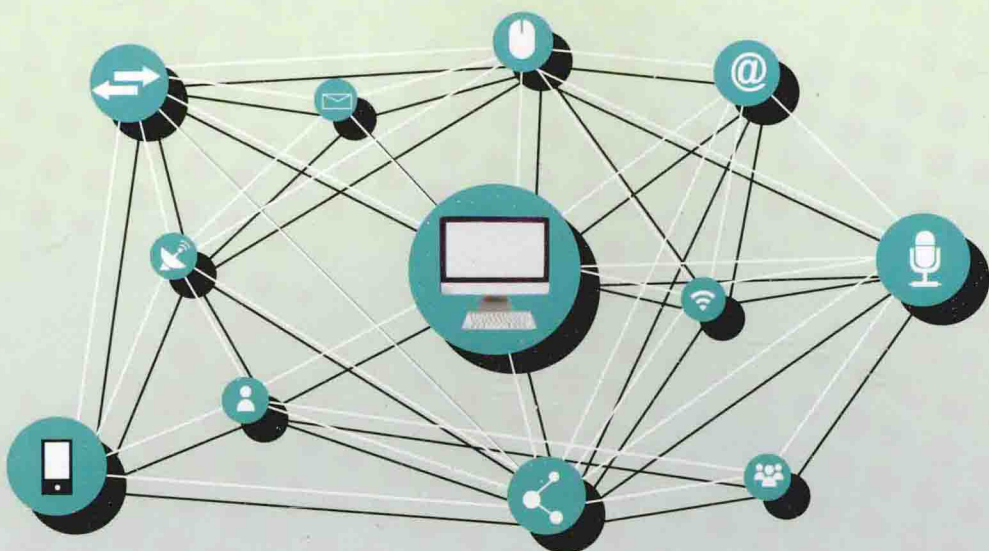




普通高等教育“十二五”应用型本科规划教材·计算机系列
山西省高等学校省级教学成果奖配套教材



程序设计基础

——思想与方法

CHENGXU SHEJI JICHU

中国高等教育学会 组织编写

冯俊 编著

 普通高等教育“十二五”应用型本科规划教材·计算机系列
山西省高等学校省级教学成果奖配套教材



程序设计基础 ——思想与方法

中国高等教育学会 组织编写

冯 俊 编著

中国人民大学出版社
· 北京 ·

图书在版编目(CIP)数据

程序设计基础：思想与方法/中国高等教育学会组织编写；冯俊编著．

—北京：中国人民大学出版社，2014.8

普通高等教育“十二五”应用型本科规划教材·计算机系列．

山西省高等学校省级教学成果奖配套教材

ISBN 978-7-300-19694-7

I. ①程… II. ①中… ②冯… III. ①C语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第179268号

普通高等教育“十二五”应用型本科规划教材·计算机系列

山西省高等学校省级教学成果奖配套教材

程序设计基础——思想与方法

中国高等教育学会 组织编写

冯 俊 编著

Chengxu Sheji Jichu Sixiang Yu Fangfa

出版发行 中国人民大学出版社

社 址 北京中关村大街31号

邮政编码 100080

电 话 010-62511242(总编室)

010-62511770(质管部)

010-82501766(邮购部)

010-62514148(门市部)

010-62515195(发行公司)

010-62515275(盗版举报)

网 址 <http://www.crup.com.cn>

<http://www.ttrnet.com> (人大教研网)

经 销 新华书店

印 刷 北京昌联印刷有限公司

规 格 185mm×260mm 16开本

版 次 2014年8月第1版

印 张 24.25

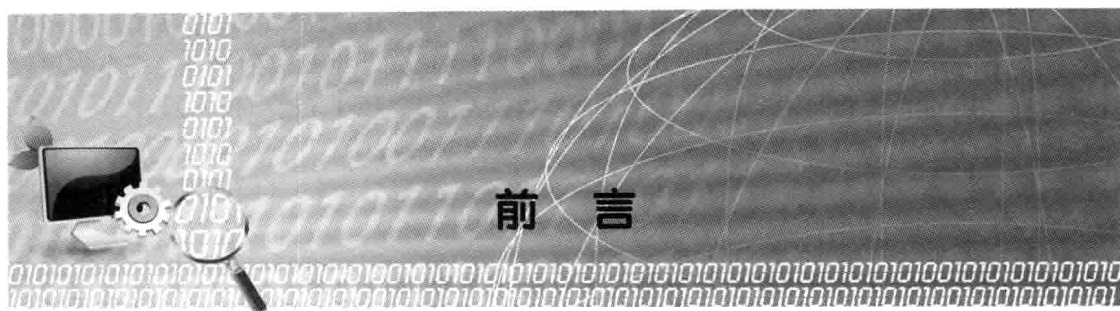
印 次 2014年8月第1次印刷

字 数 570 000

定 价 48.00元

版权所有 侵权必究

印装差错 负责调换



为贯彻落实《国家中长期教育改革和发展规划纲要（2010—2020 年）》，结合“高等学校本科教学质量与教学改革工程”万种新教材建设项目，充分发挥教材在提高人才培养质量中的基础性作用，积极推进高等院校教学改革和教材建设，中国高等教育学会启动了《普通高等教育“十二五”应用型本科规划教材》立项项目。本书是普通高等教育“十二五”应用型本科规划教材，是山西省高等学校省级教学成果奖配套教材。教材充分吸收学科新知识、新技术、新成果，使其具有基础性、理论性、指导性、概括性，注重应用性、实践性，以增强读者计算思维能力与运用计算机解决实际问题的能力。

2005 年 6 月，美国总统信息技术咨询委员会（PITAC）给美国总统提交报告《计算科学：确保美国竞争力》（Computational Science: Ensuring America's Competitiveness）。报告中写道：虽然计算本身也是一门学科，但是其具有促进其他学科发展的作用。21 世纪科学上最重要的、经济上最有前途的研究前沿都有可能通过熟练掌握先进的计算技术和运用计算科学而得到解决。报告建议：将计算科学长期置于国家科学与技术领域的中心领导地位。

美国卡内基·梅隆大学计算机科学系原系主任、时任美国国家科学基金会计算机与信息科学与工程部负责人周以真（Jeannette M. Wing）教授于 2006 年 3 月在美国计算机权威杂志、ACM 会刊《Communications of the ACM》杂志上给出计算思维（Computational Thinking, CT）的定义，计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

20 世纪 80 年代，计算技术开始渗透到大多数学科领域。1985 年春，（美国）计算机协会（Association for Computing Machinery, ACM）与（美国）电气和电子工程师学会计算机分会（Institute of Electrical and Electronics Engineers—Computer Society, IEEE—CS）联手组成攻关组，经过近 4 年工作，提交了《计算作为一门学科》（Computing as a Discipline）的报告。1990 年，攻关组在该报告的基础上提交了计算学科教学计划的 Computing Curricula 1991（CC1991）报告。1998 年秋，ACM 与 IEEE—CS 再次联手组成任务组，经过 3 年多的努力工作，于 2001 年 12 月提交了计算学科教学计划的 Computing Curricula 2001（CC2001）报告。该报告将计算学科划分为 14 个领域，每个领域分成若干知识单元，每个知识单元又包括若干主题。CC2013 又将计算学科扩展到 18 个领域。CC2001 将这些领域的大多数知识单元规定为计算机及相关学科本科学生必须掌握的核心知识。程序设计基础主要涉及程序设计基础（PF）、算法与复杂性（AL）和程序设计语言

(PL) 等领域。掌握这些领域的知识对于利用计算机资源, 开发高效的程序系统是必不可少的。本书内容涵盖或涉及上述 3 个领域中的多个知识单元, 主要包括 PF1 (基本程序设计结构)、PF2 (算法和问题求解)、PF3 (基本数据结构)、PF4 (递归技术)、PF5 (事件驱动程序设计)、AL1 (基本算法分析)、AL2 (算法策略)、AL3 (基本计算算法)、PL1 (程序设计语言概述)、PL3 (语言翻译)、PL4 (数据类型)、PL6 (面向对象程序设计) 和 PL9 (数据抽象) 等。

计算机运行的过程就是程序执行的过程, 运用计算机解决现实世界中的任何实际问题, 最终都要将现实问题转换成计算机程序, 在这一转换过程中, 需要运用多方面的知识进行程序设计, 程序是程序设计的结晶, 程序设计是开发和应用计算机的钥匙。

程序设计基础课程是为高校大多数专业设置的, 长期以来, 关于如何讲授程序设计基础课程, 许多人把争论的焦点放在了语言的选择上, 把讲授的重点放在了语言本身, 却忽略了程序设计真正实用的基本思维方式和方法, 最后导致学生几乎没有分析问题、解决问题的技能。程序设计基础课程教学的核心目标, 应该是让学生学习和掌握对于实际问题, 如何分析问题和设计解决它们的算法; 帮助学生理解程序设计的基本思想和科学原理; 掌握程序设计的基本知识、基本技术和基本方法; 掌握程序设计中的数据组织结构和程序流程控制结构; 培养学生的计算思维能力。为学生能用计算机处理实际问题打下良好基础。

本书在脱离具体语言环境下, 以计算思维为基础, 围绕程序设计的中心问题——如何分析解决实际问题、如何构思算法, 简明扼要地介绍了程序设计中的基本概念、基本思想、基本技术和基本方法。主要内容大体可分为六个组成部分。第 1 章为第一部分, 主要介绍程序设计的基础知识。包括程序设计的基本概念、基本思想、基本方法和基本步骤等。帮助读者理解这些概念, 弄清这些术语之间的关系, 为学习后续内容打好基础。第 2 章为第二部分, 主要介绍 C 语言与 C++ 语言。包括 C 语言的发展与特点、C 语言应用程序结构、C++ 语言集成开发环境、运行应用程序方法与上机操作步骤、从面向过程到面向对象以及 C++ 语言中的核心内容类与对象的运用。方便读者能尽快进入一个良好的语言编程和上机实践环境。第 3 章、第 5 章、第 7 章为第三部分, 主要阐述程序设计中的数据组织。包括基本数据类型、构造数据类型和基本数据结构。帮助读者掌握数据的组织形式, 能够选择合适的数据组织形式解决实际问题。第 4 章与第 6 章的 6.6、6.7 节为第四部分, 主要阐述程序设计中的控制结构。包括三种基本控制结构、函数、子程序与过程文件等内容。帮助读者掌握程序的三种基本控制结构(顺序结构、选择结构、循环结构)与实现过程的封装。在解决实际问题中, 能够设计出具有良好结构的程序。第 6 章的 6.1~6.5 节与第 8 章为第五部分, 主要阐述程序设计中的技术与方法。包括程序设计的技术与方法(模块化设计技术与方法、自顶向下设计技术与方法、逐步求精设计技术与方法、结构程序优化技术与方法等)以及算法设计中的常用方法(包括枚举法、递归与递推、分治法、动态规划法、贪心法、回溯法与分支限界法等), 在这一部分将通过对能付诸实施的、生动有趣的大量应用示例进行讲解, 使读者进一步熟悉前面所学内容, 熟练掌握程序设计的基本思想、基本技术和基本方法。第 9 章为第六部分, 以解决问题为中心。通过几个典型问题, 综合运用前面所学知识, 完成问题分析、算法设计、程序实现与程序运行。使所学知识综合应用到解决实际问题中。

由于 Pascal 语言具有丰富的数据类型和良好的结构, 所以在数据组织描述中, 拟选用

类 Pascal 语言作为工具；为了着重体现算法设计的思想与算法结构，对算法的描述拟选用结构化流程图（N—S 图）作为工具；为了方便读者上机实践，将选用较流行的 C 语言或 VFP 语言对所有算法进行编程实现。既让读者在脱离复杂语言环境下轻松学习程序设计思想和方法，又不至于使他们陷入只有思想的纸上谈兵，这是本书的特色之一。程序设计既是一门实践性很强的带有艺术特性的变换技术，又是一门科学。本书在每一章都开辟了一个课程设计题目，包括问题描述、基本要求、测试数据、实现提示和问题拓展，旨在提高读者分析问题和解决问题的能力，这是本书的又一个特色。

本书条理清楚，内容详实。概念表述严谨，逻辑推理严密，语言精练，算法构思精巧，结构清晰。既注重程序设计思想的介绍，又重视算法设计能力的培养。本书深入浅出，配有大量实例和图示，每一章都有一个课程设计题目和丰富的习题，适合于自学。

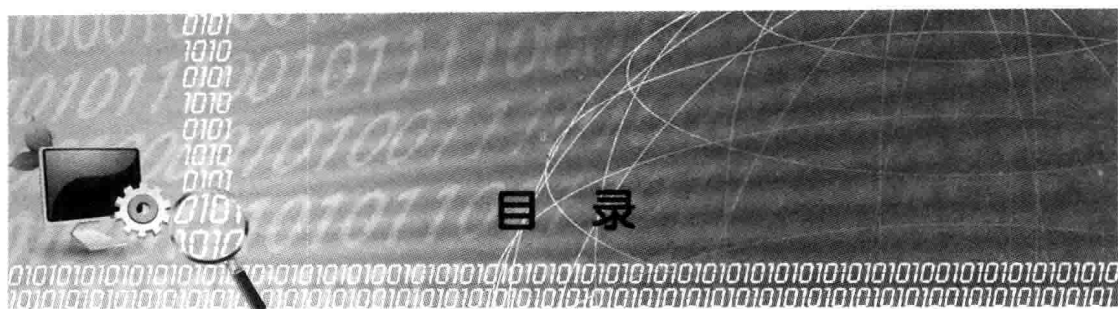
程序设计基础课程在教学计划中至少应为 6 学分，课堂教学在 60~80 学时之间。本课程是一门技术性、实践性很强的课程，为了使学能真正掌握有关理论知识和应用技术，在整个教学过程中至少应安排 5 个以上课程设计，必须保证学生有足够的课下思考作业时间和上机实践时间。上机时数、课下作业时数和课堂讲授时数的比例应不低于 0.5 : 6 : 1。

本书作为山西省普通本科高等教育教学改革研究项目——“《程序设计基础》课程教学内容和体系结构改革研究与实践”以及山西省高等学校省级教学成果奖的组成部分，在教育教学改革研究与课程建设过程中，得到各级领导的支持、专家的指导和同事的帮助。在此表示谢意。

本书凝结了作者 30 多年来的教学科研成果和在讲授《程序设计基础》等课程中的教学经验。在编写过程中，参考了多种优秀教材。在编辑出版过程中，得到了中国人民大学出版社各级领导的支持，负责本书编辑工作的全体同仁付出了辛勤劳动。在此表示衷心感谢。

由于作者水平有限，加之学科理论与技术发展日新月异，书中疏漏谬误之处在所难免，恳请广大读者指正。E-mail: feng1682000@126.com。

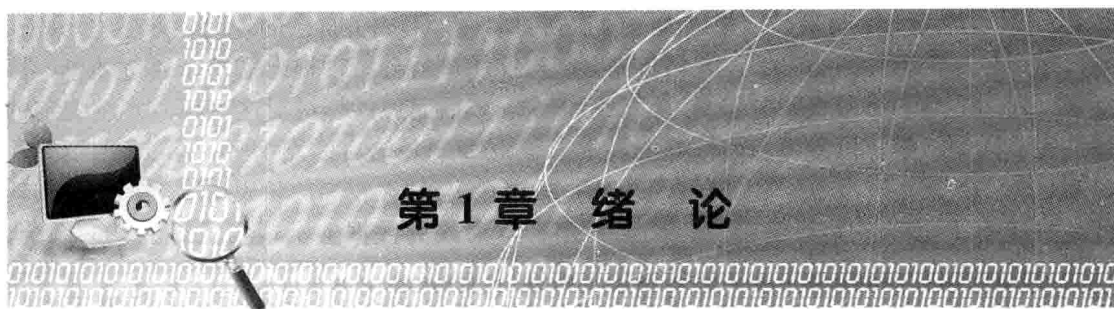
作 者
2014 年 5 月



第1章 绪 论	1
1.1 计算学科	1
1.2 计算思维	3
1.3 程序=数据结构+算法	5
1.4 程序设计=数据结构+算法+程序设计方法	11
1.5 语言工具和环境	15
1.6 程序设计步骤与程序设计风格	22
1.7 算法设计与分析	24
1.8 课程设计相关知识	33
1.9 课程设计题目——求最大公因子	35
习题	36
第2章 C语言与C++语言	37
2.1 C语言的发展与特点	37
2.2 C语言应用程序结构	41
2.3 Visual C++ 6.0 集成开发环境	42
2.4 运行C语言应用程序方法与上机操作步骤	46
2.5 从面向过程到面向对象	60
2.6 C++语言	62
2.7 课程设计题目——类与对象	77
习题	78
第3章 简单数据类型与表达式	79
3.1 数据类型	79
3.2 常量与变量	82
3.3 运算符与表达式	87
3.4 课程设计题目——求最小公倍数	93
习题	94

第4章 程序的基本控制结构	95
4.1 程序的基本控制结构	95
4.2 顺序结构程序设计	98
4.3 选择结构程序设计	101
4.4 循环结构程序设计	111
4.5 课程设计题目——求解方程的根	123
习题	124
第5章 构造数据类型	125
5.1 数组类型	125
5.2 结构体类型	142
5.3 其他构造数据类型	153
5.4 课程设计题目——排序算法	164
习题	165
第6章 结构化程序设计	167
6.1 结构化方法概述	167
6.2 模块化设计技术与方法	168
6.3 自顶向下设计技术与方法	172
6.4 逐步求精设计技术与方法	173
6.5 结构程序优化技术与方法	181
6.6 子程序与过程文件	185
6.7 函数	192
6.8 课程设计题目——学生成绩管理系统	207
习题	207
第7章 基本数据结构	209
7.1 顺序表	209
7.2 链表	220
7.3 栈	235
7.4 递归与非递归过程	242
7.5 队列	250
7.6 二叉树	258
7.7 课程设计题目——一元多项式计算器	275
习题	276
第8章 算法设计中的常用方法	279
8.1 问题的解空间	279

8.2 枚举法	281
8.3 递归与递推	290
8.4 分治法	304
8.5 动态规划法	312
8.6 贪心法	319
8.7 回溯法	325
8.8 分支限界法	333
8.9 课程设计题目——0—1 背包问题	339
习题	340
第 9 章 以解决问题为中心	342
9.1 一元多项式问题	342
9.2 八皇后问题	349
9.3 骑士游历问题	360
9.4 哈夫曼树与哈夫曼编码	364
9.5 课程设计题目——哈夫曼编/译码系统	373
习题	374
参考文献	375



计算思维是当今颇受关注的涉及计算学科本质问题和未来走向的基础性概念；程序、程序设计、数据结构和算法是计算学科中最具有方法论性质的核心概念。计算机运行的过程就是程序执行的过程，运用计算机解决现实世界中的任何实际问题，都需要通过计算思维方式最终将现实问题转换成计算机程序。在这一转换过程中，需要运用多方面知识进行程序设计，程序是程序设计的结晶，程序设计是开发和应用计算机的钥匙。本章介绍计算学科、计算思维、程序设计的基本概念和基础知识。

1.1 计算学科

计算学科是对描述和变换信息的算法过程的系统研究，包括它的理论、分析、设计、有效性、实现和应用等。计算学科是一门交叉学科，它将各个学科的知识经过高度综合，形成一整套有关信息表示、变换、存储、处理、控制和利用的理论、技术与方法。

计算作为一门学科，究竟是科学、是技术、还是工程？计算学科可分为科学型、技术型和工程型 3 类，它包括计算机科学、计算机技术、计算机工程和计算机应用等。科学、技术与工程是现代科学技术中的三个不同领域或不同层次。科学是对客观世界本质规律的探索与认识，发展的主要形态是发现，主要手段是研究，主要成果是学术论文和专著；技术是科学与工程之间的桥梁，发展的主要形态是发明，主要手段是研发，主要成果是专利，也包括论文和专著；工程是科学与技术的应用和归宿，是以创新思维对现实世界出现的新问题进行求解，发展的主要形态是综合集成，主要手段是设计、制造、应用和服务，主要成果是产品、作品、工程实现与产业。计算学科既需要优秀的科学家，又需要发明家和工程师，更需要大量高素质的能够创造性解决实际问题的实用型人才。

1.1.1 计算学科的根本问题

计算学科是研究计算机及其周围各种现象和规律的科学，亦即研究计算机系统结构、程序系统（即软件）、人工智能以及计算本身的性质和问题的学科。计算学科是一门包含各种各样与计算和信息处理相关的系统学科，从抽象的算法分析、形式化语法等，到更具体的主题。计算学科根植于电子工程、数学和语言学，是科学、工程和艺术的结晶。它在

20 世纪最后的三十年间兴起，并成为一门独立的学科，形成了自己的理论、方法与术语。

运用计算机进行问题求解，人们首先必须对问题进行分析，提出需求，通过抽象，建立模型，用适当的符号将问题求解过程表示出来，然后通过计算机对这些符号实施规定的计算完成问题求解。世界著名的计算机科学家、PASCAL 语言的发明者、第 19 位图灵奖（1984 年）获得者 N. 沃思（Niklaus Wirth）教授指出：“在较高的认识层次上，硬件和软件是一样的。”因为它们最终都可以归结为用一定形式的数据表示现实世界的某一系统，并使用算法通过对这些数据的变换获得相应的处理结果。

研究表明，并不是所有问题都是“可计算”的。在 1989 年 1 月的《ACM 通讯》（Communications of the ACM）杂志上刊登的《计算作为一门学科》的报告中，对计算学科中的根本问题作了这样的概括：什么能被（有效地）自动进行？即计算机科学讨论的根本问题是“可行性”的有关内容。或者可以更直接地说，计算学科各分支领域的根本任务就是进行计算，其实质就是字符串的变换。因此，计算学科的根本问题不仅要研究什么可以被有效地进行自动计算，而且还要研究如何进行有效地自动计算。在论述人的计算能力方面，就机械过程的计算而言，著名的美籍华裔数理逻辑学家和计算机科学家王浩教授认为：人要做机械永远不能做的某些计算是不容易的。

1.1.2 计算学科的基本特征

计算机硬件系统的非连续特性，确定了计算机系统对客观对象的描述是离散的，该离散性体现在系统的状态及其变化的刻画上。另一方面，计算学科所关注的计算已经从早期的实例计算迁移到当前以模型为主的计算。根据这一事实，符号、模型及其变换成为计算机系统处理问题的重要特征，使得离散结构和算法在计算学科中占有重要地位。

以符号表示和变换为核心的抽象和形式化是计算学科最强有力的工具。形式化、模型化是基本的描述手段，符号是基本表示形式，符号变换是基本求解方式，抽象思维和逻辑思维是基本思维方式。计算学科的基本特征是抽象表示和自动计算，如图 1—1 所示。

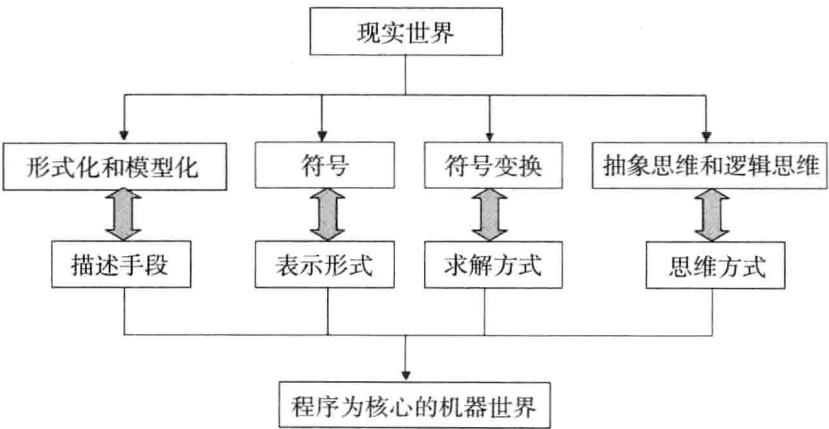


图 1—1 计算学科的基本特征

计算学科具有“抽象、理论、设计”三个学科形态，这表明计算学科同时具有理科与工科特征。计算学科的抽象性、逻辑性和构造性并存，决定了计算学科的教育、科学研究

和项目开发都具有理论与实践紧密结合的特征。科学理论与技术应用相辅相成、相互作用、高度融合是计算学科的突出特点。

计算学科的专业人才应该具备计算思维能力、系统开发和应用能力、算法设计与分析能力、程序设计与实现能力。

1.2 计算思维

科学是反映现实世界各种现象的本质和规律的分科的知识体系。科学一般包含：自然科学、社会科学和思维科学。

科学思维（简称思维）是高级的心理活动形式。是理性认识及认识的过程，即经过感性阶段获得的大量材料，通过整理和改造，形成概念、判断和推理，以反映事物的本质和规律。人脑对信息的处理包括分析、抽象、综合和概括。

科学思维不仅是一切科学研究和技术发展的起点，而且始终贯穿于科学研究和技术发展的全过程，是创新的灵魂。科学思维主要分为理论思维、实验思维和计算思维三大类。

一般认为，理论、实验和计算是推动人类文明进步和科技发展的三大支柱。这种认知被科学文献广泛引用。

理论源于数学，理论思维支撑着所有的学科领域。正如数学一样，定义是理论思维的灵魂，定理和证明则是它的精髓。公理化方法是最重要的理论思维方法，科学界一般认为，公理化方法是世界科学技术革命的源头。用公理化方法构建的理论体系称为公理系统。

理论思维又叫推理思维，以推理和演绎为特征，理论思维以数学学科为代表。

实验思维又叫实证思维，以观察和总结自然规律为特征，实验思维以物理学科为代表。实验思维的先驱是意大利科学家伽利略，他开创了以实验为基础具有严密逻辑理论体系的近代科学，被人们誉为“近代科学之父”。一般来说，伽利略的实验思维方法可以按以下三个步骤进行：（1）提取从现象中获得的直观认识的主要部分，用最简单的数学形式表示出来，以建立量的概念；（2）由此式用数学方法导出另一易于实验证实的数量关系；（3）通过实验证实这种数量关系。

与理论思维不同，实验思维往往需要借助于某些特定的设备，并用它们来获取数据以供分析。以实验为基础的学科有物理、化学、天文学、生物学、医学、农业科学、冶金、机械，以及由此派生的众多学科。

计算思维又叫构造思维，以设计和构造为特征，计算思维以计算学科为代表。

1.2.1 计算思维的概念

2006年3月，美国卡内基·梅隆大学计算机科学系主任周以真（Jeannette M. Wing）教授在美国计算机权威杂志，ACM会刊《Communications of the ACM》杂志上给出计算思维（Computational Thinking, CT）的定义：计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

计算思维是通过约简、嵌入、转化和仿真等方法，把一个困难的问题阐释成如何求解它的思维方法；计算思维是一种递归思维，是一种并行处理方法，是一种把代码译成数据又能把数据译成代码的方法，是一种多维分析推广的类型检查方法；计算思维是采用抽象和分解的方法来控制庞杂的任务或进行巨型复杂系统的设计，是基于关注点分离的方法（SoC 方法）；计算思维是一种选择合适的方式陈述一个问题，或对一个问题的相关方面建模使其易于处理的思维方法；计算思维是按照预防、保护及通过冗余、容错、纠错的方式，并从最坏情况进行系统恢复的一种思维方法；计算思维是利用启发式推理寻求解答，即在不确定情况下的规划、学习和调度的思维方法；计算思维是利用海量数据来加快计算，在时间和空间之间、在处理能力和存储容量之间进行折中的思维方法。

1.2.2 计算思维的 6 个特征

（1）计算思维是概念化的，不是程序化的。计算机科学不只是计算机编程。像计算机科学家那样去思考意味着远远不只能为计算机编程，还要求能够在抽象的多个层次上思考。计算机科学不只是关于计算机，就像音乐产业不只是关于麦克风一样。

（2）计算思维是根本的，不是刻板的技能。计算思维是一种根本技能，是每一个人在现代社会中发挥职能所必须掌握的。刻板的技能意味着简单的机械重复。

（3）计算思维是人的思维，不是计算机的思维。计算思维是人类求解问题的一条途径，但决非要使人类像计算机那样去思考。计算机枯燥且沉闷，人类聪颖且富有想象力。是人类赋予计算机激情。计算机赋予人类强大的计算能力，人类应该好好地利用这种力量去解决各种需要大量计算的问题。

（4）计算思维是思想，不是人造品。不只是将我们生产的软硬件等人造物到处呈现给我们的生活，更重要的是计算的概念，它被人们用来求解问题、管理日常生活，以及与他人进行交流和互动。

（5）计算思维是数学和工程思维的互补与融合。计算机科学在本质上源自数学思维，它的形式化基础建筑于数学之上。计算机科学又从本质上源自工程思维，因为我们建造的是能够与实际世界互动的系统。所以计算思维是数学和工程思维的互补与融合。

（6）计算思维面向所有的人，所有地方。当计算思维真正融入人类活动的整体时，它作为问题求解的有效工具，人人都应当掌握，处处都会被使用。

1.2.3 计算思维的本质

计算思维的本质（Essence）是抽象（Abstraction）和自动化（Automation）。计算思维中的抽象完全超越物理的时空观，并完全用符号来表示，其中，数字抽象只是一类特例。

计算思维中的抽象显得更为丰富，也更为复杂。比如，堆栈（Stack）是计算学科中常见的一种抽象数据类型；算法也是一种抽象；程序也是一种抽象。计算思维中的抽象与其在现实世界中的最终实施有关。

抽象层次是计算思维中的一个重要概念，它使我们可以根据不同的抽象层次，有选择

地忽视某些细节，最终控制系统的复杂性；在分析问题时，计算思维要求我们将注意力集中在感兴趣的抽象层次或其上下层；我们还应当了解各抽象层次之间的关系。

计算思维中的抽象最终要能够机械地一步步自动执行。为了确保机械的自动化，就需要在抽象的过程中进行精确和严格的符号标记和建模。

计算思维不仅仅属于计算机科学家，它应当是每个人的基本技能。在培养人们的解析能力时，不仅要求人们掌握基本的阅读、写作和算术（Reading, writing, and arithmetic, 3R），并且还应该要求人们学会基本的计算思维。

1.3 程序 = 数据结构 + 算法

数据结构与算法是计算学科中研究的基本课题。世界著名的计算机科学家、PASCAL 语言的发明者、第 19 位图灵奖（1984 年）获得者 N. 沃思（Niklaus Wirth）教授曾提出了这样一个有名公式：

$$\text{程序} = \text{数据结构} + \text{算法}$$

它清楚地揭示了计算学科中数据结构与算法这两个概念的重要性和统一性。人们不能离开数据结构去抽象地分析求解问题的算法，也不能脱离算法去孤立地研究程序的数据结构。N. 沃思教授还说，不了解施加于数据上的算法，就无法决定如何构造和组织数据；反之，算法的选择也常常在很大程度上要依赖于作为基础的数据结构。

1.3.1 程序

“程序”一词，从广义上讲可以认为是一种行动方案或工作步骤。这里的程序指的是计算机程序（Program），它实际上表示的是一种处理事务的步骤和顺序。由于组成计算机程序的基本单位是指令，因此，计算机程序就是按照操作步骤事先编制好的、具有特定功能的有限指令序列。

一个计算机程序必须对问题的每个对象和处理规则给出正确详尽的描述。针对问题所要处理的对象，设计合理的数据结构，常常可以有效地简化算法，数据结构与算法是计算机程序的两个重要方面。数据结构是加工处理的对象，一个计算机程序要进行计算或处理总是以某些数据元素为对象，要设计一个好的程序就需要将这些数据按照某种要求组织成一个适合的数据结构。算法是程序的核心，它在程序编制、软件开发，乃至在整个计算机科学中都占有重要地位。程序是算法和数据结构两要素统一的全过程，或者说，程序就是在数据的某种特定表示方式以及存储结构的基础上对抽象算法用某种程序设计语言进行的具体描述（实现）。程序与算法不同，程序不一定满足算法要遵循的准则。例如，操作系统是一个在无限循环中执行的程序，它不是一个算法。

1.3.2 什么是数据结构

数据结构是随着计算机科学和技术的发展而逐渐形成的一门计算机相关专业的核心课

程。当今，计算机应用已渗透到人类社会的各个领域，除了用于科学计算之外，更广泛地用于科学管理等方面。因此，计算机处理的数据量越来越多，数据间的关系越来越复杂，这就要求人们必须研究如何有效地组织数据和处理数据，这正是数据结构要研究的内容。下面通过一个例子说明数据结构在计算机科学和技术中的重要地位。

例 1—1

工厂生产模型与计算机解题模型。

工厂的生产过程可以看成是对原材料的加工处理，最后得到产品的过程。在这个过程中，显然包括两个关键阶段：

(1) 原材料的管理——原材料如何在仓库中进行组织、存储和管理。

(2) 原材料的加工处理——采用什么样的工艺技术、按照什么样的操作顺序对原材料进行加工处理，最后得到合格产品。

工厂生产模型，如图 1—2 所示。

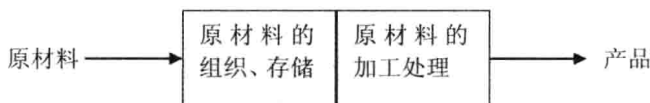


图 1—2 工厂生产模型

由此可见，原材料的组织、存储管理和原材料的加工处理是工厂进行正常生产的关键。

学习计算机科学和技术的目的就是运用计算机来解决实际问题。计算机解题过程也可以看作是对原材料进行组织、存储管理、加工处理，最后得到产品的过程。只不过这里的原材料是数据，产品是处理结果，对数据的加工处理是由算法决定的。数据是对客观事物采用计算机能够识别、存储和处理的形式所进行的描述。对数据组织结构的研究越来越受到重视。

计算机解题模型，如图 1—3 所示。

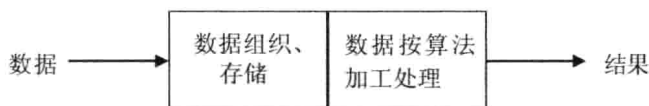


图 1—3 计算机解题模型

由此可见，计算机解题的关键是数据的组织和算法的设计。数据结构就是研究数据组织和算法设计的。

简单地说，**数据结构**（Data Structure）是研究一些数据的集合。就是根据数据的性质、数据元素之间的关系，研究如何表示、存储、操作这些数据的技术。

在**计算学科教学计划 2001**（Computing Curricula 2001, CC2001）的 14 个研究领域中，数据结构和算法的基本内容主要涵盖在**程序设计基础**（Programming Fundamentals, PF）、**算法与复杂性**（Algorithms and Complexity, AL）和**程序设计语言**（Programming Language, PL）3 个领域中。CC2001 强调了算法和程序设计。由此可见，人们越来越重视数据结构和算法，认为程序设计的实质就是对确定的问题，选择一种好的数据结构和设计一个好的算法。因此，《数据结构》课程已不仅仅是计算机相关专业教学计划中的核心

课程之一，而且是非计算机专业的主要选修课程之一。

究竟什么是数据结构？计算机求解问题一般经过这样几个步骤：首先从问题中抽象出一个适当的数学模型，然后设计一个解决数学模型的算法，最后编出程序、调试程序、进行测试、最终得到解答。对于求解数值计算问题，数学模型一般可以用数学方程描述。然而，对于更多的求解非数值计算问题，通常是无法用数学方程描述的，这类问题数据量大、数据间关系复杂，求解的不是某个数值或几个数值，而是要得到某种检索结果、某种排列状态或某种设计表示等，这类问题通常是用一种被称为数据结构的工具来描述数据及其数据之间的关系。下面通过例子来认识数据结构。

 例 1—2

图书馆的书目检索问题。

当你想借阅一本参考书又不知道书库中是否有此书的时候；或者当你想找某一方面的参考书又不知道图书馆内有哪些这方面书的时候，都需要到图书馆去查阅图书目录卡片。在图书馆内有各种名目的卡片：有按书名编排的、有按作者编排的、还有按分类编排的。若利用计算机进行检索，则计算机处理的对象就是这些目录卡片上的书目信息。列在卡片上的一本书的书目信息由登录号、书名、作者名、分类号、出版单位和出版时间等若干数据项组成，每一本书都有唯一登录号，但是不同的书目之间可能有相同的书名、或者有相同的作者名、或者相同的分类号。因此，在书目自动检索系统中，可以建立一个按登录号顺序排列的书目文件和三个分别按书名、作者名和分类号顺序排列的索引表，如图 1—4 所示。这 4 张表就是书目检索问题的数学模型描述。

登录号	书名	作者	分类号	...
1001	高等数学	樊映川	S01	...
1002	理论力学	罗远祥	L01	...
1003	高等数学	华罗庚	S01	...
1004	线性代数	漆汝书	S02	...
...	...	...	...	...

(a) 按登录号排列

书名	登录号
高等数学	1001,1003
理论力学	1002, ...
线性代数	1004, ...
...	...

(b) 按书名排列

作者	登录号
樊映川	1001, ...
罗远祥	1002, ...
华罗庚	1003, ...
漆汝书	1004, ...

(c) 按作者名排列

分类	登录号
S	1001,1003
L	1002, ...
...	...

(d) 按分类号排列

图 1—4 书目文件和索引表示例

这个例子中的问题是非数值计算问题，此数学模型无法用数学公式或数学方程描述，它使用了被称为表的数据结构进行描述。简单地说，数据结构是指数据及其数据之间的关系。对于数据结构这个概念，至今尚未有一个被一致公认的标准定义。不过在讨论任何一种数据结构时，都会自然地联系到这种类型的数据所需要的运算，以及为了在计算机上实现这些运算，如何将这些数据存储到计算机中。因此，在讨论数据结构的时候，一般考虑

- (1) 数据之间的逻辑关系——称为**数据的逻辑结构**。
- (2) 数据在计算机中的存储形式——称为**数据的存储结构**。
- (3) 定义在逻辑结构上的运算及其在存储结构上运算的算法实现——称为**数据的运算**。

设有某单位职工工资表，如表 1—1 所示。

编号	姓名	基本工资	津贴	扣除费	实发工资
1 001	王一华	1 420	1 260	200	2 480
1 002	李静	980	880	180	1 680
1 003	张丽华	1 420	1 260	210	2 470
...	...	...	...	...	...

事实上，这个数据结构可以看做是一个线性表，表中结点与结点之间是一种简单的线性关系，即表中有且仅有一个结点是开始结点，有且仅有一个结点是终端结点，对于表中任一结点，与它相邻且在它前面或后面的结点最多只有一个，这就是数据的逻辑结构。当将这个线性表存入计算机时，是采用顺序存储方式还是采用链接存储方式，这是数据存储结构要解决的问题。至于数据的运算问题，是指对表中的结点进行查找、修改、删除、插入等操作。这些问题都弄明白了，该表的数据结构也就完全清楚了。

因此,对数据的逻辑结构 $DS=(K, R)$ 的存储表示(数据的存储结构)需要考虑以下三方面问题。