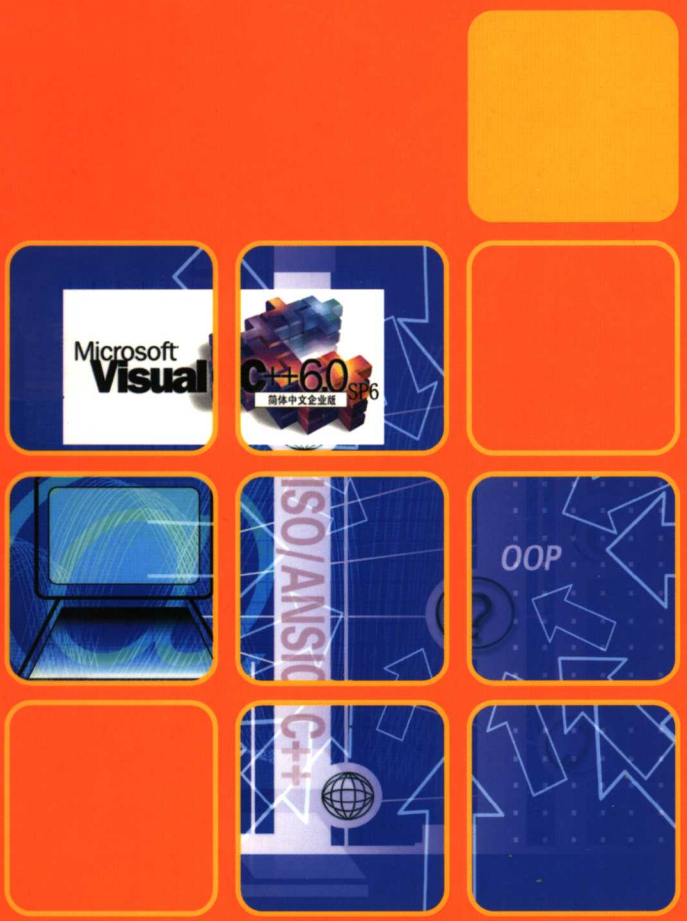
计算机应用能力培养丛书

程序设计技术与C++语言简明教程

陈笑 陈晓霞 李维杰 编著



清华大学出版社

计算机应用能力培养丛书

程序设计技术与C++语言简明教程

陈　笑　陈晓霞　李维杰　编著

清华大学出版社

北　京

内 容 简 介

本书从C++的基本概念和编程方法入手，介绍了C++面向对象程序设计的方方面面，内容十分丰富，包括 C++程序设计的基础知识、基本数据类型和表达式、C++的程序控制语句、数组与函数、指针和引用、C++的高级数据类型、类和对象、继承和派生类、多态与虚函数、C++的I/O机制、异常处理和命名空间等。

本书内容丰富，讲解通俗易懂，提供的大量简短精辟的代码有助于初学者理解问题的精髓。通过本书的学习，读者能够更好地理解面向对象编程的思想，为以后的学习打下坚实基础。本书具有很强的操作性和实用性，可作为高等院校、高职学校“C++程序设计”或“程序设计技术”课程的教材，也可作为各类培训班“C++程序设计与开发”课程的教材，同时本书也是广大 C++编程开发爱好者非常实用的自学参考书籍。

图书在版编目(CIP)数据

程序设计技术与C++语言简明教程/陈笑，陈晓霞，李维杰　编著. —北京：清华大学出版社，2006.9
(计算机应用能力培养丛书)
ISBN 7-302-13408-1
Ⅰ. 程…　Ⅱ. ①陈…②陈…③李…　Ⅲ. C语言—程序设计—教材　Ⅳ. TP312

中国版本图书馆CIP数据核字(2006)第078055号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：王　军
文稿编辑：杜一民
封面设计：康　博
版式设计：康　博
印 刷 者：北京季蜂印刷有限公司
装 订 者：三河市新茂装订有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：23　**字数**：559千字
版　　次：2006年9月第1版　2006年9月第1次印刷
书　　号：ISBN 7-302-13408-1/TP·8421
印　　数：1～5000
定　　价：29.80元

前　言

高职高专教育以就业为导向，以技术应用型人才为培养目标，担负着为国家经济高速发展输送一线高素质技术应用人才的重任。近年来，随着我国高等职业教育的发展，高职院校数量和在校生人数均有了大幅激增，已经成为我国高等教育的重要组成部分。

根据目前我国高级应用型人才的紧缺情况，教育部联合六部委推出“国家技能型紧缺人才培养培训项目”，并从 2004 年秋季起，在全国两百多所学校的计算机应用与软件技术、数控项目、汽车维修与护理等专业推行两年制和三年制改革。

为了配合高职高专院校的学制改革和教材建设，清华大学出版社在主管部门的指导下，组织了一批工作在高等职业教育第一线的资深教师和相关行业的优秀工程师，编写了适应新教学要求的计算机系列高职高专教材——《计算机应用能力培养丛书》。

《计算机应用能力培养丛书》主要面向高等职业教育，遵循“以就业为导向”的原则，根据企业的实际需求来进行课程体系设置和教材内容选取。根据教材所对应的专业，以“实用”为基础，以“必需”为尺度，为教材选取理论知识；注重和提高案例教学的比重，突出培养人才的应用能力和实际问题解决能力，满足高等职业教育“学校评估”和“社会评估”的双重教学特征。

每本教材的内容均由“授课”和“实训”两个互为联系和支持的部分组成，“授课”部分介绍在相应课程中，学生必须掌握或了解的基础知识，每章都设有“学习目标”、“实用问题解答”、“小结”、“习题”等特色段落；“实训”部分设置了一组源于实际应用的上机实例，用于强化学生的计算机操作使用能力和解决实际问题的能力。每本教材配套的习题答案、电子教案和一些教学课件均可在该丛书的信息支持网站(http://www.tupwk.com.cn/GZGZ)上下载或通过Email(wkservice@tup.tsinghua.edu.cn)索取，读者在使用过程中遇到了疑惑或困难可以在支持网站的互动论坛上留言，本丛书的作者或技术编辑会提供相应的技术支持。

C++语言是目前最流行的面向对象程序开发语言之一，和其他面向对象语言相比，C++更能表达编程的本质。C++的应用层面十分广泛，不仅适用于高层次的快速程序开发，也能胜任 C 语言所支持的系统底层开发任务。此外，熟练掌握 C++的人员也能够快速转移到 Java 和 C＃语言的开发。

本书依据教育部《高职高专教育计算机公共基础课程教学基本要求》编写而成，从 C++面向对象程序设计的基础知识和编程技巧入手，由浅入深、循序渐进地介绍了利用 C++进行程序开发的方方面面。在内容编排上充分考虑到初学者的实际需求，通过大量的程序示例来帮助读者理解复杂的理论知识。全书共分 15 章，其中，第 1、2 章介绍 C++程序设计基本概念和知识；第 3~7 章介绍 C++面向过程程序设计知识和编程技巧，包括程序控制语

句、数组、指针和函数、预处理等内容；第 8~12 章介绍面向对象程序设计的知识和编程技巧，包括 C++高级数据类型、类和对象、继承性与派生类、多态与虚函数等；第 13 章介绍 C++的输入输出机制；第 14 章介绍异常处理和命名空间；第 15 章为实训部分，通过学习读者能巩固前面所学知识并提高将它们应用于实际编程开发中的能力。

由于计算机科学技术发展迅速，再者受自身水平和编写时间所限，错误或不足之处在所难免，欢迎广大读者对我们提出意见或建议。

作　者

2006 年 4 月

目　录

第 1 章

程序设计导论

本章主要介绍 C++语言的历史和特性，程序开发的过程，使用 VC 编写 C++程序的基本方法，输入输出语句的基本使用以及 C++编码规范。通过本章的学习，应该完成以下学习目标：

- ☑ 了解 C/C++的历史
- ☑ 理解程序开发的过程和编译的作用
- ☑ 掌握 Visual C++ 6.0 的使用方法，能够创建工程和源程序
- ☑ 掌握使用 Visual C++ 6.0 编译和运行程序的方法
- ☑ 掌握程序的基本结构
- ☑ 掌握 cin 和 cout 语句的使用方法
- ☑ 了解 C++的编码规范

1.1 C++ 概 述

C++语言是目前最流行的编程语言之一，使用它可以进行面向对象程序开发。和其他面向对象语言相比，C++更能表达编程的本质；C++的应用层面非常广泛，不仅适用于高层次的快速程序开发，也能胜任 C 语言所支持的系统底层开发任务；熟练掌握 C++的人员也能够快速转移到 Java 和 C# 语言的开发。当然，学习 C++语言并不是一件十分容易的事，本书将深入浅出地引领读者进入 C++的世界，首先了解一下 C++语言的发展历程。

1.1.1 C 语言的历史

C 语言是由 Unix 操作系统的创始人丹尼斯·里奇(Dennis Ritchie)和肯·汤普逊(Ken Thompson)于 1970 年在 B 语言的基础上发展和完善起来的一种计算机高级语言，是面向过程开发的经典代表。

在结构化编程出现之前，大型程序非常难编写，因为程序中杂乱无序的跳转、调用和返回使得人们很难理清其中的头绪。结构化编程语言通过添加定义清晰的控制语言、带有局部变量的子例程和其他改进措施解决了这个问题。尽管当时还出现了其他的结构化语言，例如 Pascal，但是 C 语言率先实现了集功能强大、简洁易用和表达式丰富等特征于一身，并兼具高级语言和低级语言的特点，特别适合于编写系统软件，因此引起了人们的广泛重视。在短短的十几年中，C 语言风靡全世界，用 C 语言编程成了程序开发人员的一项

基本功。

1.1.2 C++的诞生

C 语言是结构化和模块化的语言，它是面向过程的。在处理小规模的程序时，程序员用 C 语言还比较得心应手，但是随着软件规模的扩大，程序员使用结构化编程的方法往往感到力不从心。为了解决这种问题，出现了新的编程方法——面向对象编程(Object-Oriented Programming，OOP)，利用面向对象编程方法，程序员可以处理更大型、更为复杂的程序。为了让当时使用人数最多的、已经流行了近 20 年的 C 语言能平滑过渡到面向对象语言，AT&T Bell 实验室的 Bjarne Stroustrup 博士和其同事，于 80 年代初期在 C 语言基础上成功开发出了 C++语言。

C++继承了 C 语言原有的所有优点，增加了面向对象的机制。由于 C++对 C 的改进主要体现在增加了适用于面向对象程序设计的“类”，因此最初它被 Bjarne Stroustrup 称为“带类的 C”。后来为了强调它是 C 的增强版，用了 C 语言中的自加运算符“++”，改称为 C++。

C++由 C 发展而来，与 C 完全兼容。用 C 语言编写的程序基本上可以不加修改地用于 C++环境。C++是 C 的超集，既可用于面向过程的结构化程序设计，又可用于面向对象的程序设计，是一种功能强大的混合型程序设计语言。

1994 年 1 月 25 日，由 ANSI(American National Standards Institute，美国国家标准化协会)和 ISO(International Standards Organization，国际标准化组织)组成的联合标准化委员完成了第一个 C++标准草案。该草案完成不久，由于标准模板库 STL 的出现，标准委员会建议将其包含到 C++规范中。STL 的加入大大扩展了 C++的功能，同时也延缓了 C++的标准化工作。直到 1998 年 C++的 ANSI/ISO 标准才最后形成。

尽管设计 C++最初的目标是帮助管理超大型的程序，但是它的作用并不局限于此。事实上，C++的面向对象的功能可以高效地运用在所有编程任务上。C++在编辑器、数据库、个人文件系统、联网工具和通信程序等项目中的应用随处可见。由于 C++继承了 C 语言的高效性，所以许多大型高性能系统软件都使用 C++创建。同时，C++也是编写 Windows 程序时最常使用的语言。

虽然 C++是一种面向对象语言，但是学习面向对象不应该和面向过程对立起来，两者是互补的。在面向对象中仍然要使用大量的结构化程序设计知识，例如函数、方法中的内部实现都离不开面向过程。有些简单问题，直接用面向过程的方法实现起来更简便。

C++同 Java 和 C#相比，有何区别？

前者和后两者的主要区别在于计算环境的类型不同。C++适用于为特定类型的 CPU 和操作系统编写高性能的程序。例如，如果编写一个在 Intel Pentium 处理器上运行的 Windows 程序，C++就是最好的选择。

Java 和 C#的开发则是为了满足 Internet 联机环境下的编程需求。Internet 连接着许多不同类型的 CPU 和操作系统。因此，创建跨平台、可移植程序的能力就成为压倒一切的要求。

1.2　程序开发的过程

本节介绍程序的开发过程。C++编写好的完整程序被称为源程序，源程序在形式上只是一个或一组按某种规范编写的文本文件，要让这些文件变成计算机能够理解的机器语言就必须使用编译器。任何高级语言从源程序转化为机器语言都要经过编写、编译、调试和运行这几个过程，如图 1-1 所示。

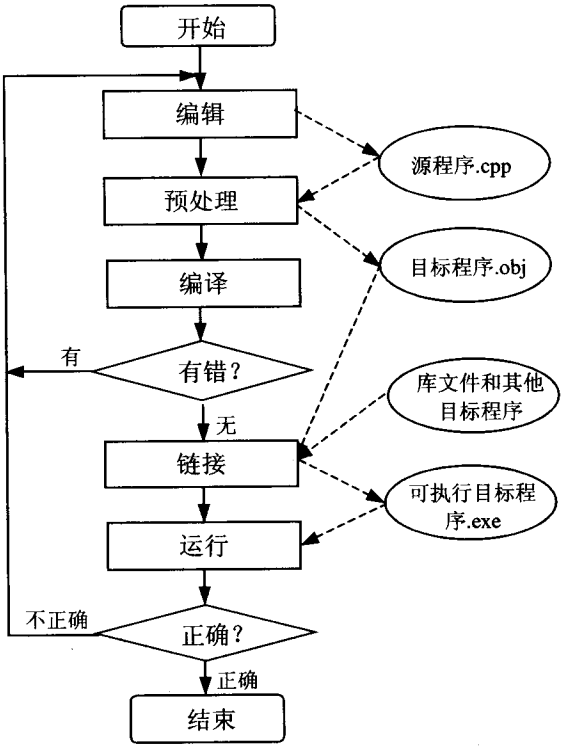


图 1-1　从源程序转化为机器语言的过程

1.2.1　编写程序

编写程序是指把源程序输入到计算机中并生成磁盘文件的过程。C++程序的编写可以使用计算机的文本编辑工具或者某种专门的编辑器，但是不要使用 Word 之类的字处理软件，因为字处理软件通常在文本之中还存储格式信息，这会导致编译错误。编写好的源代码要用扩展名.CPP(即 C Plus Plus 的缩写)保存在磁盘上，供编译器调用。

事实上，C++编译器都提供了编辑器，使用 C++编译器提供的编辑器编写 C++程序是很方便的，比如本书使用的微软 Visual C++开发环境，就是一个建立在微软 C++编译器基础上的编辑器，功能非常强大。编写好代码之后，只要在菜单中选择编译命令就可以编译执行，不必手动输入复杂的编译命令。

1.2.2 编译程序

C++是以编译方式实现的高级语言，其程序的实现，必须要借助编译器。编译器的功能是将程序的源代码转换成机器代码的形式，即目标代码；然后，再使用目标代码进行链接，生成可执行文件并运行。

编译可以分成3个子过程：预处理、编译和链接。

1. 预处理

源程序进行编译时，先进行预处理，如果源程序中有预处理命令，则先执行这些预处理命令，然后再进行下面的编译过程。如果程序中没有预处理命令，就直接进行代码的编译。

2. 编译

编译主要是指进行词法分析和语法分析的过程，又称源程序分析。该阶段基本上与机器硬件无关，分析过程中，如果发现有不符合要求的语法错误，便及时报告给用户并显示在屏幕上。在该过程中还会生成一个符号表。最终生成目标代码程序。下面对编译过程中的主要工作原理进行简要介绍，这将有助于读者理解C++语言和程序。

- **词法分析**：主要是对由字符组成的单词进行词法分析，检查这些单词的使用是否正确，删除程序中冗余的代码。单词是程序使用的基本符号，是最小的程序单元。按照C++的词法规则逐一检查，并登记注册。如发现错误，则及时显示错误信息。
- **语法分析**：语法又称文法，主要是指构造程序的格式。分析时按该语言中使用的语法规则来分析检查每条语句的逻辑结构是否正确，如发现有错误，便及时通报用户。
- **符号表**：符号表又称为字段。它用来映射程序中的各种符号及它们的属性。例如某个变量的类型，占据内存的大小和分配内存的相对位置等。符号表是在进行词法分析和语法分析时生成的，它在生成中间代码和可执行的机器代码时使用。
- **错误处理程序**：在进行词法分析和语法分析过程中将遇到的语法错误交给该程序处理，该程序根据出现的错误性质分为警告错误和致命错误显示给用户，并且尽可能指出出错的原因，供用户修改程序时参考。
- **生成目标代码**：将词法分析和语法分析的结果以及符号表中的信息，由中间代码生成机器可以执行的目标代码，并将这些代码以.obj为扩展名存在磁盘文件中，这些文件称为目标代码文件。机器可以识别目标代码文件中的代码，但并不能直接执行，还需要将目标代码进行链接，生成可执行文件。

3. 链接

这是编译的最后一个过程。编译后的目标代码文件还不能由计算机直接执行。其主要原因是编译器对每个源文件分别进行编译，如果一个程序有多个源文件，编译后这些源文件的目标代码文件将分布在不同地方，因此需要把它们链接到一块儿。即使该程序只有一个源文件，这个源文件生成的目标代码文件还需要系统提供的库文件中的一些其他代码，因此，也需要把它们链接起来。用户程序生成的多个目标代码文件和所需库文件之间的链接是通过链接程序(又称链接器)来完成的。链接程序将由编译器生成的目标代码文件和库文件中所需的文件进行链接处理，生成一个可执行文件，扩展名为.exe，因此，有人又称它为exe文件。库文件的扩展名为.lib。

C++的编译器有哪些？如何选择使用？

标准 C++是一个公开的标准，技术实力雄厚的厂商都开发了自己的编译器，这些编译器完全或基本完全支持标准 C++，一般的 C++源程序在这些编译器上都可以进行编译。

这些厂商包括 IBM、Borland、微软、Watcom 和 Symantec 等公司，产品的名称分别是 Visual Age C/C++、Borland C/C++，Visual C/C++、Watcom C/C++和 Symantec C/C++。其中 Borland C/C++和微软的 C/C++是市场占有率最高的 C++编译器，历史上 Borland C++ 3.1 版本曾经盛极一时，在当时有着最快的编译速度，如果读者有兴趣可以把 Borland C++找来，作为自己学习 C++的工具，就像 Turbo C 一样，这个开发工具也是运行在 DOS 下的。

但是随着 Windows 平台的普及，在 Windows 下的可视化开发工具渐渐成了主流。微软在 Windows 平台上推出的 C++程序开发工具 Visual C/C++(可视化 C++，简称 VC)，目前其最高版本是 6.0。相对于其他编程工具，Visual C++在提供可视化编程方法的同时，也适用于编写直接对系统进行底层操作的程序，其生成代码的质量也要优于其他的很多同类产品。

Borland 公司在 Windows 平台上也推出了可视化的 C++开发工具—— Borland C++ Builder(简称 BCB)，目前最高版本 6.0。Borland C++ Builder 是一个 BC 编译器，是用来优化 BC 开发系统的工具，它包括对最新版本的 ANSI/ISO C++语言的支持，包括 RTL，C++的 STL(标准模板库)框架结构等。C++ Builder 的优势在于它内部的 VCL(可视化组件库)体系提供了强大功能，可以比 VC 更容易开发出可视化的 Win32 程序。

1.2.3　运行程序

C++的源程序经过编译和链接后生成了可执行文件。运行可执行文件的方法很多，一般在编译器提供的编辑器中就有运行的功能，通过选择相应菜单命令即可运行，也可以通过在 Windows 系统中双击可执行程序来运行。如果需要运行参数，还可以在编译环境提供的编辑器中设置相应的参数。

程序运行后，一般在屏幕上显示出运行的结果。用户可以根据运行结果来判断程序是否还有要改进的地方。程序在编译后，在生成可执行文件之前需要改正编译和链接时出现的一切致命错误和警告错误，这样才可能生成无错的可执行文件。在程序中存在警告错误时，也会生成可执行文件，但是一般要求改正警告错误后再去运行可执行文件，有的警告错误会造成运行结果错误。

VC 和 C++是两种语言吗？

不是。Visual C++提供的 Microsoft 基础类库(Microsoft Foundation Class Library，简写为 MFC)，对 Windows 所用的 Win32 应用程序接口(Win32 Application Programming Interface)进行了十分彻底的封装，这使得可以使用完全面向对象的方法来进行 Windows 应用程序的开发，从而大量节省了应用程序的开发周期，降低了开发成本，也使得 Windows 程序员从大量的复杂劳动中解脱出来。正是因为 VC 提供了如此多的高级特性，以至于有些人误认为 VC 和 C++是两种不同的语言，要特别说明的是，VC 是一种 C++的集成开发环境(IDE)，它虽然有 MFC、ATL 等高级功能支持快速开发 Win32 程序，但它本质上还是 C++程序，请不要混淆这二者的概念。本书中的示例代码都是按照标准 C++的编码规范在 VC 6.0 中完成的，如果要移植到其他的编译器上，请参考其他编译器的编码规范说明。

1.3 用 VC 开发一个简单的 C++程序

本节指导读者使用 Visual C++开发第一个 C++程序。在正式编写程序前，读者应正确安装 Visual C++ 6.0，并了解其基本使用方法。

1.3.1 启动 Visual C++ 6.0 集成开发环境

安装好 VC 6.0 后，执行【开始】|【程序】|【Microsoft Visual Studio 6.0】|【Microsoft Visual C++ 6.0】 命令，启动 VC 6.0，如图 1-2 所示。

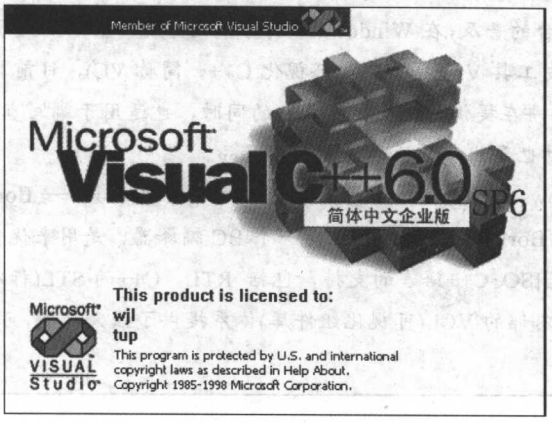


图 1-2 VC 6.0 启动界面

在 VC 6.0 中，工程是组织程序的基本单位，一组相关的文件都将被包含在同一个工程中，在 VC 6.0 主界面上，选择菜单中的【文件】|【新建...】命令，打开【新建】对话框，如图 1-3 所示。

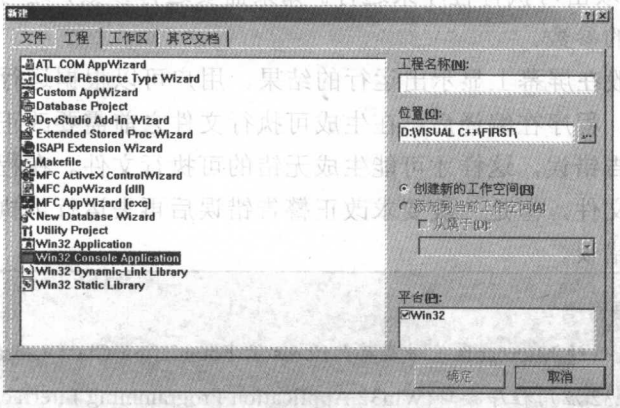


图 1-3 【新建】对话框

在【新建】对话框中，打开【工程】选项卡，在列表框中选择【Win32 Console Application】选项，然后在【工程名称】文本框中输入新建工程的名称 First，在【位置】文本框中输入保存工程的路径，单击【确定】按钮。

这时可以看到如图 1-4 所示的对话框，选中【一个空工程】单选按钮，单击【完成】

按钮，此时出现【新工程信息】窗口，单击【确定】按钮即可。

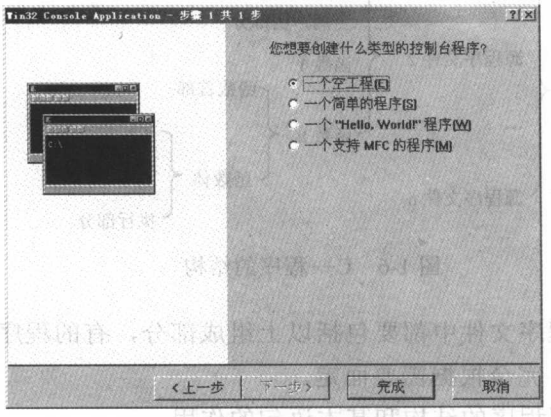


图 1-4　选择空工程

现在已经有了一个新的空工程，下面通过新建 C++源文件的方式向工程中添加空的程序文件。选择【文件】|【新建...】命令打开【新建】对话框，然后打开【文件】选项卡，再在列表中选择【C++ Source File】选项，然后在【文件名】文本框中输入文件名 hello.cpp，再选中【添加到工程】复选框，在下拉列表中选择存放新建文件的工程，这里只有刚才已经建好的 First 工程，然后在【位置】文本框中输入保存文件的路径。单击【确定】按钮即可，如图 1-5 所示。

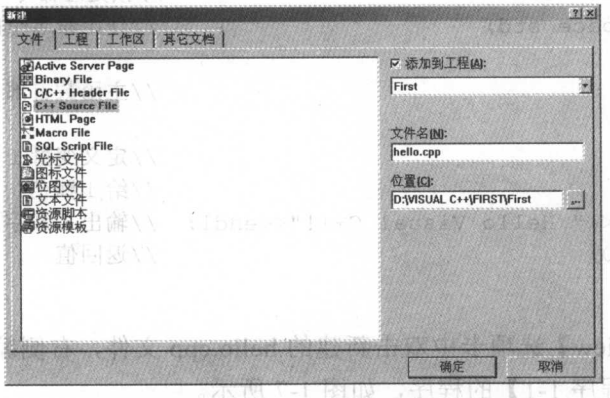


图 1-5　新建 C++源文件

建好空的源文件后，在开发环境中左侧的【File View】选项卡中可以看到整个工程中文件的组织情况，包括刚才新建的文件，此时已经做好了编写程序的准备。

1.3.2　编写程序

在 C++程序中，一个程序可以由一个程序单位或多个程序单位组成。每个程序单位作为一个文件。在程序编译时，编译系统分别对每个文件进行编译。因此，一个文件是一个编译单元。一般情况下，一个程序单元包括预处理、全局声明和函数三个部分。C++程序的结构如图 1-6 所示。

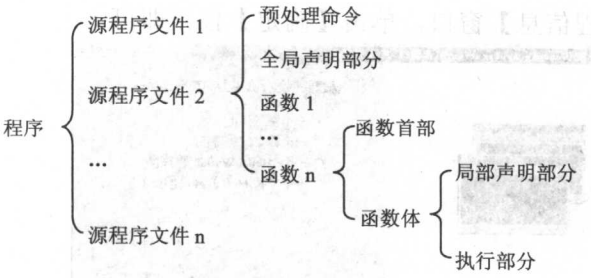


图 1-6　C++程序的结构

当然，并非每一个程序文件中都要包括以上组成部分，有的程序中不包含全局声明部分，也有的不包括函数，完全根据需要而定。

下面通过例子来介绍程序的结构和基本语句的作用。

【程序 1-1】编写一个简单的 C++程序。

程序 1-1

```
/*
This is a simple C++ program.

Call this file Hello.cpp
*/
#include <iostream>                          //预处理命令
using namespace std;                         //使用命名空间 std

int main()                                   //主函数首部
{
    int i;                                   //定义变量 i
    i=1;                                     //给 i 赋值
    cout<<i<<" Hello Visual C++!"<<endl;     //输出 i 和字符串
    return 0;                                //返回值
}
```

在左侧的【File View】选项卡中双击新建的 hello.cpp 文件，右侧编辑区域出现空白的输入区，在此输入【程序 1-1】的程序，如图 1-7 所示。

图 1-7　输入程序