函数开始, 直到最后一个函数结束

(4) 若在Visual C++中打开一个项目, 只需打开对应的项目工作区文件, 项目工作区文件的扩展名为_____。

A. .cpp

B. .exe

C. .obj

D. .dsw

(5) 以下关于C语言和C++关系的描述中, 正确的是_____。

A. C语言是C++的子集

B. C++对C语言进行了改进

C. C++与C语言都是面向对象的语言

D. C++继承了C语言的众多优点

2. 编程题

(1) 修改例1.3, 使其能计算并输出长方形的周长和面积。

(2) 编写程序, 输出三行文字"Big data" "Internet+" "Deep learning"。

(3) 创建程序, 按如下代码输入, 进行编译、连接和运行, 分别观察显示的编译错误、连接错误和运行错误, 并进行修改。

```
#include <iostream>
using namespace std;
int mian()
{
    int a, b, c;
    c = a + b;
    a = 1;
    b = 2;
    cout>>c>>end;

    return 0;
}
```

四、问题讨论

(1) 一个C++程序中可否包含两个main()函数?

(2) 编写完成一个程序后, 如何创建下一个程序?

(3) 创建程序, 按如下程序代码输入, 而后编译、连接、运行。

```
#include <iostream>
using namespace std;
int main()
```

```
{  
    int a, b, c;  
    cout<<"Enter a = ";  
    cin>>a;                //输入a  
    cout<<"Enter b = ";  
    cin>>b;                //输入b  
    c = a * b;              //计算a和b的乘积存入c  
    cout<<"a * b = "<<c<<endl;  
    //输出结果  
    return 0;  
}
```

试思考，如何修改程序，当输入a为12，b为11时，使得运行结果为：

12 * 11 = 132

当输入a为36，b为18时，使得运行结果为：

36 * 18 = 648