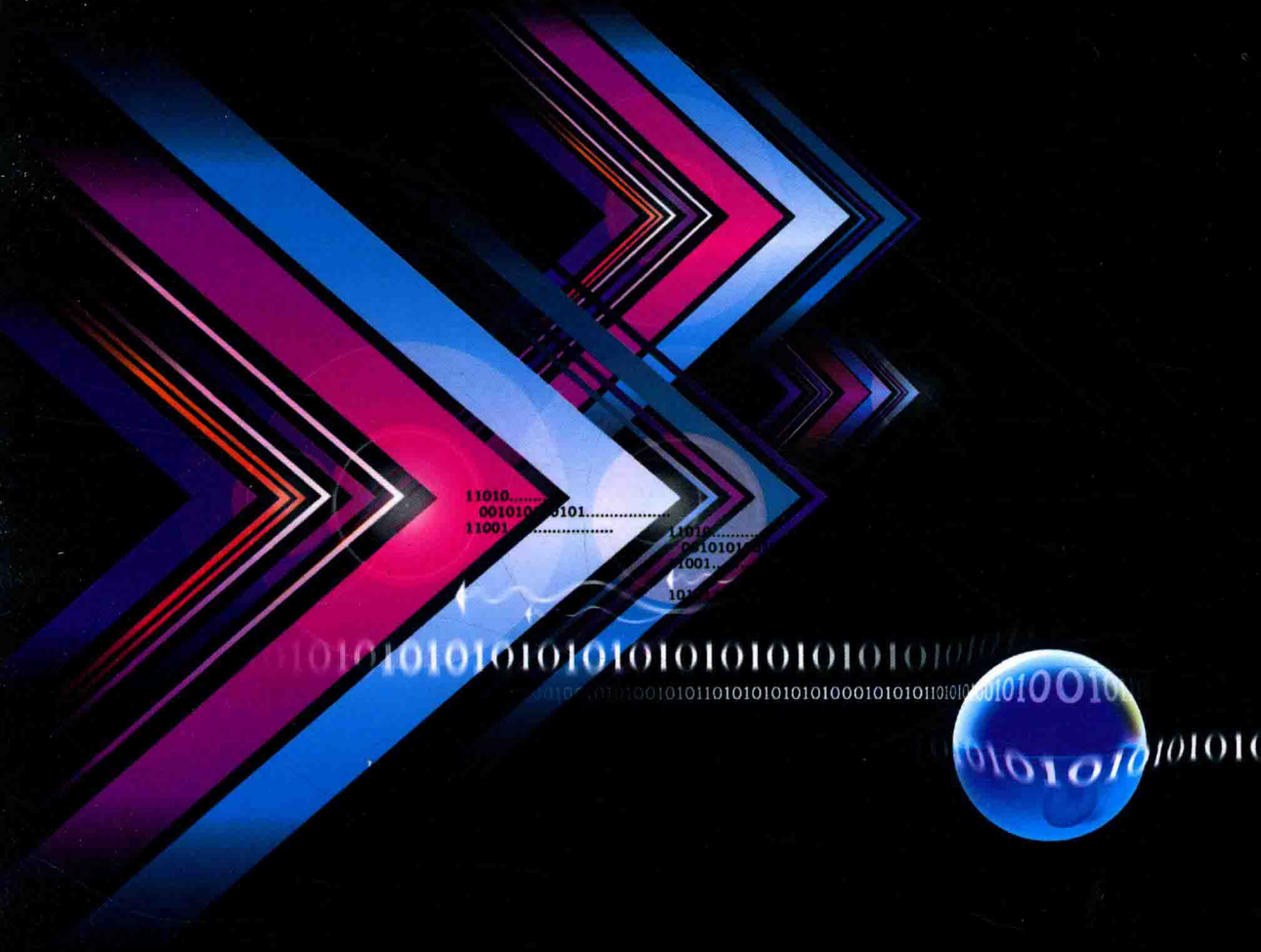
中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材



C++语言程序设计案例与实践辅导

刘瑞芳　肖波　许桂平　孙勇　徐惠民　编著

西安电子科技大学出版社
http://www.xduph.com

中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

C++语言程序设计案例与实践辅导

刘瑞芳　肖 波　许桂平　孙 勇　徐惠民　编著

西安电子科技大学出版社

内 容 简 介

本书是《C++语言程序设计》的学习辅导书。全书共 14 章，第 1 章介绍在 VC2015 环境下编程的步骤和各种平台上的 C++程序编译方法，第 2 章至第 11 章与教材《C++语言程序设计》对应，包括教材各章的习题及答案、编程案例及参考例程和实践题目。第 12 章至第 14 章作为课程设计的内容，讲解了窗口程序设计的方法、Visual Studio 环境下开发网络通信的案例和 QT Creator 环境下跨平台开发信息处理系统的案例。

本书为读者学习 C++语言程序设计提供了丰富的内容，适合作为大学各专业的 C++程序设计课程的辅导书和 C++课程设计的教材，也可供编程爱好者自学使用。

图书在版编目(CIP)数据

C++语言程序设计案例与实践辅导/刘瑞芳等编著. —西安：西安电子科技大学出版社，2017.1
ISBN 978-7-5606-4395-3

Ⅰ. ① C… Ⅱ. ① 刘… Ⅲ. ① C 语言—程序设计 Ⅳ. ① TP312.8

中国版本图书馆 CIP 数据核字(2016)第 323925 号

策　　划　毛红兵
责任编辑　万晶晶
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西华沐印刷科技有限责任公司
版　　次　2017 年 1 月第 1 版　　2017 年 1 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印　张　23.25
字　　数　551 千字
印　　数　1～3000 册
定　　价　46.00 元

ISBN 978-7-5606-4395-3/TP

XDUP　4687001-1

中国电子教育学会高教分会

教材建设指导委员会名单

前　言

C++ 语言是一门优秀的语言，全面兼容 C 语言，并在保留了 C 语言简洁、灵活、高效的同时，增加了面向对象程序设计的支持，从诞生以来一直受到广大编程人员的喜爱。

本书是《C++ 语言程序设计》的配套教材。本书编写的主要目的，是希望帮助学生提高使用 C++ 语言进行程序设计的能力，因此本书也是一本 C++ 编程指导和参考书。

本书内容主要包括以下 6 个方面：(1) C++ 编程环境介绍；(2) 习题答案；(3) 编程案例及参考例程；(4) 实践题目；(5) 窗口程序设计；(6) 课程设计案例。

全书分为 14 章，第 1 章介绍 VC2015 编程环境和各种平台上的 C++ 程序编译方法。第 2 章至第 11 章的目录和《C++ 语言程序设计》的第 2 章到第 11 章相对应。每一章都包括习题及答案、编程案例及参考例程和实践题目 3 个部分。

“习题及答案”部分不仅提供习题解答，还对习题的难度等级进行划分，对部分难题进行了剖析。

“编程案例及参考例程”是本书的重点之一。本书按照 C++ 语言的各个知识点设计了各种编程案例，把编程的思想融入例子中，目的是使读者能够对现实世界中较典型的问题用计算机语言进行描述及解决。这部分不仅有大量的编程案例，还提供了对于案例的分析、设计、参考例程以及对于例程的解释说明，力求使读者在掌握 C++ 语言的同时，能够掌握编程的思路。

“实践题目”每章只有一题，但是各章的题目有连贯性，由浅入深，直到最后可以作为课程设计题目。相应的程序和解析可以在第 14 章第 14.1 节中找到答案。

第 12 章至第 14 章作为课程设计的内容，讲解了窗口程序设计的方法，分析、设计并实现了两个案例。第 13 章是网络通信方面的案例，选用 Visual Studio 集成环境进行开发；第 14 章是信息处理方面的案例，选用 QT Creator 集成环境进行开发，支持跨平台开发、移植。这部分的内容相当有代表性，可以作为老师进行课程设计的教学参考或学生的自学参考。

本书可以作为学习 C++ 程序设计的参考书，更可供编程爱好者自学使用。书中如有不妥之处，欢迎广大读者批评指正。

编　者

2016 年 10 月

目　录

第 1 章
C++语言概述

1.1 《C++ 语言程序设计》习题及答案

1.1 在 VC 集成开发环境中，要产生一个可执行的 EXE 文件的步骤是什么？

答案：

步骤

(1) 建立一个工程；

(2) 新建或者导入源文件，然后编辑源文件；

(3) 编译源文件，产生目标文件；

(4) 将目标文件和其他库文件链接产生可执行的 EXE 文件。

难度：2

1.2 C 语言与 C++ 语言的关系是什么？

答案：C++ 包含了整个 C，C 是建立 C++ 的基础。C++ 包括 C 的全部特征、属性和优点，同时添加了对面向对象编程(OOP)的完全支持。

难度：1

1.3 结构化程序设计与面向对象程序设计有什么异同点？

答案：结构化程序设计的主要思想是功能分解并逐步求精。面向对象程序设计的本质是把数据和处理数据的过程当成一个整体对象。

难度：2

1.4 面向对象程序设计的基本特征是什么？

答案：封装性、继承性、多态性。

难度：1

1.5 为了编辑和运行 C++ 程序，在 VC 环境下已经建立了一个工程 Proj01，也建立了一个 C++ 文件 file01.cpp。现在有一个 C++ 程序 input.cpp，希望调入到这个工程中编译和运行，应该如何操作？

答案：方法有多种：

(1) 用 input.cpp 的内容替换 file01.cpp 内容，再编译和运行；

(2) 选择“项目”→“添加现有项”菜单项，将 input.cpp 添加到工程中，然后将

原来的 fie01.cpp 从工程中移除，再编译和运行。

难度：3

1.6 C++ 是否可以输出中文字符串？仿照例 1-1 编写程序，实现在屏幕上显示“北京欢迎你”。

答案：C++ 可以输出中文字符串。

仿照例 1-1 编写程序如下：

```
#include <iostream>
using namespace std;
void main()
{
    cout<<"北京欢迎你"<<endl;
}
```

难度：2

1.2 VC2015 集成开发环境简介

较早期程序设计的各个阶段都要用不同的软件来进行处理，如先用字处理软件编辑源程序，再用编译程序进行编译，然后用链接程序进行函数、模块链接，开发者必须在几种软件间来回切换操作。为了使开发者便利、高效地开发各种应用程序，现在的编程开发软件集编辑、编译、调试等功能于一身，这样就大大方便了用户。目前大多数开发语言都有相应的集成开发环境(Integration Development Enviroment，IDE)，IDE 是一种辅助程序开发人员开发软件的应用软件，一般包括代码编辑器、编译器、调试器和图形用户界面工具，集成了代码编写功能、分析功能、编译功能、调试功能等，这样的软件可以独立运行，也可以和其他程序并用。有些 IDE 还支持多种编程语言，但一般而言，IDE 主要还是针对特定的编程语言量身打造的。

Visual Studio 是一套完整的开发工具集，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C#和 Visual J# 等全都使用相同的集成开发环境，利用此 IDE 可以共享工具且有助于创建混合语言解决方案。这里仅介绍 Visual C++(VC2015)的使用，用于开发 C++ 程序。

VC2015 集成开发环境由如下若干元素组成：菜单栏，工具栏，停靠或自动隐藏在左侧、右侧、底部以及编辑器空间的各种工具窗口。在任何给定时间，可用的工具窗口、菜单和工具栏取决于所处理的项目或文件类型。启动界面如图 1-1 所示，窗口大致被划分为 2 部分，右侧是项目工作区窗口，左侧是编辑窗口。

可以从起始页开始工作，其中，“最近”这一栏表示最近几次使用 VC2015 打开的 C++ 项目目录，鼠标单击其中之一即进入到该开发项目中，或者单击“开始”这一栏下面的“新建项目”建立一个新项目，也可以从菜单栏中选择“文件”→“新建”或“文件”→“打开”开始工作。

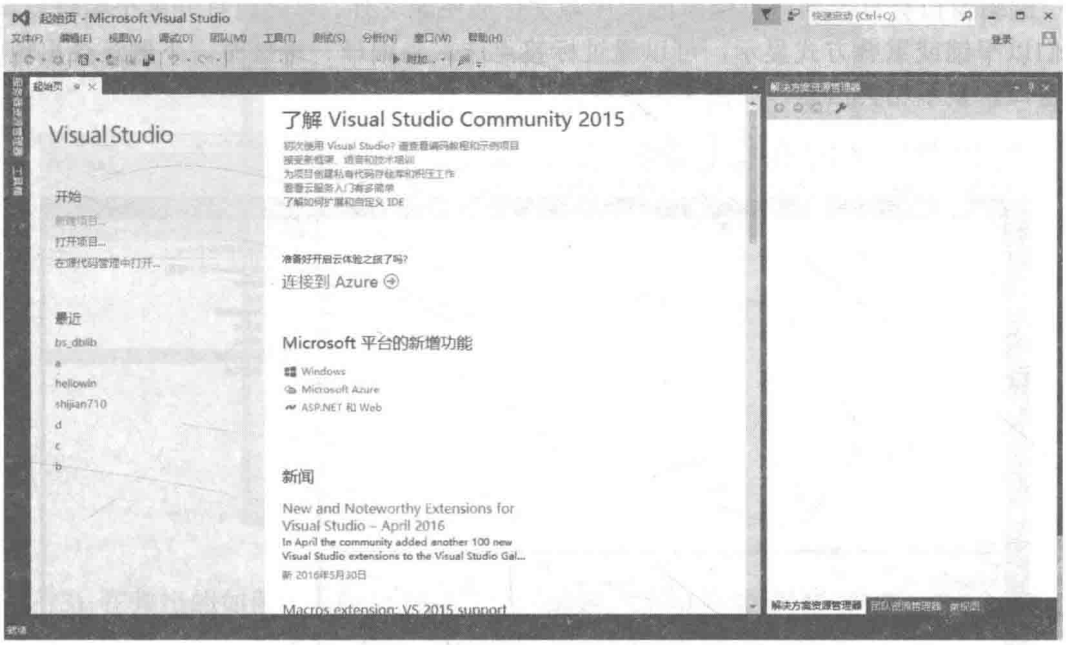


图 1-1　VC2015 开始界面

右侧的项目工作区窗口一般由解决方案资源管理器、团队资源管理器、类视图 3 个标签组成，如图 1-2 所示。一个解决方案中可以包含几个项目，一个项目是一个完整的程序，可以编译、链接，生成可执行程序，且一个项目最多只能有一个 main 函数。所以在任一时刻，解决方案中的几个项目只能有一个是激活的。单击类视图标签可以看到各项目中的所有类，以树形结构显示，然后单击相应的类进行编辑。

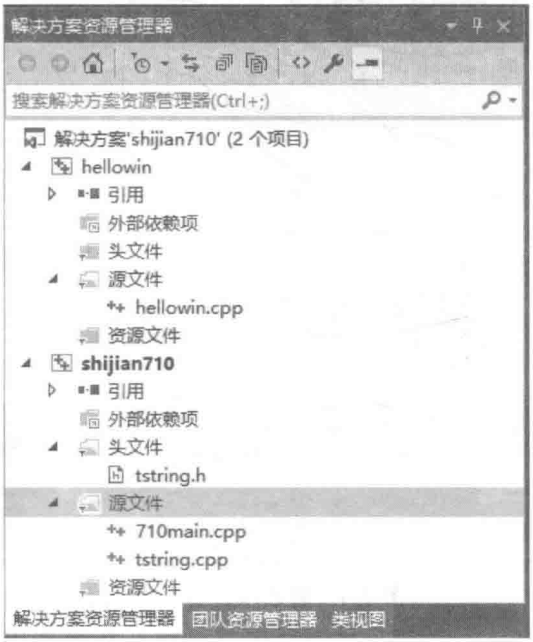


图 1-2　项目工作区

编辑窗口会显示当前可编辑的程序源文件或资源文件，可同时打开多个编辑窗口，它们以平铺或重叠方式显示，可以通过标签点选。当编译、链接时，下侧还会出现输出窗口，显示相关信息，如图 1-3 所示。

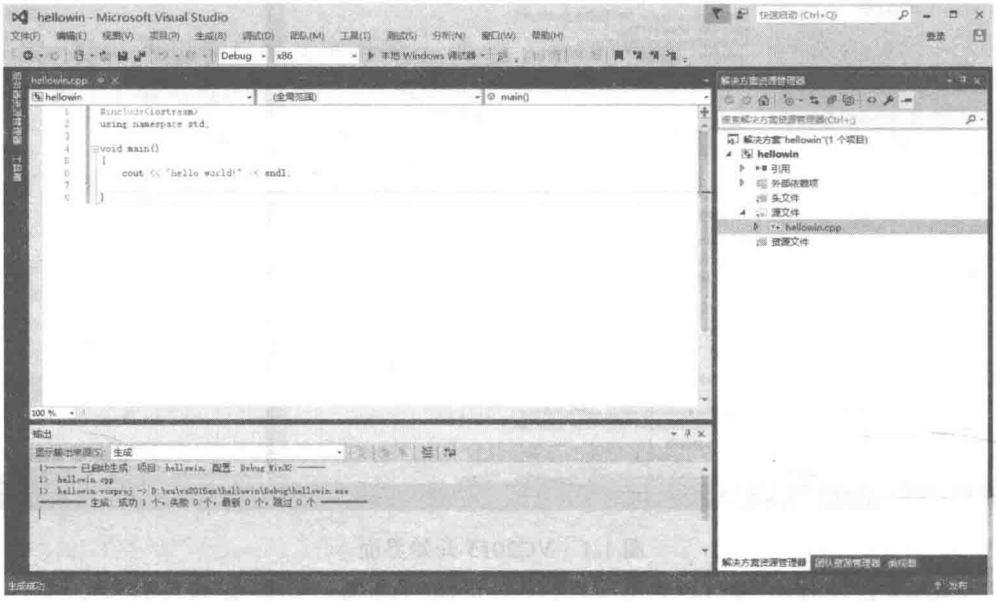


图 1-3　编辑窗口

按照下面步骤建立一个 Win32 控制台应用程序。

(1) 在窗口中选择“文件→新建→项目”，会弹出如图 1-4 所示的窗口。

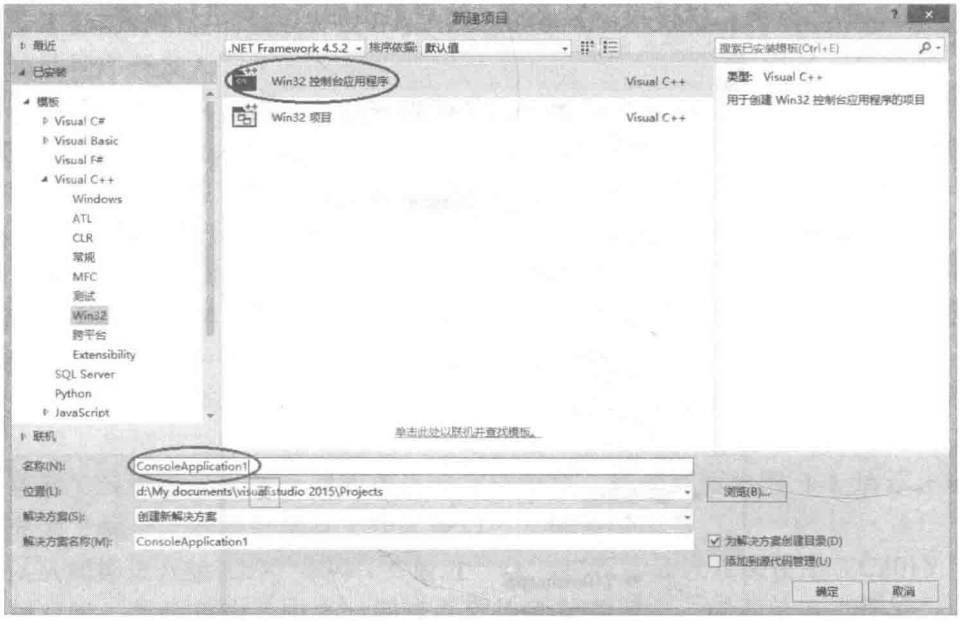


图 1-4　新建项目第一步

(2) 输入工程的名称，不必加上后缀。按“确定”按钮后出现如图 1-5 所示窗口，然后按“下一步”按钮。

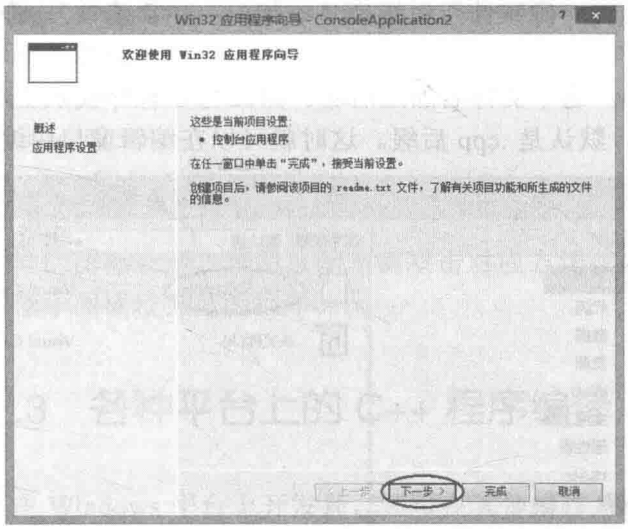


图 1-5　新建项目第二步

(3) 在弹出的如图 1-6 所示的窗口中，选择“空项目”，最后按“完成”按钮。

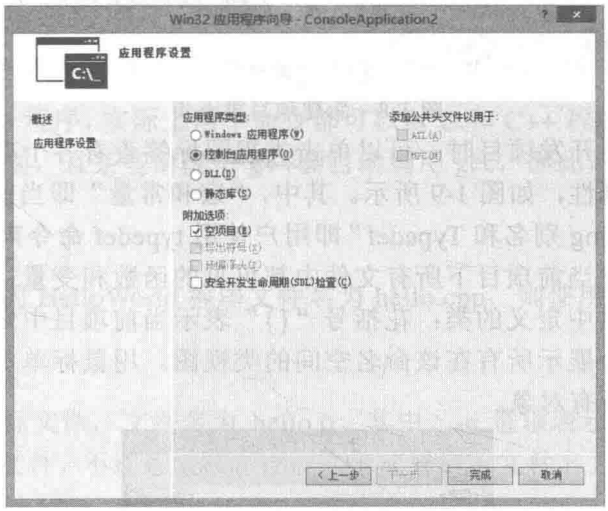


图 1-6　新建项目第三步

(4) 这时候在“解决方案”上看到“源文件”，如图 1-7 所示。

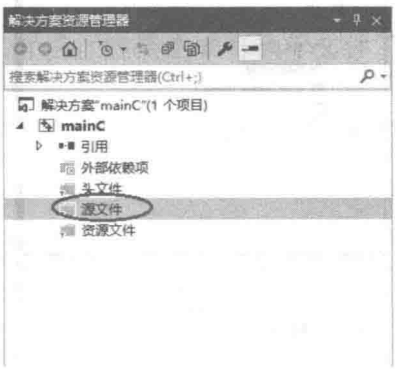


图 1-7　新建项目第四步

(5) 鼠标右键单击“源文件”后选择“添加”→“新建项”，弹出一个对话框，如图 1-8 所示。

(6) 输入文件名，这个文件名加上后缀。如果写 C 程序就加上 .c 后缀，写 C++ 程序后缀加不加都行，默认是 .cpp 后缀。这时就可以在编辑窗口中编写源文件了。

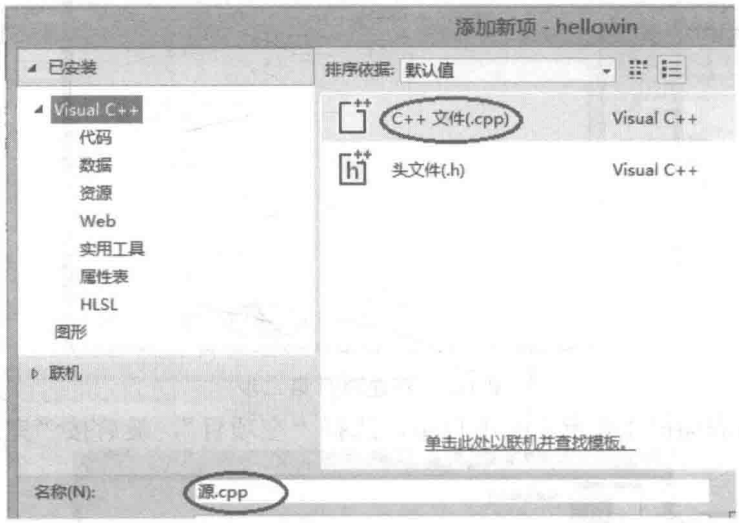


图 1-8　新建项目第五步

在编辑含有类的开发项目时，可以单击类视图标签查看各个项目下的类结构，即其数据属性与方法属性，如图 1-9 所示。其中，“宏和常量”即当前项目中定义的所有宏与常量；“全局 Using 别名和 Typedef”即用户使用 typedef 命令声明的一些别名；“全局函数和变量”即在当前项目下所有文件中都可见的函数和变量；一些三色方框对应的名称即在当前项目中定义的类；花括号“{}”表示当前项目中定义的命名空间，打开其下拉内容后，会展示所有在该命名空间的类视图。用鼠标单击其中任意内容，即可查看该类别下的所有对象。

图 1-9　类视图窗口

例如，图 1-9 中选中了类“SiteTrade”，在其下方的方框中，分别展示了该类中声明的所有成员函数与数据属性，对应如下：

```
~SiteTrade(void)
load_file_to_data(sodict_buil)
SiteTrade(u_int size_of_idx_)
mintf_filt_dup_dict
```

函数与数据由不同的图标表示。使用鼠标左键双击对应方法或属性，则跳到源代码中对应的行，非常便于阅读较大型的程序。

1.3　各种平台上的 C++ 程序编译

C/C++ 程序除了在 Windows 平台上开发外，还可在其他操作系统平台上开发。本节主要介绍在 Linux 操作系统上编译、链接 C/C++ 程序的方法。

1.3.1　Linux 操作系统的编译与链接

Linux 操作系统 C/C++ 程序常用的编译链接命令为 gcc 和 g++。通常认为 gcc 编译 C 程序，g++ 编译 C++ 程序，实际上两个命令都可以对 C 和 C++ 程序进行编译和链接，只是命令选项有些区别，且某些情况下 g++会自动调用 gcc，因此，本节主要介绍 g++ 的基本用法。

1. 编译命令

假定我们编写好的 HelloWorld 程序文件名为 hello.cpp，则使用 g++ 进行编译的方法如下：

```
g++ -c hello.cpp
```

编译器会产生目标文件，文件名为 hello.o。其中，-c 选项表示只进行预处理、编译等工作，产生目标文件，不进行链接。若指定特定的目标文件名为 h.o，则命令如下：

```
g++ -c hello.cpp -o h.o
```

其中，-o 选项是指定生成的目标文件名称。若没给出文件名，则默认的名称为源文件.o。

2. 链接命令

编译之后便可以对目标文件进行链接生成可执行文件了，命令如下：

```
g++ hello.o
```

链接器会产生可执行文件，文件名默认为 a.out，若指定特定的目标文件名为 hello，则命令如下：

```
g++ hello.o -o hello
```

其中，-o 选项是指定生成的可执行文件名称。

3. 编译与链接

若编译和链接合在一起执行，则可执行命令如下：

```
g++ hello.cpp -o hello
```

编译器会自动对源文件进行编译并链接，生成可执行文件 hello。

若程序有多个源文件构成，如 file1.cpp，file2.cpp，则可执行命令如下：

```
g++ file1.cpp file2.cpp -o program
```

除了前面介绍的 -c、-o 外，gcc/g++ 都提供了非常丰富的功能选项，完成编译和链接等过程的不同需求。这些选项的使用，在此不再赘述，读者可使用命令“g++ --help”了解各个选项的作用。

4. makefile 文件与 make 命令

Linux 下很多软件包都提供了安装源代码，可通过 make 命令进行自动编译和链接。尤其是当软件的源代码文件比较多时，make 命令可以自动确定哪些部分需要重新编译，哪些部分可以不需要再次编译，直接使用以前编译生成的文件，可以大大减少编译所需的时间。

make 命令需要 makefile 文件的支持。makefile 文件通常会和程序源代码一起发布，它描述了程序各个文件之间的关系，提供了更新每个文件的命令。例如，源文件更新了，相应的目标文件就需要更新。目标文件更新了，相应的可执行文件就需要更新。每当源文件更新时，就可以直接运行 make 命令，使得所有需要更新的文件全部完成更新，从而得到最终的可执行程序。

下面是一个简单的 makefile 文件的例子。

```
hello:   hello.o
        g++ hello.o -o hello
hello.o: hello.cpp
        g++ -c hello.cpp -o hello.o
clean:
rm –f hello hello.o
```

makefile 文件的结构比较简单，通常包含若干目标，每个目标由对应与之相关联的其他文件名以及实现这个目标的一组命令构成。如上例所示，共定义了三个目标，分别为 hello、hello.o 和 clean。每个目标从每行开头开始写，后面跟冒号(:)。冒号后为与该目标相关的其他目标或文件，用空格隔开。如上例所示，hello 目标关联的文件为目标文件 hello.o，hello.o 目标关联的文件为源文件 hello.cpp。针对某目标的命令另起一行，若该目标有多个命令，则这些命令也可以由若干行构成。需要注意的是，每行命令必须使用 tab 键缩进，而不能使用空格。

一般情况下，使用 make 命令进行编译的方法为

```
make target
```

target 是 makefile 文件中定义的目标之一。如果省略 target，则将 makefile 文件中的第一个目标作为 target 进行处理。

如上例所示，将 makefile 文件和 hello.cpp 文件放在同一目录下，并运行：

```
make  或 make hello
```

程序会自动进行编译和链接，生成目标文件 hello.o 和可执行文件 hello。运行：

```
make clean
```

会根据 clean 对应的命令自动删除生成的文件。整个过程如图 1-10 所示。

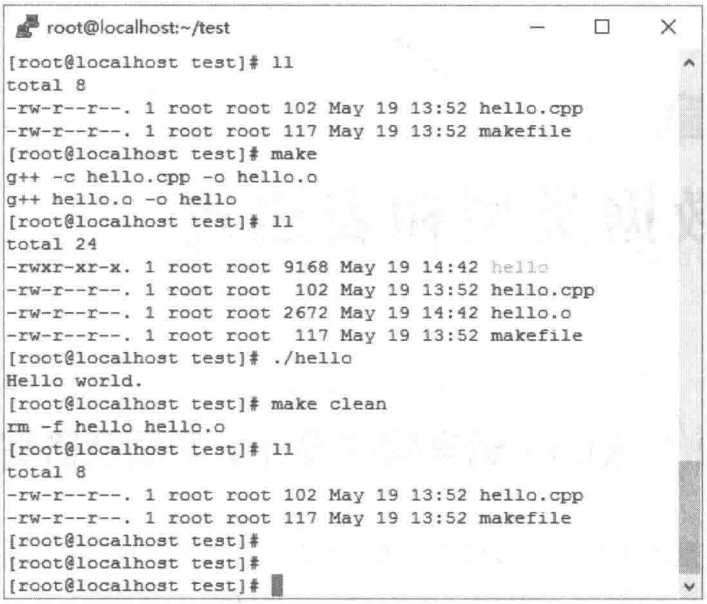


图 1-10　make 命令执行过程

make 命令可以使用 -f 选项指定某个要操作的 makefile 文件。若不指定，会在当前目录下按顺序寻找 GNUmakefile、makefile 和 Makefile 文件。

关于 makefile 文件更高级的使用在此不再赘述，请读者参考相关资料。

1.3.2　其他编译与链接工具

大多数操作系统都有很多 C/C++ 命令行编译链接工具或 IDE 工具，例如 Mac OS X 操作系统上可以使用 Xcode IDE 进行开发，Windows 下也可以使用 DEV-C++、MinGW 等 IDE 进行开发。

此外，很多 C/C++ 命令行编译链接工具或 IDE 工具是支持跨平台的。例如，QtCreator 可以在 Windows、Linux、Mac OS X 等操作系统上运行；gcc/g++、make 等命令也可以安装在 Windows 或 Mac OS X 上，使用命令行模式运行。