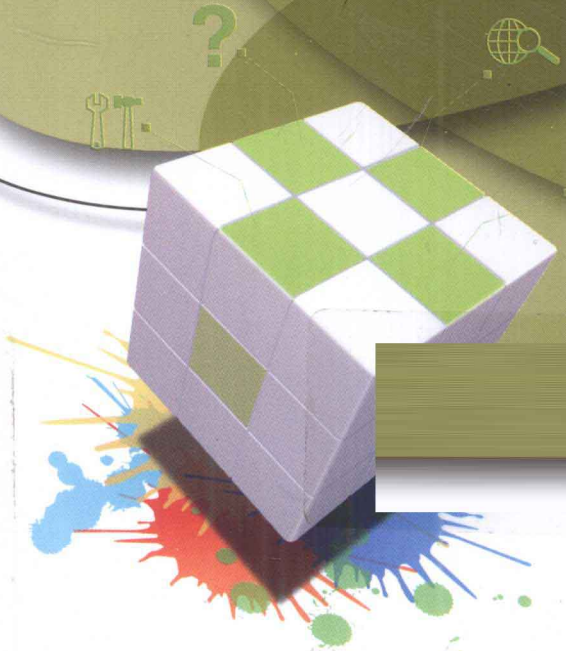
全国高校应用人才培养规划教材·网络技术系列

MIANXIANG DUIXIANG SHEJI YU YINGYONG

面向对象设计与应用

丁任霜 主编



Network
Tech

北京大学出版社
PEKING UNIVERSITY PRESS

全国高校应用人才培养规划教材·网络技术系列

面向对象设计与应用

主　编　丁任霜
副主编　巩晓秋　於肇鹏
　　李　新　王建亮

北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

面向对象作为软件设计指导思想，为计算机软件设计提供了一个全新的视角。C ++ 是一种高效实用的程序设计语言，全面支持面向对象技术。掌握 C ++ 的精髓和实质，对深刻理解面向对象技术大有裨益。本教材是编者总结多年教学科研与工程实践经验写成的，适合用作大学计算机专业和非计算机专业的程序设计课程教材，也可供自学的读者使用。

本教材共分为 3 个部分。第 1 部分包括第 1 章至第 3 章，主要讲解 C ++ 面向对象的封装与基础等基础知识；第 2 部分包括第 4 章至第 6 章，主要讲解 C ++ 面向对象的一些高级特性，如多态、运算符重载、流和异常处理等内容；第 3 部分包括第 7 章至第 11 章，主要讲解使用 Visual C ++ 开发 Win32 平台软件的基础知识和通用组件模型的概念与应用。

图书在版编目（CIP）数据

面向对象设计与应用/丁任霜主编. —北京：北京大学出版社，2011.5
（全国高校应用人才培养规划教材·网络技术系列）
ISBN 978-7-301-18349-6

Ⅰ. ①面… Ⅱ. ①丁… Ⅲ. ①面向对象语言，UML-程序设计-高等学校-教材 Ⅳ. ①TP312
中国版本图书馆 CIP 数据核字（2010）第 260410 号

书　　名：面向对象设计与应用
著作责任者：丁任霜　主编
策 划 编 辑：吴坤娟
责 任 编 辑：吴坤娟
标 准 书 号：ISBN 978-7-301-18349-6/TP·1144
出 版 者：北京大学出版社
地　　址：北京市海淀区成府路 205 号　100871
网　　址：http: //www. pup. cn
电　　话：邮购部 62752015　发行部 62750672　编辑部 62756923　出版部 62754962
电 子 信 箱：zyjy@pup. cn
印 刷 者：河北滦县鑫华书刊印刷厂
发 行 者：北京大学出版社
经 销 者：新华书店
787 毫米×1092 毫米　16 开本　20 印张　391 千字
2011 年 5 月第 1 版　2011 年 5 月第 1 次印刷
定　　价：38.00 元

前　言

面向对象程序设计不同于传统的面向过程程序设计，是一种新的程序设计思路和实现方法，针对指定的场景，使用面向对象的原则指导软件的设计与实践，可以大幅度地改善软件的通用性与可维护性，提高软件的生产效率和质量水平。

C++语言在保留C语言灵活高效的基础上，大量引入面向对象的设计思想和相应的语法结构。封装、继承和多态是C++面向对象程序设计的核心思想。进一步完善的“类型说明与定义”、新的“输入/输出流”、“引用”以及丰富强大的“标准库”的加入，为软件开发人员提供了强大、高效的支持。

Visual C++是微软公司开发的基于Win32平台C++开发工具，Visual C++包含一个比较著名的软件开发框架（Framework）：微软基础类库（Microsoft Fundation Classes，简称MFC）。MFC利用C++面向对象的特性，对Win32上面的数目众多的API加以重组、抽象和封装，在一定程度上降低了开发基于Win32平台软件的劳动强度。

本教材重点介绍了C++面向对象程序设计语言的主要概念与设计方法，Win32平台的软件开发基础以及如何使用Visual C++和MFC编写基于Win32平台的软件等内容。全书共由11章组成，根据知识结构可以分为3个部分：第1部分由第1章至第3章组成，这部分主要讲解C++面向对象的封装与基础等基础知识；第2部分由第4章至第6章组成，这部分主要讲解C++面向对象的一些高级特性，如多态、运算符重载、流和异常处理等内容；第3部分由第7章至第11章组成，这部分主要讲解使用Visual C++开发Win32平台软件的基础知识和通用组件模型的概念与应用。

本教材的几位编者都长期工作在教学第一线，教学经验比较丰富，对学生在学习过程中可能遇到的困难有比较深刻的理解；同时具有比较强的工程实践能力，许多见解和观点来自于对实际工程设计实施的亲身经历。本教材第1章由王建亮老师编写，第4、5、6和第11章由丁任霜老师编写，第2、3章由於肇鹏老师编写，第7、8章由巩晓秋老师编写，第9、10章由李新老师编写，最后由丁任霜老师统稿。感谢本教材所列参考文献的作者！感谢为本教材出版付出辛勤劳动的北京大学出版社的工作人员！

由于编者的水平有限，编写过程中难免有不足和错误的地方，诚恳接受读者朋友的批评指正。如对本教材有任何意见和见解，请与编者联系，Email：abc618382@126.com。

编　者

2011年2月

目　录

第 1 章　面向对象基础

1.1　问题领域的抽象

抽象是指在认识活动中运用概念、判断、推理等思维形式，对客观现实进行间接的、概括的反映的过程。其属于理性认识阶段。科学的抽象是在概念中反映客观现实的内在本质的思想，它是在对事物的本质属性进行分析、综合、比较的基础上，抽取出事物的本质属性，撇开其非本质属性，使认识从感性的具体进入抽象的规定，进而形成概念。

人们使用计算机解决现实环境中的问题，就是要将现实世界的问题转换成为计算机能够运行的程序，通过获取计算机的运行结果，寻求解决现实环境中问题的答案。往往现实环境的问题纷繁复杂，而使用抽象方法是解决复杂问题的一种基本方法。抽象就是对复杂问题的相关性质的局部观察，抽取对当前系统相对重要的关键部分，忽略其他次要部分。

在计算机程序设计领域中，抽象的解决方法概况起来基本上有两类：一类是面向过程的解决方案，另一类是面向对象的解决方案。

1.1.1　面向过程的抽象

面向过程的抽象是以功能为中心，将现实世界的复杂问题抽象成为功能和子功能。以功能抽象为基础的结构化程序设计，围绕着实现“面向过程”的功能来设计和构造整个系统。其基本指导思想是采用自顶向下、分而治之、逐步求精的原则，将需要解决的问题按照功能分解为若干个子问题，然后就这些子问题再依照同样的指导原则进一步分解，直到分解为一些容易控制而且功能相对独立的子问题为止，将这些子问题对应为程序的功能模块。整个系统构建在这些功能模块之上，因为现实问题的复杂性，所映射系统的功能模块之间通常存在着调用与被调用的关系。

使用面向过程的设计方式，人们把计算机程序看做是处理数据的一系列过程，这一系列的过程最终在计算机程序中体现为一个一个的模块，每个模块最终是由若干条指令构成。在这个基本原则的指导下，结构化的程序设计方法应运而生。功能分解和逐步求精是基本指导原则。面对一个比较复杂的实际任务时，可以将其拆分为一系列规模较小的功能部件集合，然后对拆分后的功能部件集合再进行同样的递归操作，继续分解，直到所得的功能部件规模较小，而且易于理解与编程为止。至于拆分到何种程度，软件工程领域给出了许多具体的标准和规范。但从另一方面来说，这种拆分活动亦可称之为软件系统的规划，需要实施人员具有该问题领域丰富的实践经验，在一定程度上具有较大的灵活性和不确定性。这些具有丰富实践经验和理论知识的实施者通常被称做系统架构师。

可以通过计算一个学校指定学期的平均课时量的例子来理解这个过程。这是一个比较复杂的数据处理过程。通过分析可以将其拆分成如下的子任务：

- 输入指定学期
- 找到一个教师
- 计算学校共有多少教师
- 计算学校指定学期的课时量之和
- 教师人数除课时量之和
- 输出计算结果

其中，计算课时量之和可以继续拆分成如下的子任务：

- 读取指定教师的身份编号
- 读取课时数据
- 累加课时数据
- 继续读取下一个教师的数据

其中，读取课时数据可以继续拆分成如下的子任务：

- 根据教师身份编号和指定学期读取课程编号
- 根据课程编号读取课时数据

面向过程的抽象机制有效地建立起现实问题与计算机之间的联系，对计算机软件技术的发展起到了重要的推动作用。随着计算机的普及和发展，非数值领域的处理需求日益增加，面对的问题领域规模不断的扩大，功能抽象的解决方案也逐渐暴露出了一些不足。

当数据量不断增加的时候，数据和处理数据的程序之间的分离导致程序变得越来越难以理解和维护。另外，外界需求的持续变化，同样会导致系统结构的不稳定。由于系统是采用自顶向下的策略分析设计的，一旦上层甚至顶层的逻辑结构发生变化，通常带来的结果就是，处于变化分层下面所有的子模块都需要重新设计和编写。这使得系统在应对需求变化的过程中往往处于被动的状态。

1.1.2 面向对象的抽象

面向对象的核心思想就是把数据和处理数据的程序视作为一个整体：对象。面向对象的抽象就是用对象的观点描述客观世界。将现实世界的问题抽象为离散的、相互关联的对象集合。每一个对象都可以视作为一个自治的单元，都具有静态的属性和动态的行为。在对现实问题领域进行建模的过程中，面向对象的设计思路不再是以功能为中心，而是首先将问题领域进行主体的划分，划分出可标识的若干个主体单元，然后将现实问题领域的关系，映射为对象之间的联系，使得这些主体单元能够构成一个有机的整体，使其更加接近于现实问题领域。这些被划分出来的，具有可标识特性的主体单元，就称之为对象。

面向对象的技术中，对象是封装了数据和操作的单元。数据是属于对象单元自己的，操作则是这个对象单元的对外接口。其他对象只能通过指定接口发送消息，不能访问内部的私有数据。比如我们要求某人“起立”（发消息），某人接受消息后调整自己的肌肉、神经使自己起立（操作），从而改变了他的状态（描述姿态的数据变了）。外界不必关心触动了内部的几条神经、收缩了几块肌肉。因此，内部的数据和实现操作的算法若有改

动，对其他程序对象没有任何影响。通过这样的方法，可以提高整个系统的扩展性和稳定性。

1.1.3 面向过程与面向对象的关系

这两种抽象方法其目的都是为现实问题领域建立计算机程序模型，然后通过编制、运行程序，获取运算结果，找到现实问题领域的解决方案。

从诞生的时间方面来看，面向过程的建模解决方案要早于面向对象，同时面向对象的抽象设计在很大程度上是由于使用面向过程的抽象设计在映射某些客观问题领域的时候遇到了困难，为了消除困难和提高效率而出现的。但是从另外一个角度来看，面向对象中有许多技术是直接来自于面向过程的，因此具有扎实面向过程的理论与实践知识，对于更好地理解和掌握面向对象技术是有着非常积极的意义。

面向过程以功能为中心，擅长于解决需求相对稳定、变化频率较低的现实问题领域，例如计算机的操作系统（如 Windows 系列和 Linux 系列等）；计算机网络智能设备（如交换机、路由器等内部的软件系统）和个人移动嵌入设备（如手机、掌上游戏机和多媒体娱乐设备等）。这些领域，采用以功能为中心、面向过程的设计解决方案，符合其实际特点，可以获得比较好的实施效果。

面向对象则在应对快速变化的需求方面具有优势。系统架构师针对实际的现实问题领域，从宏观的角度出发，找出问题领域中相对稳定的部分和经常变化的部分，对稳定的部分给出实际的解决方案，通常称之为“框架”或 FrameWork。例如：在 Win32 平台上面开发图形用户界面（GUI）常用的“微软基础类库”MFC（Microsoft Foundation Classes）；现由诺基亚开发维护的跨平台 C ++ 图形用户界面应用程序框架 Qt，提供具有跨平台能力的图形用户界面的应用程序框架以及 Apache 软件基金会 Struts 框架，用于建立基于 MVC 模式的 Web 应用。程序开发者根据实际需要解决的问题，选择适当的框架，在框架的帮助下编写面向对象的程序代码，降低劳动强度、提高生产效率和代码质量，从而获得比较好的实施效果。

因此，在选择项目的总体解决方案时，采用面向过程还是面向对象，需要根据项目的具体情况，从实际出发综合作出决定，符合客观实际的设计才是良好的设计，千万不要跟风赶时髦，为了面向对象而面向对象，导致不必要的、甚至是过度的设计。

1.2 面向对象概述

面向对象的概念起源自 20 世纪 70 年代初的计算机语言 SmallTalk，由 Alan Kay、Dan Ingalls、Ted Kaehler 和 Adele Goldberg 等在 Xerox PARC 开发。其设计和实现思想对面向对象程序设计产生了非常深远的影响。通过 SmallTalk，人们开始逐渐了解面向对象的程序设计，后来随着 C ++ 的广泛使用，使得面向对象的程序设计迅速的普及开来。经过不断的发展和创新，面向对象已经不仅仅局限于编程阶段，逐渐演变为从分析、设计、编程和测试等一系列完整的软件开发方法。

面向对象的方法认为客观世界是由各种对象所组成，任何事物都可以视作为对象，每个对象都有自己的运行规律和内部状态。从类比的角度看，某些对象同属于一个类别，而

另外某些对象属于另外一个类别，对象与类别之间可以视作为 n 对 1 的关系。复杂的对象可以由简单的对象以某种形式组合构成，不同对象的有机组合构成了人们要研究和构造的系统。通过类比的方法找出对象间的相似特性（即对象所具有的共同属性）进而形成类。将类按照父类、子类的概念构成层次结构。对于同属于一个类的对象，可以定义一组方法来说明此对象的功能，即可以施加在该对象上面的各种操作。默认情况下，子类可以自然地继承来自父类的属性和方法。

1.2.1 封装和数据隐藏

封装就是将属性和操作整合在对象中并与外界隔离开。对象的封装在现实生活中随处可见。例如：当技术人员维修升级电脑的时候，需要增加内存容量，不需要使用原始的集成电路芯片和电路板自己去制作一块内存卡，需要做的是到电脑市场购买指定容量和类型的内存条。技术人员不关心内存条内部的工作原理，内存条本身是自成一体的，这种特性就称为封装。无须知道封装单元内部是如何工作就可以使用，这被称之为数据隐藏。

面向对象的程序设计中，对象被视作为一种自治、封装的实体。通过定义对象属性和操作的可见性，决定哪些对外是可见的、即为公有的，哪些是隐藏在对象内部，对外是不可见的、即为私有的。对象之间通过接口进行消息传递，接口可视作是对象对外可见的操作集合。

C++ 通过建立用户自定义类型支持封装和数据隐藏。完备定义的类一旦被建立，即可视作是完全封装的实体，可以作为一个整体单元使用。类的实际内部工作对外应该隐藏起来，使用的时候只要知道如何使用即可，不需要关心内部的运行细节。

封装和数据隐藏可以为程序设计带来的好处主要有如下两个方面：一是可以有效地控制变化对系统带来的冲击；二是可以便于实现代码的复用，提高生产效率。

1.2.2 继承和重用

当电脑工程师在设计制造新型号的电脑时，可以有两种选择：一种是直接从零、从草图开始，另一种则是对现有的型号加以改造升级。虽然当前的型号已经可以满足要求，但是如果能够增加更多的功能、提高性能，市场需求会更好。工程师肯定不愿意从零开始，而是希望利用已有的技术，在现有的款式的基础上增加功能，提高性能。这样，新款电脑可以很快就设计制造出来，被赋予一个新的型号。这是继承和重用的实例。

C++ 采用继承支持重用的思想，程序可以在扩展现有类型的基础上声明新的类型。现有类型称为父类或者基类，从父类派生出来的类型，称为子类或者派生类。继承描述了类之间的共享机制，子类可以重用父类的属性、操作和方法。

1.2.3 多态

何谓多态？顾名思义，多态是指一个事物的多种形态。通过继承的方法构造类，使用多态性为每个子类指定具体表现行为。例如：设计学生类作为父类，具有计算平均成绩的操作，小学生和中学生分别作为子类继承自学生类，计算平均成绩的操作也被继承下来，

对于小学生来讲，计算平均成绩包括数学、语文等基础课程，而对于中学生来讲，计算平均成绩包括数学、语文、外语、物理和化学等科目。

1.3 C++基础知识

C++语言是由C语言发展而来的，但C++语言不仅仅是C语言的扩展。这节介绍C++语言的特点、程序基本结构和上机实践的基本过程。

1.3.1 C++的历史

C++语言是在C语言的基础上发展来的，C语言的起源和发展大概是在1969年到1973年之间。之所以被称为“C”，是因为C语言的很多特性是由一种更早的被称为B语言的程序语言中发展而来。早期操作系统的核心大多由抽象程度较低的汇编语言组成，随着C语言的发展，C语言逐渐被用来编写操作系统的核心。1973年，Unix操作系统的核心正式由C改写完成，这是C语言第一次应用在操作系统的核心编写上。

在C语言发展过程中，随着面向对象概念的产生和发展，出现了称为C with Class的版本，由贝尔实验室的Bjarne Stroustrup博士设计实现，作为C++语言的早期版本，在C语言中增加class关键字和类的概念，在命名时，以C语言中的++运算符来体现它是C语言的进步，因此称之为C++，同时成立了C++标准委员会。一开始C++是作为C语言的增强版出现，从给C语言增加类开始，不断有新特性加入进来，主要有以下几种：

- 虚函数（virtual function）
- 运算符重载（operator overloading）
- 多重继承（multiple inheritance）
- 模板（template）
- 异常（exception）
- 运行时类型信息RTTI（Run - Time Type Identification）
- 命名空间（name space）

1998年国际标准组织（ISO）颁布了C++程序设计语言的国际标准ISO/IEC 1488 - 1998。C++是具有国际标准的编程语言，通常称作ANSI/ISO C++，这是一个统一、完整和稳定的系统，拥有一个丰富强大的标准程序库。

1.3.2 简单的C++程序介绍

下面介绍一个非常简单的C++程序，了解C++程序的组成和建立程序的过程。在这里不详细介绍所有的细节，因为这些内容将在后面的章节中进一步探讨。

例1-1 基本C++程序。

```
#include <iostream>
using namespace std;
int main(){
  cout <<"The wealth of the mind is the only wealth.";
```

```
    return 0;
}
```

这个简单的C++程序由2部分组成：预处理部分和语句部分。

预处理部分第一行#include <iostream>为头文件包含，头文件包含定义了一组可以在需要时包含在源程序中的标准功能。C++标准库中提供的功能存储在头文件中。在这个程序中，名称cout在头文件iostream中声明。这是一个标准的头文件，它提供了在C++中使用标准输入和输出功能所需要的声明。

预处理部分第二行using namespace std;为应用命名空间，C++标准库中的实体包括代码中使用的cout对象都是在命名空间std中定义的，所以标准库中的所有实体名都用std来限定。

语句部分由一个函数main（）组成，函数是代码的一个自包含块，用一个名称来表示，在本例中是main。程序中还可以有许多其他代码，但每个C++程序至少要包含函数main（），且只能有一个main（）函数。C++程序的执行总是从main（）中的第一条语句开始。

例1-1的第一行语句是int main（），这行语句指出，这是函数main的开始。开头的int表示这个函数在执行完后返回一个整数值。因为这是函数main（），所以最初调用它的操作系统会接收到这个值。

函数main（）包含两个可执行语句，每个语句放在一行上：

```
cout <<"The wealth of the mind is the only wealth.";
return 0;
```

这两个语句会按顺序执行。通常情况下，函数中的语句总是按顺序执行，除非有一个语句改变了执行顺序。

main（）中的第一行代码利用插入运算符<<把字符串“The wealth of the mind is the only wealth.”放在输出流中，从而把它输出到屏幕上。在编写涉及到输入的程序时，应使用提取运算符>>。

在C++中，输入和输出是使用流来执行的。如果要从程序中输出消息，可以把该消息放在输出流中，如果要输入消息，则把它放在输入流中。因此，流是一种抽象的表示。在程序执行时，每个流都关联着某个设备，关联着数据源的流就是输入流，关联着数据目的地的就是输出流。这种抽象的优点是无论流代表什么，编程都是相同的。例如，从磁盘文件中读取数据的方式与从键盘上读取完全相同。在C++中，标准的输出流和输入流称为cout和cin，在默认情况下，它们分别对应计算机屏幕和键盘。

函数体中的第二条语句：return 0;代表程序结束，把控制权返回给操作系统。它还把值0返回给操作系统。它也可以返回其他值，来表示程序的不同结束状态，操作系统可以利用该值来判断程序是否执行成功。一般情况下，0表示程序正常结束，非0值表示程序非正常结束。

1.3.3 C++上机实践

首先启动VC集成开发环境，新建项目，如图1-1所示。

在“新建项目”对话框中，选择 Projects 标签，打开该选项卡，在左侧视图区选择 Win32 Console Application 项目，右侧 Project name 文本框中输入项目名称，Location 文本框输入项目保存位置，单击 OK 按钮，在“新建项目类型”向导对话框中，选择“A ‘Hello, World!’ Application”选项，如图 1－2 和图 1－3 所示。

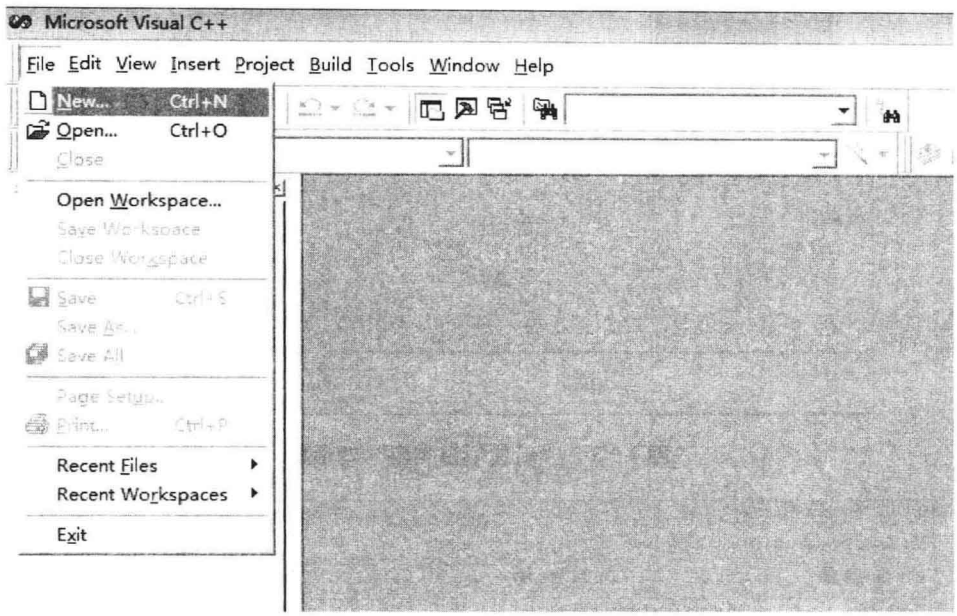


图 1－1　VC 新建项目

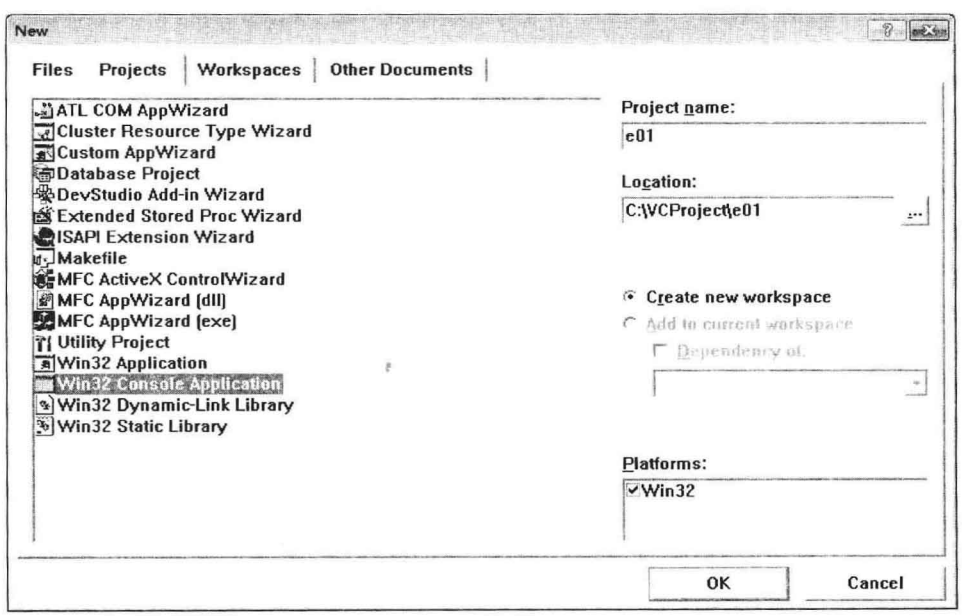


图 1－2　“新建项目”对话框

默认新建项目的源代码如图 1－4 所示，参照例 1－1，修改项目源代码，如图 1－5 所示。

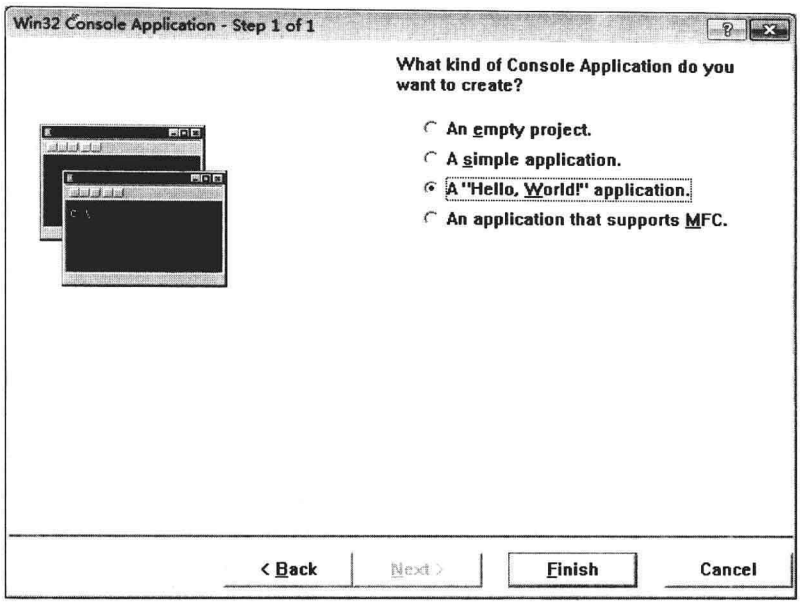


图 1－3　“新建项目类型”对话框

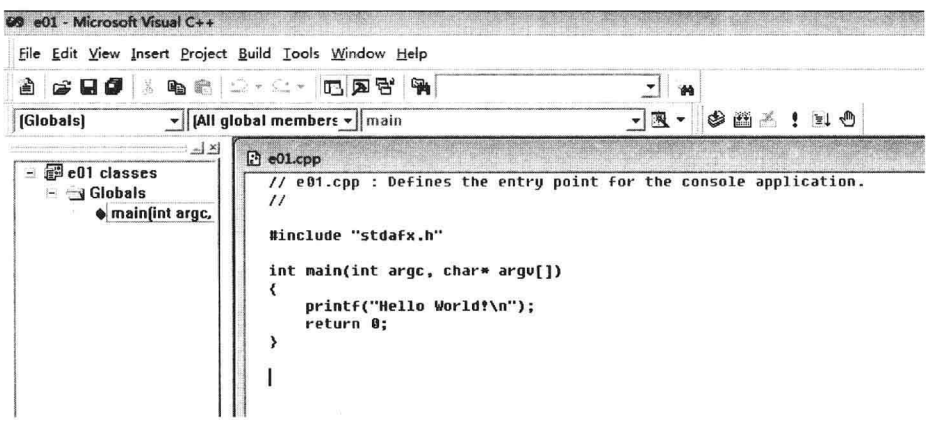


图 1－4　“默认项目”源代码窗口

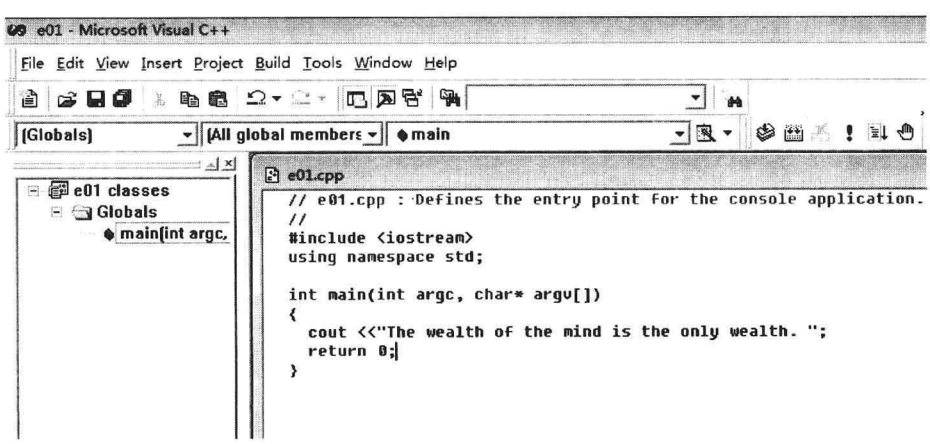


图 1－5　“当前项目”源代码窗口