HZ BOOKS

高等院校计算机教材系列

C#程序设计大学教程

罗兵 刘艺 孟武生 等编著

为教师配有电子教案和习题答案

机械工业出版社
China Machine Press

高等院校计算机教材系列

C#程序设计大学教程

罗兵 刘艺 孟武生 等编著

机械工业出版社
China Machine Press

本书以C#语言为载体，通过讨论C#程序设计的一般过程和方法，重点讲述了程序设计基础、面向对象程序设计、算法与数据结构、GUI程序设计和数据库程序设计的知识，并涉及计算机基础、数据和控制、程序设计理论、软件工程4大知识领域。同时，本书详细分析了C#作为通用程序设计语言的本质特征和语法规则，并以大量C#程序实例演示应用程序的设计过程，介绍主流的程序设计思想和方法，培养读者的代码编写能力。

本书采用案例教学法，既有丰富的理论知识，也有大量的实战范例，更提供了精心设计的课后练习。

本书内容深入浅出，覆盖面广，图文并茂，独具特色，适合作为计算机及其相关专业本科教学用书，也可用做其他专业的计算机公共课基础教材。对于自学程序设计的计算机爱好者以及从事软件开发和应用的科技人员来说，本书也是极佳的参考书。

图书在版编目（CIP）数据

C#程序设计大学教程 / 罗兵等编著. －北京：机械工业出版社，2007.8
（高等院校计算机教材系列）
ISBN 978-7-111-21721-3

Ⅰ. C…　Ⅱ. 罗…　Ⅲ. C语言－程序设计－高等学校－教材　Ⅳ. TP312

中国版本图书馆CIP数据核字（2007）第093873号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：迟振春
北京京北制版厂印刷 · 新华书店北京发行所发行
2007年8月第1版第1次印刷
184mm × 260mm · 19.75印张
定价：30.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

前　言

欢迎进入C#的世界学习计算机程序设计课程。这将是一次美妙和激动人心的探索，可能会为你今后从事充满挑战和令人兴奋的职业奠定软件编程的基础。众所周知，计算机在我们的日常生活中扮演了一个重要的角色，而且在未来仍将继续扮演这一重要角色。

计算机科学是一个充满挑战和发展机遇的年轻学科，而计算机程序设计则是这门学科的重要基础。随着计算机在各行各业的广泛应用，很多非计算机专业也把计算机程序设计列为公共基础课之一。

既然是作为基础课的教材，那么本书假定读者既不具有程序设计经验，也不具备面向对象技术的概念和Windows程序设计知识。即使是一个对计算机一无所知的人，也能通过本书获取所有有关的基本知识，进入程序设计的世界。如果读者是一位很有经验的程序员，已用过其他程序设计语言，并掌握了一定的开发技能，也能在本书中发现很多有用的信息。

本书与程序设计课程

计算机程序设计既是一门概念复杂、知识面广的理论课，也是一门面向实战、需要动手的实践课。几乎所有的编程初学者都梦想着有朝一日能在计算机上驰骋，让一行行代码在自己敲击键盘的手下源源不断地流出，真正成为驾驭计算机的主人。然而，学完程序设计课程后，实际开始编写程序时，却往往会觉得难以下手、无所适从。尽管自己刻苦学习，高分通过考试，但并不能体会到所学的知识给实际编程带来的便利和优势。

为什么会这样？一方面是因为我们的学生在学习时没有掌握程序设计的一般过程，没有深入了解通用程序设计语言的本质规律；另一方面是因为我们的教学体制僵化、教材陈旧，教学思想和内容跟不上时代的发展，与软件开发实际情况脱节。

计算机程序设计语言是一种实现对计算机操作和控制的人造语言，与人类的自然语言有一定差距。程序设计语言仅仅是程序设计的手段和途径，而不是程序设计的全部。因此，掌握程序设计语言并不意味着就精通程序设计，就能写出优秀的程序。实际上，程序设计所涉及的领域、知识和技能要远远超出我们的想象。因此，本教材对于程序设计课程在一些方面有着自己独特的理解。

程序设计首先是一个过程

程序设计过程通常分为问题建模、算法设计、编写代码和编译调试4个阶段。不同阶段的任务是相对独立的，不能混为一谈。即使是一个比较简单的程序，我们也应该养成先分析，再下手，最后调试的习惯，严格遵循程序设计过程。因为在缺乏对问题深入、全面分析的情况下，就匆匆动手编写程序，将会增加失败的风险，带来后期修改、维护的麻烦。因此，要学习程序设计，不但不能回避程序设计过程，而且还要从软件开发过程和软件生命周期的高度来了解和掌握程序设计过程，从一开始就养成遵从程序设计准则的良好习惯。有别于其他程序设计教材，本书强调程序设计过程和软件开发过程的重要性，向读者介绍了有关软件建模与测试的基本原理和技术。特别考虑到现代软件开发依赖于集体合作和项目管理，是一个汇集了很多程序设计过程的更大的过程。因此，除了在书中增加有关软件过程实施和管理的介绍外，还把如何撰写

规范的程序代码作为重要一节，使得读者在学习程序设计之初就了解程序设计的规范，注重编写程序的规范性、正确性和可靠性，从而为将来参与大型软件开发打下良好基础。

程序设计还是一种解决问题的方法和能力

程序设计课程的目标是学习用计算机解决问题的思考方法，培养编程应用能力，而不是仅仅学会某个程序设计语言的语法规则。很多学生能弄清楚顺序、选择和循环结构以及算术表达式，但很难把一个编程问题分解成结构良好的程序。这暴露了程序设计教学中偏重语法细节，忽略总体思想方法和整体过程实现的问题。

尽管程序设计理论的发展为解决问题提供了很多有效方法，但对于初学者而言，学习的捷径应该是抓住最核心的思想方法，即结构化方法和面向对象方法。为实现这个目的，我们既把结构化算法分析和设计作为教材重点，也把面向对象分析和设计作为重点。对于前者，我们以顺序结构、选择结构和循环结构这三种基本结构为基础，讲解常用的结构化算法；对于后者，我们则围绕面向对象的抽象性、继承性、多态性和封装性这4个本质特点阐述面向对象程序设计的基本方法。通过强调基本概念、基本方法、基本应用，为初学者奠定扎实的程序设计基础，树立良好的编程思想。通过大量的实例分析和范例程序设计过程演示，我们力图给初学者建立完整印象，培养其从整体上思考问题和解决问题的编程能力。

程序设计最终是对程序设计语言的应用

程序设计和程序设计语言存在着有趣的辩证关系。程序设计可以用不同的程序设计语言来实现，而不同的程序设计语言又决定着能使用怎样的程序设计思想和技术，制约着程序设计的实现能力和效率。本书之所以使用微软推出的面向对象语言C#作为学习程序设计的语言，并不是因为C#有强大的可视化编程能力，而是因为C#不但继承了众多程序设计语言的优点，而且还具有面向对象语言的优势。更可喜的是，C#还在继续发展，不断吸取现代编程语言的精华。就在本书采用目前最新的C# 2.0编写时，微软已经在计划C# 3.0了。这一切使得C#具备现代通用程序设计语言的主流特征，成为微软.NET平台的主打语言。因此，学习C#语言，掌握C#程序设计方法是本课程的另一个重要任务。

本书虽然以C#语言为背景介绍程序设计语言的相关知识，但是重点强调的是一些通用的思想方法，而放弃了C#的一些奇技淫巧。读者应该注意到，不同的程序设计语言其语法和风格可能迥异，但无论哪一种语言，都是以数据（类型）、操作（运算）、控制（逻辑流程）为基本内容。更进一步讲，学习一门程序设计语言，应该超越语言的具体表述格式，不拘泥于繁芜的语法现象，站在抽象的高度，掌握程序设计的基本概念，深入了解程序设计语言的本质规律。这样将会为深入学习其他程序设计语言带来便利。

本书的结构

这本书是为计算机程序设计课程编写的。计算机程序设计课程是理工科类大学的公共基础课，它通过讲授一门具体的计算机语言，来帮助学生掌握程序设计的基础知识和基本应用。同时，对于未接触过计算机科学的学生，本书还涉及和介绍了与程序设计相关的计算机科学知识。程序设计基础、面向对象程序设计和Windows程序设计这3个部分组成了本书的核心知识，并涉及计算机基础、数据和控制、程序设计理论、软件工程知识4大知识领域，汇集20个相关知识点。虽然本书中讨论的内容有一定的理论性，但这些理论都是用实际程序问题表达的，这是为了帮助读者建立完整的程序设计知识体系结构。

本书的组织结构如图1所示，其中有些知识点是通过各章节的迭代，循序渐进，不断深入的。

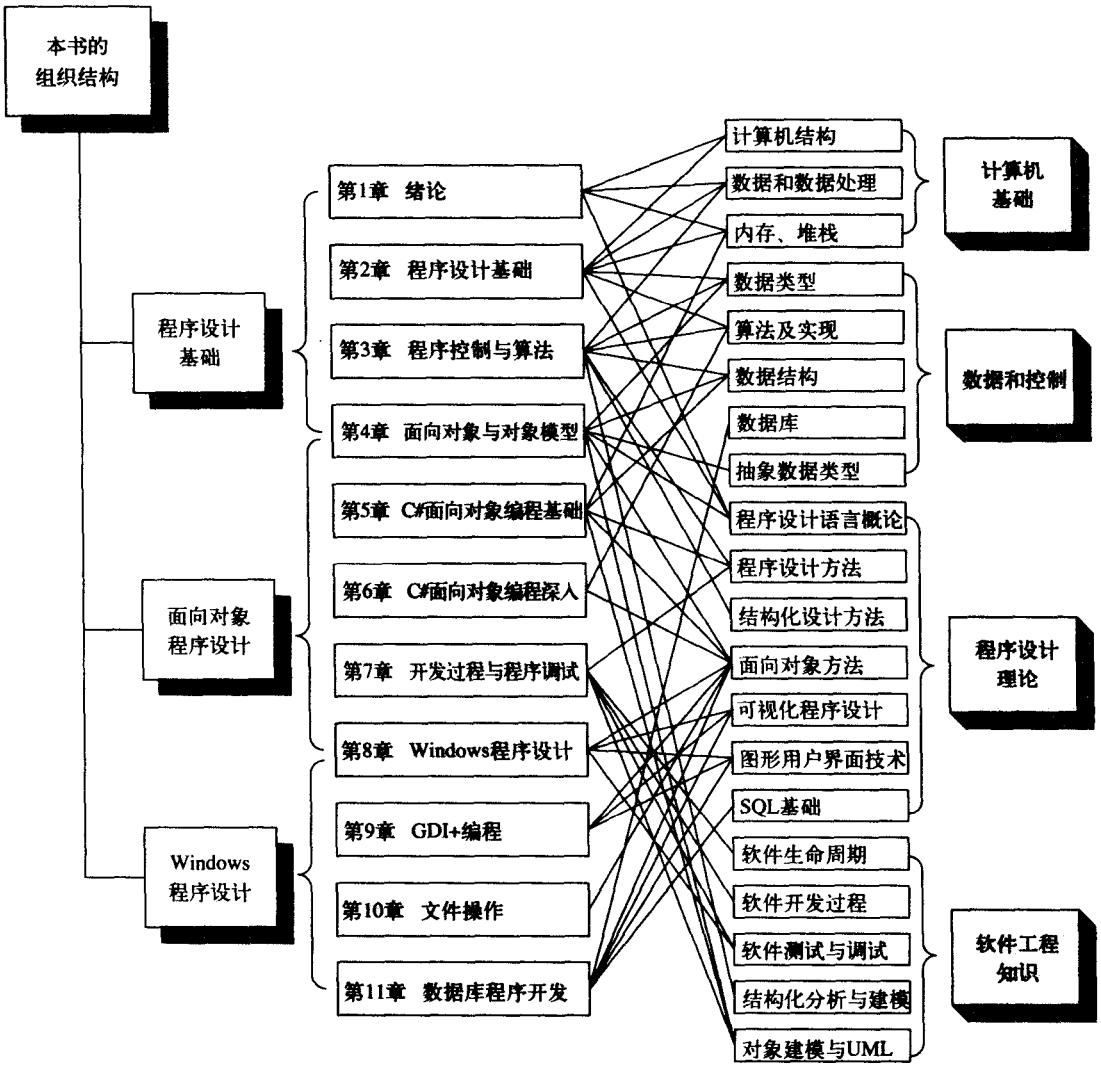

图1 本书的组织结构

本书共有11章，各章的主要内容如下：

- **第1章 绪论** 介绍程序设计的基本概念，重点帮助读者搞清计算机程序、程序设计、程序设计语言等概念。初步认识.NET和C#，了解C#的特点，并简要介绍Visual C# 2005 Express。
- **第2章 程序设计基础** 首先通过入门示例程序，了解程序的组成结构、语言要素和编写规范，建立程序的基本概念。然后，通过讲解数据和数据类型、变量和常量，开始探索C#程序设计语言。
- **第3章 程序控制与算法** 介绍C#运算符和表达式，并学习顺序、选择、循环三种控制结构，帮助读者完整地理解如何控制程序的逻辑流程。在此基础上，讲解算法的相关概念和几个常用算法。
- **第4章 面向对象与对象模型** 主要介绍面向对象的基本概念和对象建模的方法，剖析类和对象的关系，讲解类成员（属性、索引和方法）的定义和对象的生命周期。

- **第5章 C#面向对象编程基础** 围绕面向对象的抽象性、继承性、多态性和封装性这4个特点讲解面向对象程序设计的基本方法，并引入C# 2.0中的新概念——泛型，对其进行了简要介绍。
- **第6章 C#面向对象编程深入** 重点讲授如何使用对象接口和迭代器，介绍接口和迭代器的实现方法。详细讲授委托和事件，并介绍匿名方法。
- **第7章 开发过程与程序调试** 介绍软件的开发过程及过程的实施管理，从软件质量的高度讨论程序的调试与测试，重点讲述Visual C# 2005 Express程序的调试方法和程序中的异常处理。
- **第8章 Windows程序设计** 讲解Windows应用程序的一般设计方法，包括如何创建窗体、设计界面、使用控件、事件处理等。读者可以学到Windows窗体技术在快速可视化开发Windows应用程序方面的强大功能。
- **第9章 GDI+编程** 讲解图形程序设计的基本概念，了解相关的核心类`Graphics`，并分别介绍点、画笔、图形、文本和图像的使用和控制。
- **第10章 文件操作** 介绍文件和流，具体说明文件存储管理的概念和方法，并就文件的读写进行了详细讲解。
- **第11章 数据库程序开发** 介绍一些数据库理论基础和SQL语言基础知识，探讨数据库访问技术的发展过程，详细地介绍ADO.NET的体系结构，讲授数据库应用系统的开发过程和方法。

尽管本书包含以上内容，但实际的教学进度和授课内容可以根据课堂教学的安排或读者的实际技能及对所讨论问题的熟悉程度灵活确定。教学课时数建议安排在30~50课时之间。其中，标有*的章节仅供有能力的学生选学。

本书的读者对象

- 对计算机一无所知的人。这类读者想通过程序设计了解和使用计算机，并通过学习获取所有有关的基本知识。显然，C#是一种比较容易学习和入门的语言。
- 没有任何程序设计基础的学生。这类读者选用本书作为程序设计基础教材，可以掌握C#程序设计语言，为深入学习其他计算机课程奠定基础。显然，C#无论是在结构化、面向对象设计方面，还是在可视化Windows编程方面都能胜任教学的需要，从而为学习程序设计提供更全面的知识结构体系。
- 因工作或科研需要希望迅速掌握一门程序设计语言以完成不太复杂的编程任务的非专业编程人员。
- 非计算机专业的编程爱好者。这类读者改行从事程序员工作，有一些实际的经验，但没有系统学习过相关专业知识。可以通过本书重温程序设计知识，补习相关概念和理论。
- 有一定经验的程序员。这类读者已经学习过其他编程语言并在软件开发中掌握了一定的开发技能，但没有使用过C#语言或对C#语言一知半解。可以利用本书系统学习C#程序设计，发现C#与其所熟悉语言的不同点，并由此掌握C#语言。

本书的特色

本书的特色使得本书与众不同，特别有助于入门者的学习。

概念和知识面

贯穿本书，我们始终强调概念要比数学模型更重要，我们认为对概念的理解必然左右对模

型的理解。同时，我们还特别注意开阔读者的知识面，使读者能够站在现代软件开发和软件工程这个比较开阔的层面上了解程序设计，而不是局限于繁琐的程序设计语言规则。

代码编写能力

初学C#程序设计的人，大多数会被C#强大的可视化编程功能所吸引。的确，快速应用开发（RAD）的思想正在改变程序设计的方法，而可视化程序设计实际上已经重造了开发者的工作平台。以前编写程序是通过基于字符的编辑器键入一条条语句，现在我们可以交互式地在窗体上点击和拖放（click-and-drop）组件，并使用短小精悍的代码段连接它们。过去，即使是开发很小的Windows应用程序，也需要很多细心而且繁杂的基础工作，先写出非常长的源代码文件，然后才可编译和测试其结果。C#改变了这种模式。它首先创建一个Windows应用程序的初始框架代码，即默认的用户界面，而无须写任何程序。程序员的创造力被用于真正的程序设计任务，而不是浪费在窗口的几个控件上。

可是，千万不要被容易的、可视化的、基于组件的程序设计所迷惑，把程序设计变成了点击和拖放组件，然后在窗体上重新排列组件的工作。如果是这样学习程序设计，我们培养的是拖拉现成组件的“拖拉员”，而不是创造代码的程序员。真正的程序设计需要很多知识、技能和创造力。为此，我们的教材不采用大多数C#程序设计教材那种以可视化程序设计为重点的编写模式。为避免过早地引入可视化组件的使用，分散学习程序设计语言基础知识的注意力，本书在讲述程序设计的基本概念和算法设计时，尽量以控制台程序为示例讲解程序设计方法，重在学习程序设计语言的本质，培养代码的编写能力。

图文并茂

本书有大量精心设计的插图，这些插图有助于增进读者对文字的理解。

示例程序

本书尽可能地运用示例程序来表述概念和模型，同时尽量提供完整的示例程序和程序设计过程。本书所有示例程序和习题中的程序都已在Windows下.NET环境中编译运行通过。

另外，本书的部分相关资源可以从http://www.liu-yi.net以及http://www.hzbook.com获得。

CC 2004课程体系

从1990年开始，美国电气和电子工程师协会计算机社团（Computer Society of the Institute for Electrical and Electronic Engineers，简称IEEE-CS）和计算机协会（Association for Computing Machinery，简称ACM）就着手开发新的本科生计算机课程体系。1991年联合推出了Computing Curricula 1991（简称CC 1991），当时仅限于Computer Science（计算机科学）和Computer Engineering（计算机工程）两个专业的课程。1998年秋季开始，IEEE-CS和ACM联合投入新的力量更新该课程体系，并在2001年开发出Computing Curricula 2001（简称CC 2001），并将该计算机课程体系扩大到Computer Science（计算机科学）、Computer Engineering（计算机工程）、Software Engineering（软件工程）、Information Systems（信息系统）等多个专业。在CC 2001的实施中，专家们发现，计算机课程所涉及的学科专业和教学范围正在不断扩大，而且在内容和教学方面的变化也日新月异。IEEE-CS和ACM意识到10年一次的Computing Curricula修订已经难以满足要求，于是联合国际信息处理联合会（International Federation for Information Processing，简称IFIP）、英国计算机协会（British Computer Society，简称BCS）等更多的组织开发了Computing Curricula 2004（简称CC 2004），使之成为开放的、可扩充的、适合多专业的、整合了计算机教学相关原则体系观点的课程体系指南。其结构参见图2。

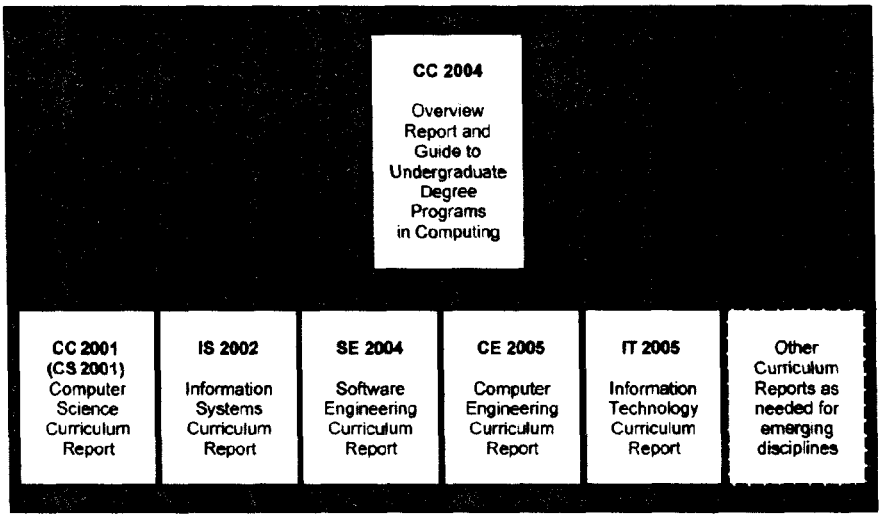

图2 CC 2004体系结构

为了进一步反映当代计算机科学技术的发展水平，与国际主流计算机教育思想接轨，通过多年来对IEEE-CS和ACM的Computing Curricula课程体系的跟踪研究，我们在本教材的编写中，借鉴了CC 2004课程体系的最新研究成果，同时吸取了国外同类教材的优秀经验，目的是进一步推动教材和课程改革，培养有竞争力的人才。

致谢

本书是作者在多年科研和教学基础上编写的，主要参考了作者已发表的文章和著作以及教学中积累的资料。书中还参考了其他中外文教材、资料，由于无法在此一一列举，现谨对这些教材和资料的作者表示衷心的感谢。

参与本教材编写工作的人员除封面署名人员外，还包括海军工程大学的段立、李启元、包磊、李亚楠、鲁晖，海军指挥学院的蔡敏，南京航空航天大学无人驾驶飞机研究所的吴英，以及刘瀚、吴永逸、洪蕾等。

一本书的出版离不开许多人的支持，尤其是这本书。为此感谢我们的家人和朋友，我们在忍受写作之苦的同时，牺牲了与他们共享天伦之乐的宝贵时光。

由于作者水平有限，书中难免有疏漏和不妥之处，恳请各位专家、同仁和读者不吝赐教，并在此表示特别感谢！

2007年6月12日于南京

http://www.liu-yi.net

目　录

第1章 绪 论

1.1 计算机概述

计算机自从问世以来经历了异常迅速的发展，其体系结构也发生了巨大的变化，从早期的以运算器为中心的冯·诺伊曼结构发展到流水线、并行处理和多处理机结构，从传统的指令驱动型计算发展到数据驱动和需求驱动型计算等。计算机的发明和应用对人类社会的发展具有深刻的影响。

1.1.1 计算机系统组成

众所周知，计算机系统是由硬件系统和软件系统组成的。

1. 计算机硬件系统

计算机硬件系统由运算器、控制器、主存储器、输入设备和输出设备五大部件组成（如图1-1所示）。运算器是对数据加工处理的部件，它主要完成算术和逻辑运算。控制器的主要功能是从主存储器中取出指令，并指出下一条指令在主存储器中的位置。取出的指令经指令寄存器送往指令译码器，经过对指令的分析发出相应的控制和定时信息，控制计算机的各个部件有条不紊地工作，以完成指令所规定的操作。存储器是计算机系统的记忆设备，用来存放程序、原始数据、中间结果及最终结果。输入设备的作用是把程序和原始数据转换成计算机中表示的二进制数，输入到计算机的主存中。输出设备的作用是把运算处理结果按照人们所要求的形式输出到外部存储介质上。

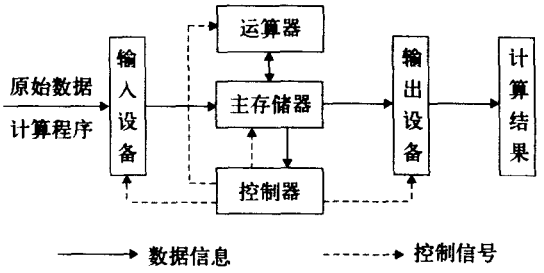

图1-1 计算机基本组成框图

通常，把运算器和控制器合称为中央处理单元，简称CPU；把中央处理单元和主存储器合称为主机。

（1）中央处理单元

中央处理单元（CPU）用于数据的运算。它是计算机的核心部件，包含三个组成部分：算术/逻辑单元（ALU）、寄存器组和控制单元。

算术/逻辑单元用于进行算术运算和逻辑运算。最简单的一元运算是增量（加1）和减量（减1）运算，最简单的二元运算是加、减、乘和除。最简单的一元逻辑运算是非，最简单的二元逻辑运算是与、或和异或。控制单元的作用就是负责选择这些运算中的一种。

寄存器是用来存放临时数据的高速独立的存储单元。CPU的运算离不开大量寄存器的使用。现代计算机中的寄存器按功能分为如下三类：

- **数据寄存器** 过去，计算机只有一个寄存器用来交替保存输入的数据（其他输入数据直接来自内存）或结果。现在，越来越多的复杂运算改由硬件设备（而不是使用软件）实现，并且需要数据寄存器来保存这些运算的中间结果，这也是在计算机CPU内部需要大

量的寄存器来提高运算速度的原因。

- **指令寄存器** 现在，计算机存储的不仅是数据，而且还在内存中存储相对应的程序。CPU从内存中逐条地取出指令，并将取出的指令存储在指令寄存器中，解释并执行指令。
- **程序计数器** CPU中另一个通用寄存器是程序计数器。程序计数器中保存着当前正在执行的指令。当前的指令执行完后，计数器将自动加1，指向下一条指令的内存地址。

CPU的第三个组成部分是控制单元，控制单元类似于人脑中控制身体各部分运动的区域，其控制是通过线路的开（高电平）或关（低电平）来实现的。

（2）输入/输出子系统

计算机中的另一个子系统是称为输入/输出（I/O）子系统的一系列设备。这个子系统可以使计算机与外界通信，并在断电的情况下存储程序和数据。输入/输出设备可以分为两个大类：非存储设备和存储设备。

- **非存储设备** 非存储设备使得CPU/内存可以与外界通信，但它不能存储信息。两个最常见的非存储设备是键盘和显示器。键盘提供输入功能，显示器显示输出并响应键盘的输入。另一种输入/输出设备是打印机。相信大家对它并不陌生，若以书面方式描述，则可称它是一种用于产生永久性记录的输出设备。虽然打印的材料中存储有信息，但是仍将打印机归为非存储设备，这是因为计算机不能直接通过打印机读取信息（除非有人采用打字或扫描的方式再次输入信息）。
- **存储设备** 它可以存储大量的信息以备后用。依据存储介质的不同，通常分为磁介质存储器和光介质存储器两种，前者包括软盘、硬盘、磁带等，后者包括CD、DVD、EVD等。存储设备要比主存便宜得多，而且它们存储的信息也不易丢失（即使断电，信息也不会丢失）。

（3）主存储器

主存储器是计算机内的另一个子系统，通常称为内存。它是存储单元的集合，每一个存储单元都有惟一的标识，称为地址。数据以字的形式在内存中传入和传出。一个字中包含多个位（位是存储体的基本单位，英文为bit，常简写为b），字可以是8位、16位、32位，甚至是64位，为了表述方便，我们定义了字节的概念，8位称为一字节（英文为byte，常简写为B）。字节在计算机科学中的使用相当普遍，因此，我们通常称16位为2B，32位为4B。

在内存中存取每个字都要有相应的标识符。尽管程序员使用命名的方式来区分一个字（或字的集合），但在硬件层次上，每个字都是通过地址来标识的。所有在内存中标识的独立的地址单元的总数称为地址空间。

由于计算机是以位模式对所存储的数进行运算，因此地址本身也是用位模式表示的。举例来说，如果一个内存是64K（2^{16}），字长为1B，那么其内存地址空间的范围为0～65 535，就需要一个16位的地址。起始地址通常是0000000000000000（十进制地址0），最后一个地址通常是1111111111111111（十进制地址65 535）。通用的计算规律是：如果一个计算机有N个内存字，就需要有$\log_2 N$位地址来确定每一个内存单元。习惯上，内存地址用无符号二进制整数定义。

2. 计算机软件系统

当然，计算机仅具备硬件设备并不能进行运算，它还必须具有软件设备。计算机软件系统由系统软件与应用软件两部分组成，我们首先用计算机语言描述解决问题的步骤，并将其编制成程序（即应用软件），并由输入设备输入到主存储器中，在系统软件的支持下完成运算。系统软件是计算机的设计者提供给用户的，它是方便用户充分发挥计算机效能的一组程序。

系统软件主要包括：

- 操作系统
- 数据库管理系统
- 实用程序
- 语言处理程序

(1) 操作系统

操作系统是系统软件的核心，它负责管理和控制计算机系统的硬件资源和软件资源，是用户和计算机之间的接口。通常，操作系统具有如下功能：

- **进程管理** 主要针对处理机进行管理。为了提高CPU的利用率，采用多道程序技术。进程管理负责协调多道程序之间的关系，使CPU得到充分利用。
- **存储管理** 负责对系统有限的主存空间进行合理的分配，以满足多道程序运行的需要。
- **设备管理** 负责对计算机系统中除了CPU和主存储器以外的所有I / O设备进行管理。它为这些设备提供驱动程序或者控制程序，为用户提供简单易用的接口，同时利用先进的数据传送技术使外围设备尽可能与CPU并行工作，以提高设备的使用效率，从而提高整个系统的运行速度。
- **文件管理** 其任务是有效地组织存储、保护文件，以方便用户的访问。文件是一组相关信息的集合，它包括用户作业、源程序、数据以及各种系统软件。
- **作业管理** 作业是用户在一次运算过程中要求计算机系统所做工作的集合。作业管理的任务是确定用户如何向系统提交作业，以及操作系统如何组织和调度这些作业的运行，以提高整个系统的运行效率。

(2) 数据库管理系统

数据库管理系统（DBMS）实现了数据独立于程序的统一管理。随着计算机技术的发展和计算机应用的普及，计算机要处理的数据越来越多，传统的文件系统导致数据冗余和难以共享，且维护困难，数据的安全保密性和一致性都很差。数据库管理系统解决了这些问题。

(3) 实用程序

实用程序（utility）是指为了帮助用户使用和维护计算机而编制的一组程序。这些程序通常作为操作系统可调用的文件存在，也可将其视为操作系统的扩充部分。

(4) 语言处理程序

语言处理程序是对程序设计语言进行翻译和处理的程序，包括汇编程序、编译程序、链接程序、调试和排错程序等。

1.1.2 数据表示与处理

计算机处理的对象是数据，那么什么是数据呢？

数据是对信息的一种组织和表达形式。古人通过结绳记事，绳子上打的结对他们来说就是数据。计算机通过电子线路的逻辑开关来处理0、1两种信号，二进制的位模式信息对计算机来说就是数据。尽管我们要处理的数据会以数字、文字、图像、声音和视频等不同的形式出现，但对于编程而言都要将这些数据以计算机程序认可的方式输入到计算机中进行处理。

1. 数据的表示

计算机是一个数据处理器，而程序就是处理数据的具体步骤和方法。但是在讨论数据处理之前，首先我们应该知道数据的本质。

我们要处理的数据会以不同的形式出现，如数字、文字、图像、声音和视频。计算机程序主要处理数字（如进行算术运算、解方程等）、文本（如文字调整对齐、移动、删除等）、

图像（图像创建、收缩、放大、旋转等）、音频视频（播放MP3音乐或电影、制作电影特技等）等数据。但是无论数据以何种形式出现，程序处理时所面对的都是同样的二进制位模式。

（1）计算机内部的数据

计算机处理不同类型的数据时采用统一的数据表示法。将以不同形式出现的数据采用统一的数据表示法转换后存入计算机中，当数据从计算机输出时再还原为原来的形式。这种通用的格式称为位模式。

位（bit，binary digit的缩写）是计算机中信息存储的最小单位，它可以是0或1。位代表设备的某一状态，这些设备只能处于两种状态中的某一种状态。例如，开关要么合上要么断开。可以用1表示合上状态，0表示断开状态。但开关只能表示一个位，换句话说，一个开关能存储一个位的信息。

只利用位并不能解决数据表示问题。然而，我们需要存储更大的数字、文本、图形等。为了表示数据的不同类型而使用的位模式是一个序列，这种序列有时也称为位流。它是0和1的组合。这就意味着，如果需要存储一个由16位组成的位模式，那么就需要16个电子开关。如果需要存储1000个位模式，每个16位，那么就需要16 000个开关。

现在的问题是：计算机存储器怎样知道所存储的位模式表示哪种类型的数据？实际上它并不知道。计算机存储器仅仅将数据以位模式存储。解释位模式是数字类型、文本类型，还是其他数据类型的工作则由输入/输出设备或程序完成。换句话说，当数据输入计算机时，它们被编码，当呈现给用户时，它们被解码。通常，长度为8的位模式称为1B。这个术语常用于测量内存或其他存储设备的大小。例如，一台能存储800万位信息的计算机有1MB的内存。

（2）文本数据的表示

位模式可以表示任何一个构成文本的语言符号。但问题是：在一种语言中，位模式到底需要多少位来表示一个符号？这取决于该语言集中到底有多少不同的符号。例如，如果要创建的某个虚构的语言仅仅使用大写的英文字母，则只需要26个符号。这种语言的位模式也至少需要表示26个符号。而对另一种语言（如中文）来说，可能需要更多的符号。在一种语言中，表示某一符号的位模式的长度取决于该语言中所使用的符号的数量。符号越多，位模式越长。

如果需要2个符号，位模式长度将是1位（$\log_2 2 = 1$），如果需要4个符号，长度将是2位（$\log_2 4 = 2$）。2位的位模式能表示四种不同的形式：00，01，10和11。这些形式中的任何一种都可以代表一个字符。同样，3位的位模式有八种不同的形式：000，001，010，011，100，101，110和111。

不同的位模式集合可以用于表示文本符号。我们把一类位模式集合称为码。表示符号的过程称为编码。下面介绍几种常用的码。

- **ASCII码** 美国国家标准协会（ANSI）开发了一种称为美国信息交换标准码（ASCII）的代码。这种代码使用7位表示一个符号，即该代码可以定义128（2^7）种不同的符号。用于表示ASCII码的完整位模式可见附录A。
- **扩展ASCII码** 为了使每个位模式的大小统一为1B（8位），ASCII位模式通过在左边增加额外的0来进行扩充。扩充后，每一个模式就能恰好存入1B大小的内存中。换句话说，在扩展ASCII码中，第一个码是00000000，最后一个是01111111。

  一些制造商曾经决定使用额外的位创建额外的128个符号。然而，这种尝试由于各个制造商最终未能制定出一个统一标准的代码集而并未获得成功。
- **Unicode码** 前面所述的两种代码仅仅能表示英语符号。为了表示更多的符号，需要更大容量的代码。于是，硬件和软件制造商联合起来共同设计了一种名为Unicode的码，

这种代码使用16位表示一个符号，能表示多达65 536个符号。代码被划分成不同部分，用于表示来自世界上不同语言的符号。其中，还有些部分用于表示图形和特殊符号。C#支持Unicode码。

(3) 其他数据的表示

在计算机中，数字是使用二进制系统来表示的。虽然在文本中也包含1，2，…，9和0，但是，计算机中并没有采用文本的编码来表示数字。其主要原因有以下两点：

- 用字符集（如ASCII）表示数字效率不高。例如，使用ASCII码表示65 535需要5B，但在二进制系统中，使用无符号整数表示这个数只需要2B。
- 用字符集表示数字将导致数学运算十分复杂。字符集中的编码方式是为了标识各个字符，并没有考虑到计算的优化。而二进制系统本身就是数学中的概念，数字的位置具有记数意义，类似于日常生活中使用的十进制系统中的个、十、百、千、万，每进一位乘10。两者的区别在于二进制系统中每进一位乘2。

图像在计算机中有两种表示方法：位图图像或矢量图。

- **位图图像** 它采用像素矩阵的形式表示图像，每个像素是一个小点。像素的大小取决于分辨率。例如，图像可以分成1000像素或者10 000像素，后者能得到更好的图像显示效果（较高的分辨率），只是需要更多的空间来存储图像。在把图像分成像素之后，每一个像素被赋值为一个位模式。模式的尺寸和值根据图像的情况来定。如果表示彩色图像，按照三原色理论，通常将每一种彩色像素分解成三种主色：红、绿和蓝（RGB）。然后测出每一种颜色的强度，并用一种位模式（通常8位）表示它。
- **矢量图** 位图图像表示方法存在的问题是一幅特定的图像采用精确位模式表示后必须存储在计算机中。如果想重新调整图像的大小，就必须改变像素的大小，这将产生波纹状或颗粒状的图像（失真）。而矢量图表示方法并不存储位模式，它是将图像分解成一些曲线和直线的组合，其中每条曲线或直线由数学公式表示。例如，一条直线可以通过它的端点坐标来作图，圆则可以通过它的圆心坐标和半径长度来作图。这些公式都被存储在计算机中。当要显示或打印图像时，将图像的尺寸作为输入传给系统。系统重新设计图像的大小并用相同的公式画出图像。在这种情况下，每画一次图像，公式也将重新估算一次。

2. 数据的处理

我们可以通过程序来编写数据处理的步骤和方法。首先，需要根据数据的不同形式（如整数、字符等）把要处理的数据划分成不同的数据类型，并以此来组织数据，然后按照处理需求设计出算法。程序编译通过后，就可以由计算机来集中处理这些数据。

计算机的CPU在重复的机器周期中依次执行编译好的程序指令，一步一步，从开始到结束。一个简化的周期包括三步：取指令、译码和执行，如图1-2所示。

- **取指令** 在取指令阶段，控制单元命令系统将下一条要执行的指令复制到CPU的指令寄存器中。指令的地址被复制并保存在程序计数器中。当复制完毕后，程序计数器自动加1，指向内存中的下一条指令。
- **译码** 当指令置于指令寄存器后，该指令将由控制单元负责译码。指令译码的结果是产生一系列系统可以执行的二进制代码，这是CPU单元能识别的机器码。

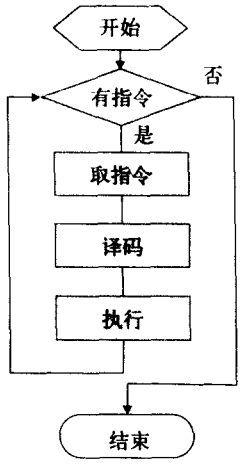

图1-2 机器周期的步骤