21世纪高等学校规划教材 | 计算机科学与技术

C++程序设计

闵联营 何克右 主编

清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

C++程序设计

闵联营 何克右 主编

清华大学出版社
北京

内 容 简 介

C++是当今应用领域最广泛的程序设计语言，它被用于从系统软件到各种大型应用系统的开发。由于其语言设施的丰富、高效、灵活，C++能支持从面向过程、面向对象，到泛型编程等多种程序设计范型。

本书是一本全面、系统地介绍C++程序设计的教程。全书共分13章，从内容上组织为5个部分。第1部分是C++概述；第2部分介绍C++面向过程的程序设计方法，主要包括C++的基本数据类型、程序控制语句、数组和结构体、函数、指针和引用等内容；第3部分作为一个过渡，用一个具体的实例探讨了程序设计从过程抽象到数据抽象的转变；第4部分讲述C++面向对象的程序设计方法，包括类和对象、继承和派生、多态性和虚函数、流类库等内容；第5部分探讨C++支持泛型编程的模板机制和泛型编程的实际应用标准模板库STL等内容。

本书内容全面、实例丰富、语言简洁、通俗易懂，注重理论与实践相结合。书中所有例题均在Visual C++ 6.0上运行通过。

本书可以作为高等院校计算机专业和相关专业程序设计课程的教材和计算机专业面向对象程序设计教材，也可以作为全国计算机等级考试（二级 C++）的培训教材和参考书，还可供各类程序设计培训班学员和C++语言自学者参考。

图书在版编目(CIP)数据

C++程序设计/闵联营，何克右主编. —北京：清华大学出版社，2010.9
（21世纪高等学校规划教材·计算机科学与技术）
ISBN 978-7-302-22911-7

Ⅰ. ①C…　Ⅱ. ①闵…　②何…　Ⅲ. ①C语言－程序设计－教材　Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2010）第100210号

责任编辑：魏江江
责任校对：时翠兰
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京四季青印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：25.75　**字　数**：621千字
版　　次：2010年9月第1版　　**印　　次**：2010年9月第1次印刷
印　　数：1～3000
定　　价：39.00元

产品编号：034749-01

编审委员会成员

（按地区排序）

清华大学	周立柱　教授
	覃　征　教授
	王建民　教授
	冯建华　教授
	刘　强　副教授
北京大学	杨冬青　教授
	陈　钟　教授
	陈立军　副教授
北京航空航天大学	马殿富　教授
	吴超英　副教授
	姚淑珍　教授
中国人民大学	王　珊　教授
	孟小峰　教授
	陈　红　教授
北京师范大学	周明全　教授
北京交通大学	阮秋琦　教授
	赵　宏　教授
北京信息工程学院	孟庆昌　教授
北京科技大学	杨炳儒　教授
石油大学	陈　明　教授
天津大学	艾德才　教授
复旦大学	吴立德　教授
	吴百锋　教授
	杨卫东　副教授
同济大学	苗夺谦　教授
	徐　安　教授
华东理工大学	邵志清　教授
华东师范大学	杨宗源　教授
	应吉康　教授
上海大学	陆　铭　副教授
东华大学	乐嘉锦　教授
	孙　莉　副教授
浙江大学	吴朝晖　教授

扬州大学	李善平 教授
南京大学	李 云 教授
	骆 斌 教授
	黄 强 副教授
南京航空航天大学	黄志球 教授
	秦小麟 教授
南京理工大学	张功萱 教授
南京邮电学院	朱秀昌 教授
苏州大学	王宜怀 教授
	陈建明 副教授
江苏大学	鲍可进 教授
武汉大学	何炎祥 教授
华中科技大学	刘乐善 教授
中南财经政法大学	刘腾红 教授
华中师范大学	叶俊民 教授
	郑世珏 教授
	陈 利 教授
江汉大学	颜 彬 教授
国防科技大学	赵克佳 教授
中南大学	刘卫国 教授
湖南大学	林亚平 教授
	邹北骥 教授
西安交通大学	沈钧毅 教授
	齐 勇 教授
长安大学	巨永峰 教授
哈尔滨工业大学	郭茂祖 教授
吉林大学	徐一平 教授
	毕 强 教授
山东大学	孟祥旭 教授
	郝兴伟 教授
中山大学	潘小轰 教授
厦门大学	冯少荣 教授
仰恩大学	张思民 教授
云南大学	刘惟一 教授
电子科技大学	刘乃琦 教授
	罗 蕾 教授
成都理工大学	蔡 淮 教授
	于 春 讲师
西南交通大学	曾华燊 教授

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展，顺应并符合21世纪教学发展的规律，代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色

精品教材包括：

（1）21 世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

（2）21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

（3）21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

（4）21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

（5）21 世纪高等学校规划教材·信息管理与信息系统。

（6）21 世纪高等学校规划教材·财经管理与计算机应用。

（7）21 世纪高等学校规划教材·电子商务。

清华大学出版社经过二十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
联系人：魏江江
E-mail:weijj@tup.tsinghua.edu.cn

前言

C++是当今应用领域最广泛的程序设计语言，它被用于从系统软件到各种大型应用系统的开发。由于其语言设施的丰富、高效、灵活，C++能支持从面向过程、面向对象，到泛型编程等多种程序设计范型。

目前，国内高校普遍开设了“面向对象程序设计”之类的课程，一些高校将C++语言作为程序设计语言课程的首选语言，故而需要有一本全面系统地介绍C++程序设计语言的教材。为此，我们在总结多年教学实践经验的基础上编写了这本书。

1. 本书内容

本书全面、系统地介绍了C++面向过程、面向对象的程序设计方法和泛型编程。

全书共分13章，从内容上组织为5个部分。第1章是C++概述，介绍了C++的发展和特点、C++程序的基本框架，以及C++程序的开发过程等。第2～6章介绍C++面向过程的程序设计方法，主要介绍了C++的基本数据类型、运算符和表达式、程序控制语句、数组和结构体、函数、指针和引用等。第7章作为一个过渡，用一个具体的实例探讨了程序设计从过程抽象到数据抽象的转变，简单介绍了面向对象的基本特征和概念。第8～10章和13章讲述C++面向对象程序设计的方法，主要包括类和对象、继承和派生、多态性和虚函数、I/O流类库等内容。第11和12两章探讨C++支持泛型编程的模板机制和泛型编程的实际应用标准模板库STL等内容。

2. 本书学习方法

由于C++语言是从C语言发展而来，为了与C语言兼容，C++允许继续使用C语言的语法。笔者认为，既然学习C++，就应遵循C++标准，故本书所有的语法和程序都是依据C++标准介绍的。在教学实践中作者感到学生对面向对象的基本特征和概念的理解有一定的困难，因此专门用一章的篇幅介绍了程序设计从过程抽象到数据抽象的转变，逐步引入面向对象的概念，帮助读者更好地理解面向对象程序设计。对于学过C语言而需要学习面向对象程序设计的读者，可以直接从第7章开始学习，本书将是一本很好的教材。另外，由于泛型编程近年来得到了大量关注和应用。因此，本书第12章用较大的篇幅对标准模板库STL的相关知识进行了深入的阐述。

本书内容全面，语言简洁，通俗易懂，注重理论与实践相结合。书中所有例题均在Visual C++ 6.0上运行通过。

本书由闵联营、何克右主编，刘传文、伍新华和谭新明参加编写。其中，第1、2、5章由闵联营编写，第3、4、6章由何克右编写，第7、10章由谭新明编写，第8、9、13章由伍新华编写，第11和12两章由刘传文编写。闵联营审校全书。

由于作者水平有限，书中难免有不足之处，恳请读者批评指正。

编　者

2010年8月于武昌

目 录

第 1 章 C++概述 ····· 1

1.1 程序设计语言 ····· 1
1.1.1 机器语言 ····· 1
1.1.2 汇编语言 ····· 2
1.1.3 高级语言 ····· 2
1.2 算法及算法的表示方法 ····· 4
1.2.1 算法的概念 ····· 4
1.2.2 算法的表示方法 ····· 5
1.3 C++的发展和特点 ····· 8
1.3.1 C++的发展 ····· 8
1.3.2 C++的特点 ····· 9
1.4 C++程序的基本框架 ····· 9
1.4.1 最简单的 C++程序 ····· 9
1.4.2 结构化程序设计框架 ····· 12
1.4.3 面向对象程序设计框架 ····· 13
1.5 C++程序的开发过程 ····· 15
1.5.1 C++程序开发的一般过程 ····· 15
1.5.2 用 Visual C++开发 C++应用程序的步骤 ····· 16
习题 ····· 18

第 2 章 C++数据类型 ····· 20

2.1 基本数据类型 ····· 20
2.1.1 数据在计算机中的存储 ····· 20
2.1.2 标识符 ····· 21
2.1.3 数据类型修饰符 ····· 22
2.2 常量和变量 ····· 23
2.2.1 常量 ····· 23
2.2.2 变量 ····· 25
2.3 运算符和表达式 ····· 27
2.3.1 算术运算符和算术表达式 ····· 27
2.3.2 赋值运算符和赋值表达式 ····· 28
2.3.3 关系运算和逻辑运算 ····· 29

2.3.4 条件运算符 …… 31
2.3.5 逗号运算符 …… 31
2.3.6 位运算符 …… 32
2.3.7 sizeof 运算符 …… 32
2.3.8 运算符优先级与结合性 …… 33
2.3.9 数据类型转换 …… 34
2.4 数据的输入和输出 …… 35
2.4.1 用 cout 进行输出 …… 35
2.4.2 用 cin 进行输入 …… 36
习题 …… 37

第 3 章 程序控制语句 …… 38

3.1 选择语句 …… 38
3.1.1 if 语句 …… 38
3.1.2 switch 语句 …… 43
3.2 循环语句 …… 45
3.2.1 while 语句 …… 45
3.2.2 do 语句 …… 47
3.2.3 for 语句 …… 48
3.2.4 循环的嵌套 …… 50
3.2.5 break 和 continue 语句 …… 52
3.2.6 goto 语句 …… 53
习题 …… 54

第 4 章 数组和结构体 …… 55

4.1 数组 …… 55
4.1.1 数组的定义和存储 …… 55
4.1.2 数组元素的引用 …… 57
4.1.3 数组的初始化 …… 58
4.1.4 数组的应用 …… 61
4.2 结构体 …… 70
4.2.1 结构体概述 …… 70
4.2.2 结构体类型变量的定义和引用 …… 71
4.2.3 结构体数组 …… 74
习题 …… 75

第 5 章 函数 …… 77

5.1 函数概述 …… 77
5.2 函数的定义和调用 …… 79

5.2.1 函数定义的一般形式 …… 79
5.2.2 函数的调用 …… 80
5.2.3 函数原型 …… 82
5.2.4 函数的参数传递 …… 83
5.3 函数的嵌套和递归 …… 94
5.3.1 函数的嵌套调用 …… 94
5.3.2 函数的递归调用 …… 95
5.4 内联函数 …… 97
5.5 函数重载 …… 98
5.6 变量的作用域和存储期 …… 101
5.6.1 程序的内存区域 …… 101
5.6.2 局部变量和全局变量 …… 103
5.6.3 变量的存储期 …… 106
5.6.4 存储类别小结 …… 110
5.7 预处理命令 …… 112
5.7.1 宏定义命令 …… 112
5.7.2 文件包含命令 …… 113
5.7.3 条件编译命令 …… 114
习题 …… 116

第 6 章 指针和引用 …… 117

6.1 指针 …… 117
6.1.1 地址和指针的概念 …… 117
6.1.2 指针的定义和使用 …… 118
6.1.3 指针与数组 …… 124
6.1.4 指针与函数 …… 133
6.1.5 指针与结构体 …… 136
6.1.6 const 修饰符和指针 …… 145
6.2 引用 …… 146
6.2.1 引用的概念 …… 146
6.2.2 引用作函数参数 …… 148
6.2.3 引用返回值 …… 149
6.2.4 用 const 限定引用 …… 151
习题 …… 151

第 7 章 从过程抽象到数据抽象 …… 153

7.1 从过程抽象到数据抽象 …… 153
7.1.1 集合的实现与使用 …… 153
7.1.2 将集合的实现与使用分开 …… 156

7.1.3 将集合用链表实现 …… 159
7.1.4 将集合的数据表示和操作封装在一起 …… 161
7.2 面向对象程序设计的基本概念和特征 …… 165
7.2.1 面向对象程序设计的基本概念 …… 165
7.2.2 面向对象程序设计的三大特征 …… 167
7.2.3 从面向过程的程序设计到面向对象的程序设计 …… 172
习题 …… 175

第 8 章 类和对象 …… 176

8.1 类的定义 …… 176
8.1.1 类的定义 …… 176
8.1.2 类的数据成员 …… 177
8.1.3 类的成员函数 …… 178
8.1.4 类成员的访问控制 …… 180
8.2 对象 …… 181
8.2.1 对象的定义和使用 …… 181
8.2.2 对象在内存中的存放 …… 183
8.2.3 this 指针 …… 185
8.3 构造函数和析构函数 …… 186
8.3.1 构造函数 …… 186
8.3.2 带默认参数的构造函数 …… 188
8.3.3 重载构造函数 …… 189
8.3.4 复制构造函数 …… 190
8.3.5 析构函数 …… 194
8.3.6 成员对象的构造和析构 …… 195
8.4 对象数组和对象指针 …… 198
8.4.1 对象数组 …… 198
8.4.2 对象指针 …… 200
8.5 静态成员 …… 203
8.5.1 静态数据成员 …… 204
8.5.2 静态成员函数 …… 205
8.6 友元 …… 206
8.6.1 友元函数 …… 207
8.6.2 友元类 …… 209
8.7 常类型 …… 209
8.7.1 常对象 …… 209
8.7.2 用 const 修饰的类成员 …… 210

习题 212

第 9 章 继承与派生 214

9.1 派生类 214
9.1.1 派生类的定义 214
9.1.2 派生类生成过程 215
9.1.3 派生类的构造函数和析构函数 218
9.2 访问控制 220
9.2.1 公有继承 221
9.2.2 私有继承 222
9.2.3 保护继承 223
9.3 虚基类 225
9.3.1 多继承中的二义性问题 225
9.3.2 虚基类的概念 229
9.3.3 虚基类的构造函数 230
习题 234

第 10 章 多态性 236

10.1 多态性概述 236
10.1.1 静态联编 236
10.1.2 动态联编 238
10.2 运算符重载 240
10.2.1 运算符重载的方法和规则 240
10.2.2 运算符重载为成员函数 244
10.2.3 运算符重载为友元函数 247
10.3 虚函数 248
10.3.1 虚函数的定义及使用 248
10.3.2 动态联编的一种典型实现方案 254
10.3.3 纯虚函数和抽象类 256
10.3.4 虚函数的应用示例 260
习题 264

第 11 章 模板 265

11.1 模板的概念 265
11.2 函数模板和模板函数 266
11.2.1 函数模板的定义和模板函数的生成 266
11.2.2 函数模板的使用 268
11.2.3 函数模板的特化 270
11.2.4 用函数模板实现基本算法 271

11.3 类模板和模板类 273
11.3.1 类模板的定义 273
11.3.2 类模板的使用 275
11.3.3 类模板的派生 278
11.3.4 类模板的特化 282
11.3.5 string 类的使用 285
习题 290

第 12 章 标准模板库 STL 292

12.1 标准模板库 STL 概述 292
12.1.1 STL 的发展历史 292
12.1.2 STL 的组成 293
12.2 容器类 295
12.2.1 顺序容器 296
12.2.2 关联容器 313
12.3 STL 算法 323
12.3.1 for_each 324
12.3.2 count 和 count_if 325
12.3.3 copy 和 remove 327
12.3.4 transform 329
12.4 迭代器 331
12.4.1 迭代器的分类 331
12.4.2 迭代器的特性 335
12.4.3 traits 技术 337
12.4.4 与迭代器相关的函数 342
12.4.5 指针与迭代器 343
12.5 函数对象 345
12.5.1 函数对象的概念 345
12.5.2 STL 中的函数对象 348
12.6 适配器 349
12.6.1 容器适配器 349
12.6.2 迭代器适配器 355
12.6.3 函数适配器 361
习题 371

第 13 章 C++的 I/O 流类库 372

13.1 C++的流及流类库 372
13.1.1 C++的流 372
13.1.2 流类库 372

13.2 格式化 I/O ····· 375
13.2.1 用 ios 类的成员函数进行格式控制 ····· 375
13.2.2 使用 I/O 操纵符进行格式控制 ····· 378
13.3 重载 I/O 运算符 ····· 380
13.3.1 重载输出运算符“<<” ····· 380
13.3.2 重载输入运算符“>>” ····· 382
13.4 文件流 ····· 383
13.4.1 文件的打开和关闭 ····· 383
13.4.2 文件的读写 ····· 386
习题 ····· 392

第1章 C++概述

程序设计语言是人们为了描述计算机解决问题时的计算过程而设计的一种具有语法、语义描述的记号。C++是近年来国内外广泛使用的一门高级程序设计语言，它既支持面向过程的程序设计，也支持面向对象的程序设计。C++程序的实现过程一般要经过编辑、编译、连接和运行4个步骤。

1.1 程序设计语言

一个完整的计算机系统包括硬件系统和软件系统两大部分。所谓硬件是指构成计算机的物理设备，是计算机完成计算工作的物质基础。所谓软件是指使计算机运行所需要的程序以及程序运行时所需要的数据和有关的技术文档资料。没有软件，计算机是一台“裸机”，是什么也不能干的；有了软件，计算机才能灵动起来，成为一台真正的“电脑”。

要使计算机完成各种预定的操作，不仅应该告诉计算机做什么，而且还要告诉计算机如何做，这都是通过计算机执行一条条指令来完成的。

指令是指挥计算机完成指定操作的命令，它在计算机中是以一组二进制代码来表示的，一条指令对应计算机的一定动作。一台计算机所有指令的集合称为这台计算机的指令系统。指令系统的完善和齐全程度在一定程度上反映了这台计算机的功能与作用的强弱，它是由计算机的硬件决定的。不同的CPU具有不同的指令系统，通过执行各种指令可以使计算机完成预定的操作。

用计算机进行数据处理时，要把处理的内容、步骤和运算规则用一系列指令表达出来，这一系列指令的有序集合就称为程序。程序通过输入设备送入计算机的存储器中存储起来，然后根据程序的要求一条条执行其中的指令，这样计算机的各部件就会在程序控制下自动完成指令规定的各种操作，操作完成后，通过输出设备送出结果，这就是存储程序的基本思想，它是由美国计算机科学家冯·诺依曼提出来的。

程序是用计算机程序设计语言编写的。程序设计语言在发展的过程中经历了由低级到高级的发展过程，可以分为机器语言、汇编语言和高级语言。

1.1.1 机器语言

计算机所能处理的最基本的信息单位是二进制数字，即计算机只能识别理解由“0”与“1”构成的二进制序列。计算机能够直接识别和执行的二进制指令称为机器指令，机器

指令的集合就是机器语言指令系统，简称为机器语言，它是最早的程序设计语言。

例如，在8088 CPU构成的微型计算机上计算两个整数（18和37）的和，并将结果存放在内存数据段的0000H存储单元的机器语言程序为：

```
10111000 00010010 00000000      ;将整数18送CPU内的寄存器AX中暂存
00000101 00100101 00000000      ;将整数37加到CPU内的寄存器AX中
10100011 00000000 00000000      ;将CPU寄存器AX中保存的和送内存单元中
```

机器语言是由一连串的0和1组合起来的二进制编码组成的，用机器语言编写的程序可以直接被计算机识别和执行，执行效率高、速度快。由于机器语言是特定于机器的，不同的机器有不同的指令系统，人们无法把为一种机器编写的程序直接搬到另一种机器上运行。一个问题如果要在多种机器上求解，那么就要对同一问题编写多个应用程序，造成了重复工作。而且编写机器语言程序是一种非常枯燥而烦琐的工作，要记住每一条指令的编码与含义极其困难，编写出的程序既不易阅读也不易于修改。

1.1.2 汇编语言

为了克服机器语言的缺点，使得更多的人可以使用程序设计语言，人们对机器语言进行了改进，使用一些便于记忆的助记符号来代替机器指令，比如，用“MOV”代表数据传送，用“ADD”代表加法等。使用这些助记符号代替机器指令所产生的语言称为汇编语言。

例如，在8088 CPU构成的微型计算机上计算两个整数（18和37）的和，并将结果存放在内存数据段的0000H存储单元的汇编语言程序可写为：

```
MOV  AX,18              ;将整数18送CPU内的寄存器AX中暂存
ADD  AX,37              ;将整数37加到CPU内的寄存器AX中
MOV  [0000],AX          ;将CPU寄存器AX中保存的和送内存单元000H中
```

汇编语言中所使用的助记符号可以让人们很容易地理解指令的含义，便于记忆和使用。汇编语言比机器语言易读、易懂，而且修改也较为方便。然而计算机是不能识别和执行这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

汇编语言一般都是针对特定的计算机系统设计的，因此它对机器的依赖性很强，移植性不好，但效率高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精练且质量高，所以至今它仍是一种常用而强有力的程序设计语言。

尽管与机器语言相比，汇编语言的抽象程度要高得多，但由于汇编指令与机器指令之间是一对一的关系，用汇编语言编写哪怕是一个很简单的程序，也要使用数百条指令。为了解决这个问题，人们又研制出了宏汇编语言，一条宏汇编指令可以翻译成多条机器指令，这使得人们的程序设计工作量得以减少。为了解决由多人编写的大程序的拼装问题，人们又研制出了连接程序，它用于把多个独立编写的程序块连接组装成一个完整的程序。

1.1.3 高级语言

汇编语言从可读性方面已经比机器语言有了很大的进步，但仍然没有摆脱指令系统的束缚，也不符合人们的表达习惯。人们意识到，应该设计一种接近于数学语言或人的自然语言（例如计算公式更接近于数学公式，易于理解），同时又不依赖于计算机硬件，编出的程序能