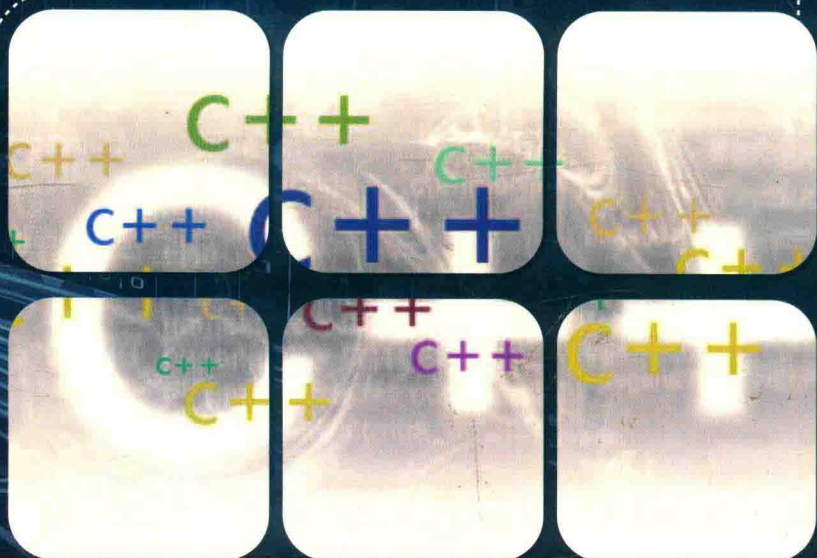


C++ 语言程序设计 习题与实验指导

姚雅鹃 石礼娟 主编



科学出版社

C++语言程序设计习题 与实验指导

姚雅鹃 石礼娟 主编

科学出版社

北京

内 容 简 介

本书是与《C++语言程序设计》(姚娟、汪毅主编,科学出版社出版)相配套的实验教材。全书包含三部分,第一部分是集成开发环境介绍,给出了基于 Visual C++ 2010 的 C++编写程序的步骤和程序调试的方法,以帮助读者掌握编程环境的使用方法;第二部分是实验内容,结合教材各章节的理论知识,精心设计了大量趣味性、应用性的题目,作为读者上机编程实践操作的训练题;第三部分是习题训练,紧密结合理论教材各章节的知识点,在形式和内容上贴近全国计算机等级考试,提供了大量的习题,且题型多样,以帮助读者巩固所学的程序设计知识。

本书既可以作为普通高等院校 C++程序设计课程的辅助教材,也可以作为各类工程技术人员进行 C++编程练习和上机训练的指导用书。

图书在版编目(CIP)数据

C++语言程序设计习题与实验指导/姚雅鹃,石礼娟主编. —北京:科学出版社, 2018

ISBN 978-7-03-058189-

I. ①C… II. ①姚… ②石… III. ①C语言-程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 140775 号

责任编辑:戴薇 王国策 王会明 / 责任校对:陶丽荣

责任印制:吕春珉 / 封面设计:东方人华

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>

三河市骏杰印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2018 年 8 月第 一 版 开本:787×1092 1/16

2018 年 8 月第一次印刷 印张:12 1/4

字数:280 000

定价:34.00 元

(如有印装质量问题,我社负责调换〈骏杰〉)

销售部电话 010-62136230 编辑部电话 010-62135397-2008

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303



任何一门程序设计课程，不上机实践是不可能学好的。学习程序设计时，很多读者会觉得看书或听课时，程序语言的各种语法要素和编程方法的叙述都非常简单，老师讲的内容基本上也能理解，但需要自己针对具体问题编程时往往又无从下手，要么编写的程序漏洞百出，要么无法利用已学知识来正确表达自己的逻辑思维，因此感到程序设计课程太难。应该说，这是学习编程过程中的一种正常现象，程序设计是一门实践性很强的课程，要把自己所学的知识变成编程解决问题的能力，所需要的就是上机实践。

上机实践是学习程序设计的重要教学环节，只有通过上机实践，才能真正理解程序设计的基本思想，正确、灵活地使用语言中的各种要素，熟练掌握各种集成开发环境的应用和编写、调试程序的方法，从而获得应用程序设计解决实际问题的经验和技巧。

本书是与教材《C++语言程序设计》配套的辅导与实验教材，旨在为读者在学习 C++ 语言程序设计的上机实践和知识巩固的过程中提供训练和帮助。

本书包括三部分，第一部分是集成开发环境介绍，给出了基于 Visual C++ 2010 的 C++ 编写程序的步骤和程序调试方法，使读者在了解编程环境的基础上，明确使用 C++ 语言进行程序设计的步骤和过程，掌握编程环境的应用，学习程序调试的方法和技巧。

第二部分是实验内容，结合教材各章节的理论知识，这一部分设置了 12 个实验，每个实验的内容围绕核心编程技能，设计了应用性较强同时也不乏趣味的题目，作为读者上机编程实践操作的训练内容。实验内容包括程序填空和程序设计，对问题有清楚的输入、输出描述及样例，帮助读者更好地自测程序设计是否正确。实验内容设置难易均衡，读者可有选择地选取难度适合的问题做上机实践。

第三部分是习题训练，帮助读者巩固所学的程序设计知识，紧密结合理论教材各章节的知识点，对应教材的章节顺序，提供了 9 章习题训练，习题数量大，且题型多样，包括单选题、填空题、判断题、程序阅读题、程序填空题等，并在后面给出了习题的参考答案。

本书所给出的每个题目都在 Microsoft Visual C++ 2010 环境下通过调试和运行。需要指出的是，无论是习题解答还是程序编写，解题的方法都不是唯一的。本书中给出的程序不一定是最优的，希望能对读者有所启发，欢迎读者提出自己的思路和想法，编写出更高质量的程序。

本书的编写人员全部是多年从事一线教学的教师，具有丰富的实践教学经验。本书由姚雅鹃、石礼娟担任主编，章程、陈芳、田芳担任副主编。具体编写分工如下：第一部分、实验十二及习题一、八、九由姚雅鹃编写，实验一、二及习题二由章程编写，实验三、四、五及习题三、四由陈芳编写，实验六、七、八及习题五、六由石礼娟编写，

实验九、十、十一及习题七由田芳编写。全书的统稿工作由姚雅鹃负责完成。本书的编写得到了科学出版社的大力支持和帮助，许多长期致力于程序设计实践教学的教师也对本书提出了宝贵的意见和建议，在此表示衷心的感谢。

感谢读者选择本书，由于编者水平有限，书中难免存在不妥之处，敬请广大专家、读者不吝指教，对本书的内容提出批评和修改建议。

编者

2018年5月



第一部分 集成开发环境介绍

| | |
|--------------------------------|---|
| Visual C++ 2010 集成开发环境介绍 | 3 |
|--------------------------------|---|

第二部分 实验内容

| | |
|------------------------|----|
| 实验一 顺序与选择结构 | 13 |
| 实验二 循环结构 | 18 |
| 实验三 数组 | 24 |
| 实验四 函数（一） | 29 |
| 实验五 函数（二） | 35 |
| 实验六 指针（一） | 40 |
| 实验七 指针（二） | 46 |
| 实验八 结构化程序设计 | 52 |
| 实验九 类和对象 | 54 |
| 实验十 友元和运算符重载 | 61 |
| 实验十一 类的继承、派生与多态性 | 67 |
| 实验十二 输入/输出流 | 74 |

第三部分 习题训练

| | |
|----------------------|----|
| 习题一 基本数据类型和表达式 | 81 |
| 习题二 控制结构 | 84 |

| | |
|---------------------|-----|
| 习题三 数组 | 105 |
| 习题四 函数 | 115 |
| 习题五 指针 | 124 |
| 习题六 结构化程序设计 | 141 |
| 习题七 面向对象程序设计 | 146 |
| 习题八 输入/输出流 | 161 |
| 习题九 简单数据结构及算法 | 169 |
| 习题参考答案 | 174 |
| 参考文献 | 187 |



第一部分

集成开发环境介绍



Visual C++ 2010 集成开发环境介绍

集成开发环境 (integrated development environment, IDE) 是指集代码编写、分析、编译、调试等功能于一体的软件开发工具。目前有许多软件产品可以帮助软件工程师完成 C++ 程序的开发。例如, 微软公司的 Visual C++; Borland 公司的 C++ Builder、开放源码软件 Code::Blocks 等; 在 Linux 平台下, 还有 GNU 的 GCC 和 GDB 等。

Visual C++ 实际上是微软公司开发的工具集 Microsoft Visual Studio 中的一个重要组成部分。微软公司早在 1998 年推出了 Visual Studio 6.0, 随着新标准的推出和软件技术的发展, 微软公司后来又不断地推出了多个 Visual Studio 版本。目前, 常用的版本是 Visual Studio 2010。Visual Studio 2010 产品发布了 Express、Professional、Premium、Ultimate、Test Professional 五个不同的版本, 以满足不同程度的专业需求。其中, Express 版为学习版, 可以免费下载, 供初级软件开发者学习。本章将介绍 Visual C++ 2010 集成开发环境的使用, 由此也让读者了解在 Visual C++ 中使用标准 C++ 语言编写程序的过程。

一、C++ 程序的开发过程

C++ 是一种高级程序设计语言, 它的语法规则更接近人类自然语言和数学表达的习惯。而计算机能够识别的只能是机器语言, 因此要让计算机“读懂”一个 C++ 程序, 需要经过转换才能识别。高级语言转换为计算机可识别的机器语言的方式主要有两种: 一种是解释执行方式, 它类似于英语翻译成汉语时采用的同声翻译, 应用程序源代码一边由相应语言的解释器翻译成目标代码 (机器语言), 一边执行, 因此效率比较低, 而且不能生成可独立执行的可执行文件, 应用程序不能脱离其解释器, 但这种方式比较灵活, 可以动态地调整、修改应用程序; 另一种是编译方式, 编译指在应用源程序执行之前, 就将程序源代码翻译成目标代码, 它类似于英语翻译成汉语时所采用的笔译方法, 因此其目标程序可以脱离其语言环境独立执行, 使用比较方便、效率较高, 但修改应用程序不方便。

下面简要说明使用 C++ 语言设计应用程序的过程。

1) 编写程序。在文本编辑器中用 C++ 语言编写源代码文件, 以扩展名 .cpp 保存源代码程序。

2) 编译程序。编译器将 C++ 程序源代码翻译成目标代码。如果是在 .NET 平台上执行的程序, 编译器则将程序编译成中间语言代码。实际上, 在编译之前还要完成一项称为“预处理”的工作, 它的目的是根据源代码中的预处理指令在源代码中相应地插入和替换字符文本, 程序中的注释也在预处理过程中被删除, 因此注释中的内容除了方便阅读代码之外, 不会对程序的执行产生任何影响。

3) 连接程序。程序通常包含对标准类库或其他类库所定义的函数和数据的调用, 连接器将被调用的相关代码组合到可执行文件中。最后生成的可执行文件的扩展名

为.exe，这是一个在操作系统中或.NET 框架上可执行的程序。

4) 执行程序。由操作系统加载可执行文件，将其先读入到计算机内存中，最后 CPU 根据程序中的指令完成各种操作。

5) 调试程序。程序在编译、连接和执行阶段都可能出现错误，程序员需要用系统提供的调试工具帮助发现并指出错误及原因，修改源程序中的错误。

二、Visual C++ 2010 集成开发环境

Visual C++ 2010 支持快速应用开发 (rapid application development, RAD)。RAD 工具可以帮助程序员直观、快速地创建高质量的和丰富用户体验的应用程序。Visual C++ 2010 学习版的启动画面如图 1-1 所示。其集成开发环境的主窗口包括菜单栏、工具栏、“解决方案资源管理器”窗格、程序设计窗格、“工具箱”窗格、“属性”窗格等部分，如图 1-2 所示。



图 1-1 Visual C++ 2010 学习版启动画面

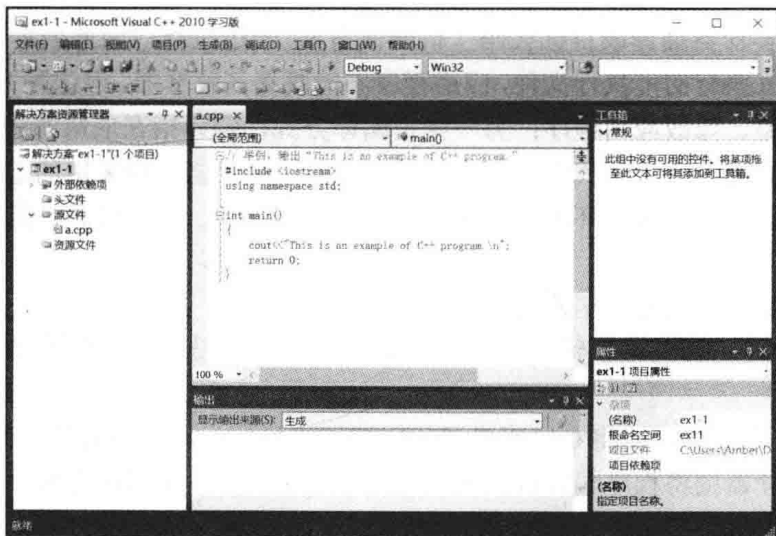


图 1-2 Visual C++ 2010 学习版的集成开发环境主窗口

菜单栏由多个菜单项组成，菜单中包含用于管理 VC++ IDE 及开发、维护和执行程序过程中的所有命令。

工具栏中包含常用的命令按钮，如新建项目、保存文件、执行程序等。将鼠标指针停留在按钮上几秒后，会显示按钮的功能描述。右击工具栏，会弹出快捷菜单，通过其中的命令能设置工具栏中的项目。

“解决方案资源管理器”窗格提供了设计方案中的所有文件的便捷访问，双击其中的文件项将打开相应的文档。

程序设计窗格位于整个主窗口的中央，在此窗格中进行程序的源代码编写。此区域在进行窗体应用程序的设计时，可以提供一种近似所见即所得的设计视图形式来显示用户控件、HTML 页和内容页，可以对文本和元素进行添加、定位、调整大小，以及使用特殊菜单或“属性”窗格设置属性的操作。

“工具箱”窗格用于分类存放各种控件，可将控件拖曳到设计视图上，实现可视化界面设计。

“属性”窗格可以显示设计视图中当前所选中控件、代码文件的属性。

三、项目和解决方案

Visual C++ 2010 是不能单独编译一个.cpp 文件的，这些文件必须依赖于某一个项目，因此在 Visual C++ 2010 中开发设计一个 C++ 程序必须创建一个项目。对于不同类型的项目，Visual C++ 2010 均会进行编译器设置，并生成起始代码。一般情况下，建立首个项目的同时会默认建立一个解决方案，一个解决方案可以加入多个项目。为解决方案生成起始代码时，会将解决方案下的所有项目全部生成。

下面通过创建一个简单程序说明在 Visual C++ 2010 中使用 C++ 语言设计程序的主要步骤。

设计步骤如下：

1) 从菜单栏中选择“文件”→“新建”→“项目”命令，或单击工具栏中的“新建项目”按钮，弹出“新建项目”对话框，选择“Win32 控制台应用程序”选项，如图 1-3 所示。



图 1-3 “新建项目”对话框

2) 在“新建项目”对话框下方的“名称”文本框中输入项目名称“ex1-1”，当创建项目时，Visual C++ 2010 会将该项目放入一个解决方案。默认情况下，解决方案的名称与项目名称相同；在“位置”文本框中指定项目保存的目录位置，可以使用默认位置；或输入一个另外的目录位置；还可以单击“浏览”按钮，在弹出的对话框中选择项目存放的目录。最后，单击“确定”按钮，弹出“Win32 应用程序向导”对话框，在此对话框中，单击“下一步”按钮，出现“应用程序设置”界面。在此界面中，选中“空项目”复选框，如图 1-4 所示，然后单击“完成”按钮，回到主窗口。

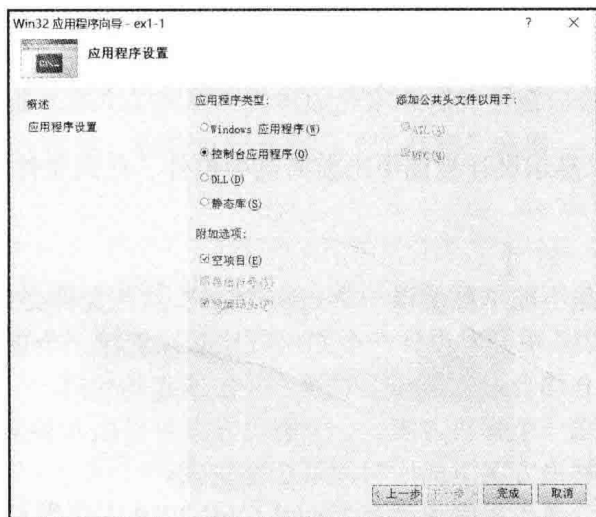


图 1-4 “应用程序设置”界面

现在有了项目，但项目还没有源代码文件，下面为 ex1-1 项目添加源代码文件。

3) 在“解决方案资源管理器”窗格中，右击“源文件”文件夹，在弹出的快捷菜单中选择“添加”→“新建项”命令，如图 1-5 所示，将弹出“添加新项”对话框，在此对话框中选择“C++文件 (.cpp)”选项，这里输入名称“test”，如图 1-6 所示，最后单击“添加”按钮，出现 test.cpp 的编辑窗口。



图 1-5 添加源文件



图 1-6 “添加新项”对话框

4) 在 test.cpp 的编辑窗口中, 输入以下代码(行号是为了方便后文解释, 不用输入)。

```
1 // ex1-1, 输出"This is an example of C++ program."字符串
2 #include<iostream>
3 using namespace std;
4 int main()
5 {
6     cout<<"This is an example of C++ program."<<endl;
7     return 0;
8 }
```

5) 从菜单栏中选择“生成”→“生成 ex1-1”命令, 如果在下方的“输出”窗格中显示生成“成功”, 表示项目已编译, 并且未发生错误, 再按下组合键 Ctrl+F5, 或者从菜单栏中选择“调试”→“开始执行”命令, 输出如图 1-7 所示的执行结果。

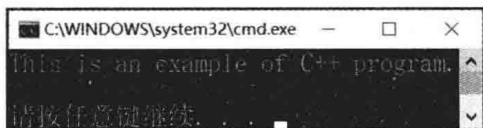


图 1-7 ex1-1 程序的执行结果

下面通过对 ex1-1 的分析, 初步认识 C++ 程序的基本结构。

第 1 行是注释语句, 用于说明整个程序。C++ 程序有两种注释方法: 一种是用“//”, 其后的内容(在同一行)全是注释; 另一种是用“/*”开头, 用“*/”结束, 中间的内容全是注释, 这是从 C 语言传承的注释方法。

注释虽然不对程序的执行产生任何作用, 但它可以给程序增添可读性, 为程序添加注释是一种良好的编程风格。注释还可用于程序调试, 在编写程序时, 有些语句还不能确定是否需要删除, 可以先加注释符使其暂时不产生作用, 这样可避免语句删除之后发现其有用又要重写的麻烦。

第2行是编译预处理指令,它告诉编译器将用于输入/输出流的头文件 `iostream` 包含在该文件中,该文件提供了与输入/输出相关的声明,否则编译器将无法识别 `cout` 对象。

第3行是使用命名空间 `std`。C++标准库中的类和函数是在命名空间 `std` 中声明的,第2行的 `#include` 指令告诉编译器程序要用到标准的输入/输出流,而本行的“`using namespace std;`”语句表示从命名空间 `std` 中导入代码。

第4行的 `main` 是主函数名。

第5行的左花括号和第8行的右花括号分别表示函数体的开头与结尾。在设计 C++ 应用软件的项目中,有且仅有一个 `main` 函数。该函数是执行程序入口,程序都是从中开始执行的。

第6行是输出语句。计算机调用标准库的功能输出字符串“`This is an example of C++ program.`”。

第7行是返回语句。该语句返回 `0`,告诉系统程序正常结束,否则表示程序有异常。`main` 函数前面的 `int` 与该语句相呼应,表示函数需要返回一个整数。

四、调试程序的工具

程序设计是一项复杂的脑力劳动,编程过程中出现语法错误和逻辑错误是十分常见的现象,排除程序中的错误是程序员最常做的工作,也是程序设计的基本功。

Visual C++ 2010 提供了一些用于在生成应用程序时对其进行测试和调试的工具。借助于调试工具,能让程序在某个位置暂停执行,进而观察到程序的内部结构和内存的状况,帮助程序员找到错误产生的原因。初学者应尽快学会调试工具的使用,这有助于其理解 C++ 语言中的概念和计算机程序的执行机理,提高其编程的能力。

Visual C++ 2010 的程序调试器功能强大,可以中断(或挂起)程序的执行以检查代码;计算和编辑程序中的变量,查看寄存器,查看从源代码创建的指令,以及查看应用程序所占用的内存空间。使用编程工具的“编辑并继续”功能,可以在调试时对代码进行更改,然后继续执行。下面列出 Visual C++ 2010 编程环境中调试程序的主要方法。

1) 设置与取消断点。将光标定位在程序中的某一行,右击,在弹出的快捷菜单中选择“断点”→“插入断点”命令,或者按 `F9` 键,在光标所在行的左侧出现红色圆点,如图 1-8 所示。关闭断点的方法是单击红色圆点或再次按 `F9` 键。系统允许在程序的多个位置设置断点。

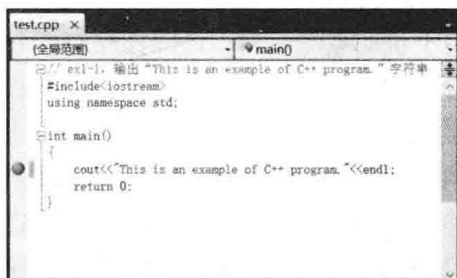


图 1-8 设置断点

2) 启动与停止调试。设置了断点后,在菜单栏中选择“调试”→“启动调试”命令,或者单击工具栏中的“启动调试”按钮,程序开始执行并在断点处停止;在菜单栏中选择“调试”→“停止调试”命令,或者单击工具栏中的“停止调试”按钮,程序从调试状态退出。

3) 程序跟踪执行。进入调试状态后,通过在菜单栏中选择“调试”→“逐语句”命令或按 F11 键,也可在菜单栏中选择“调试”→“逐过程”或按 F10 键,使程序进入一次执行一行代码的“单步执行”状态。“逐语句”和“逐过程”这两个命令都指示调试器执行下一行的代码,区别仅在于它们处理函数调用的方式不同:如果某一行包含函数调用,“逐语句”仅执行函数调用本身,然后在函数体内的第一行代码处停止;而“逐过程”则执行整个函数,然后在函数调用的下一条执行语句处停止。

如果程序调试位于函数调用的内部,可使用“跳出”功能立刻返回调用函数的位置,在工具栏中单击“跳出”按钮或按组合键 Shift+F11 快速使用跳出功能。

4) 观察程序内部状态。程序进入调试状态后,可以通过“自动窗口”“局部变量”“监视”“调用堆栈”等窗口来查看程序执行到当前语句时内存中的变量、寄存器等状态。

