

## 第3章 计算思维及其作用体现

**【问题引出】** 人类计算工具的进步、计算环境的演变、计算学科的形成，拓展了人们的思维方式——计算思维。随着计算机科学技术的发展，计算思维正在不断渗透到其他学科领域。那么，什么是计算思维，它具有哪些功能特性，计算思维与计算学科形态有何关系，如何用计算思维来求解问题和进行系统设计，如何培养学生的计算思维能力等，这些都是本章所要学习的内容。

**【教学重点】** 计算思维的基本概念及其本质特征、计算思维与问题求解、计算思维在计算机科学中的作用、计算思维在人才培养中的作用。

**【教学目标】** 掌握计算思维的基本概念；熟悉计算思维的本质与计算机学科形态特征的关系；深刻认识计算思维在计算机学科中的作用体现和计算思维对其他学科的影响。

### 3.1 计算思维及其本质特性

思维（thinking）是人脑对客观事物本质属性和内部规律间接或概括的反映过程，是在表象概念基础上进行分析、综合、判断、推理等认知活动的过程，是人类认识世界的一种高级反映形式。计算思维是人类解决问题或问题求解的一种思想方法，并在其过程中呈现思维本质特性。

#### 3.1.1 人类思维的类别

思维作为一种心理现象，是由一系列知识构成的解决问题的基本思路，是感知的概念化和理性化，也是人类认知思想和智慧的结晶。随着人类认知及其科学探索的深入，不仅概括出人类思维的基本类型和基本特征，还把基于抽象的基本思维引申到科学思维。

##### 1. 思维的基本类型

人类思维有多种类型，按照思维的抽象程度和规律，可以分为感性思维、理性思维、抽象思维、形象思维、灵感思维和系统思维等。

(1) 感性思维（perceptual thinking）是指认识建立在感觉基础上，以意识片段为形式的描述，靠自己的经验和直觉去思考和判断。感性思维是理性思维的基础，理性思维是感性思维的深化。

(2) 理性思维（rational thinking）是指在直观感性的基础上，经过界定概念、客观推理、科学判断后形成的正确反映客观世界的本质和规律的认识过程。理性思维主要是靠已经掌握的科学方法去思考和判断，因而它是形成正确决策的关键，同时也是个人素养的重要组成部分。

(3) 抽象思维（abstract thinking）是指人类特有的一种思维方式，是利用逻辑工具对思维内容进行抽象的思维活动，因而也称为逻辑思维。抽象思维过程得以形式化、规则化和通用化，就是要求创造出与科学相适应的科学逻辑，如形式逻辑、数理逻辑和辩证逻辑等。

(4) 形象思维（vivid thinking）是指综合一切可以利用的素材并加以整理，构筑成一门形象思维的学问。形象思维又称为直觉思维，分为直观动作思维和表象思维两种。直观动作思维

是指思维者能直接影响思维对象，并通过思维者自身的动作去影响思维对象，即思维者与思维对象处于同一环境，现实生活中人们所做的每一件事都离不开这种形式的思维。表象思维是指思维时通过想象对思维对象进行加工改造的思维，思维者与思维对象并不处于同一环境。

(5) 灵感思维 (inspiration thinking) 也称为创造性思维，是指在科学研究中不受或较少受传统思维和范式的束缚，超越常规、构筑新意、独树一帜、捕捉灵感或相信直觉，用于实现科学研究突破的一种思维方式。相比之下，抽象思维是线性的，形象思维是二维的，灵感思维是三维的。

(6) 系统思维 (system thinking) 是指考虑到客体联系的普遍性和整体性，认识主体在认识客体的过程中，将客体视为一个相互联系的系统，以系统的观点来考察研究客体，并主要从系统的各要素之间的联系、系统与环境的相互作用中，综合地考察客体的认识心理过程。

## 2. 思维的基本特征

事物的直接感知过渡到抽象思维的中间环节，语言是表述思维活动的工具。正是基于这种表象和概念，使得人类思维具有如下特征。

(1) 概括性特征 (generality feature) 是指在人的感性基础上，将一类事物的共同、本质的特征和规律提取出来并加以概括。人的感觉和知觉只能反映事物的个别属性，不具备概括性，而思维能反映一类事物的本质和事物之间的规律性联系。例如，通过感觉和知觉，只能感知太阳每天从东方升起，从西方落下，而通过思维能揭示这种现象是地球自转的结果。

(2) 间接性特征 (indirectness feature) 是指思维凭借知识和经验对客观事物进行非直接的、以其他事物为媒体来反映客观事物。例如医生根据医学知识和临床经验，并通过病史询问以及使用医疗设备进行诊断检查，就能判断患者内脏器官的病变情况，从而确定其病因并做出治疗方案。

(3) 能动性特征 (active feature) 是指不仅能认识和反映客观世界，还能对客观世界进行改造。例如，人的肉眼看不到 DNA，但人的思维揭示了 DNA 分子的双螺旋结构，从而揭示了大自然潜藏的遗传密码。又如，人类不仅认识到物体离开地球所需的宇宙速度，还制造出了地球卫星和宇宙飞船飞向太空。

## 3. 科学思维

随着人类认识的不断深化，在探索科学研究方法（理论、实践、计算）过程中逐渐形成了与其相适应的思维方式——科学思维 (scientific thinking)。科学思维是人们在科学探索活动中形成的、符合科学探索活动规律与需要的思维方法及其合理性原则的理论体系。

国科发财〔2008〕197号文件指出“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”。对此，人们站在不同的视角，对科学思维给出了不同的概念性描述。

**描述一** 科学思维是指形成并运用于科学认识活动的、人脑借助信息符号对感性认识材料进行加工处理的方式与途径。一般地，科学思维比人类基础思维更具有严谨性和科学性。

**描述二** 科学思维是指认识自然界、社会和人类意识的本质和客观规律性的思维活动，其思维内涵主要表现为高度的客观性，是围绕求得科学答案而展开的思维并采取理论思维的形式。

**描述三** 科学思维是指理性认识及其过程，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，以便反映事物的本质和规律。

**描述四** 科学思维是指人脑对自然界中事物的本质属性、内在规律及自然界中事物之间的

联系和相互关系所做的有意识的、概括的、间接的和能动的反映，该反映以科学知识和经验为中介，体现为对多变量因果系统的信息加工过程。简而言之，科学思维是人脑对科学信息的加工活动。

虽然上述对科学思维定义的描述不同，但都有其共同的理解，即科学思维是主体对客体理性的、逻辑的、系统的认识过程，是人脑对客观事物能动的和科学的反映。

科学思维之所以有多种不同的描述，是因为科学思维具有多种类型。如果着眼于科学思维的具体手段及其科学求解功能，那么科学思维可分为以下4种。

(1) 发散求解思维是指人们在科学探索中不受思维工具或思维定式的制约，从多方面自由地思考问题答案，其中包括求异思维、形象思维和直觉思维等。

(2) 逻辑解析思维是指人们在科学探索中自觉运用逻辑推理工具去解析问题并由此推得问题解的思维方法，包括类比思维、隐喻思维、归纳思维、演绎思维和数理思维等。

(3) 哲理思辨思维指人们在科学探索中运用不同程度的思辨性哲学思维去寻求问题答案，包括协调思维、系统思维和辩证思维等。

(4) 理论建构与评价思维指人们在科学探索中总结解题成果进而形成和完善理论系统的思维，包括理论形成思维、理论检验思维和理论评价思维等。

如果从人类认识世界和改造世界的思维方式出发，科学思维可以分为理论思维、实证思维和计算思维，这3种思维被称为“科学发现的三大支柱”，对人类进步和文明传承具有巨大的贡献。

(1) 理论思维 (theoretical thinking)，又称为逻辑思维，是通过抽象概括，建立描述事物本质的概念，应用科学的方法探寻概念之间联系的一种思维方法。理论思维以推理和演绎为特征，以数学学科为基础，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，定理和证明是它的精髓，公理化方法是最重要的理论思维方法。

(2) 实证思维 (experimental thinking)，又称为实验思维，是通过观察和实验获取自然规律法则的一种思维方法。实证思维以观察和归纳自然规律为特征，以物理学科为基础。实证思维的先驱是意大利科学家伽利略·伽利莱 (Galileo Galilei, 1564—1642年)，被誉为“近代科学之父”。与理论思维不同，实证思维往往需要借助某种特定的设备，从而获取数据，以便进行分析。

(3) 计算思维 (computational thinking)，又称为构造思维，是从具体算法设计入手，通过算法过程的构造与实施来解决实际问题的思维方法。计算思维以设计和构造为特征，以计算学科为基础，是思维过程的计算模拟方法论，其研究目的是提供适当的方法，使人们借助计算机逐步实现人工智能的目标。诸如模式识别、决策、优化和自动控制等算法都属于计算思维范畴。

#### 4. 思维方式与科学方法的关系

人类文明的发展和科技的进步都离不开思维与科学。人类思维方式可以分为理论思维、实证思维和计算思维，科学方法可以分为理论科学、实验科学和计算科学，思维方式与科学方法具有如下对应关系。

(1) 逻辑思维对应理论科学，以推理和演绎为特征。逻辑思维和实证思维的对应关系是最早明确和建立的。其中，逻辑思维起源于希腊时期，主要科学家有苏格拉底、柏拉图、亚里士多德，他们构建了基本的现代逻辑学体系。逻辑思维符合两个主要原则：一是要有作为推理基础的公理集合，二是要有一个可靠和协调的推理系统。逻辑思维结论的正确性源于公理的正确

性和推理规则的可靠性。为了保证推理结论的可接受程度，要求作为推理基础的公理体系是可证明的。

(2) 实证思维对应实验科学，以观察和归纳自然规律为特征。实证思维起源于物理学的研究，主要科学家有开普勒、伽利略和牛顿。开普勒是现代科学中第一个有意识将自然现象观察总结成规律并表示出来的科学家；伽利略建立了现代实证主义的科学体系，强调通过观察和实验获取自然规律法则；牛顿把观察、归纳和推理完美地结合起来，形成了现代科学的整体框架。

相对于计算思维，实证思维符合 3 个原则：一是可以解释以往的实验现象，二是在逻辑上不能自相矛盾，三是能够预见新的现象，即思维结论经得起实验的验证。

(3) 计算思维对应计算科学，以抽象化和自动化（或以设计和构造）为特征。计算思维是与人类思维活动同步发展的思维模式，是人类思维的重要组成部分。由于计算思维考虑可构造性和可实现性，而相应的手段和工具的研究进展缓慢，因此计算思维概念的明确和建立经历了漫长过程。

### 3.1.2 计算思维的概念

计算思维是人类认识世界和改造世界的一种新的思维方式，也是科学思维的基本方式之一，属于思维科学的一个专门领域，现已成为计算机科学技术问题求解的思想方法。

#### 1. 计算思维的形成

在人类社会进程中，一直进行着认识和理解自然界的活动，并随着社会的发展和进步，其认识水平和理解能力不断提高。古代，人类主要以观察或实验为依据，经验地描述自然现象。后来，人类开始用科学的技术手段对观测到的自然现象加以假设，然后构造其模型，在经过大量实例验证模型的基础上，对新的自然现象用模型进行解释和预测。在工业社会，人们解决问题的思维方式是首先了解事物的物理特性，然后思考如何用原料生成新事物。由此可见，人类对自然的认识和理解经历了经验的、理论的和计算的 3 个阶段，但不论哪个阶段，人类认识、理解甚至改造自然所必备的能力是由解决问题的思维方式、技术手段、科学方法所决定的，并与社会环境密切相关。

20 世纪中期，随着计算机的出现和计算机科学的发展，派生出了基于计算的研究方法，人们通过数据采集、数据处理、结果分析与统计，然后用计算机来辅助分析复杂现象。

进入信息社会后，人们解决问题的思维方式是利用现代计算工具进行数据处理和构想新的数据模型，需要进行数据抽象、数据处理以及计算机科学理论的支持。这种解决问题的思维方式就是“计算思维”，它是人类思维与计算工具能力的综合。

#### 2. 计算思维的定义

计算思维的概念是 1996 年由美国麻省理工学院(Massachusetts Institute of Technology, MIT)的西蒙·派帕特(Seymour Papert)教授提出的。由于计算思维对科学发展具有举足轻重的作用，因而引发了许多学者的研究探索。2006 年 3 月，美国卡内基-梅隆大学(Carnegie Mellon University, CMU)计算机科学系主任周以真(Jeannette M. Wing)教授给出了计算思维的定义：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。这一定义揭示了计算思维概念的核心，指出了计算思维的思想观点和方法。计算思维的提出引发了全球计算机教育界的广泛关注和积极探讨，已成为当今

国际计算机界对计算机科学本质以及未来发展走向的研究热点。对于计算思维的研究和探索，可以概括为以下 3 个方面。

(1) 问题求解中的计算思维。问题求解是科学研究的基本目的之一，利用计算机求解的问题既可以是数据处理、数值分析等问题，也可以是求解物理、化学、经济学、社会学等提出的问题。例如，求解物理问题，使用计算机求解的过程可分为 4 步：首先，把实际的应用问题转换为数学问题，即把问题转换为偏微分方程 (Partial Differential Equations, PDE)，并将 PDE 离散为代数方程组；其次，建立数据模型，且评估模型是否可解；然后，设计算法和编程实现；最后，在计算机中由程序实行自动运行求解。前两步是计算思维中的抽象，后两步是计算思维中的自动化。

(2) 系统设计中的计算思维。任何自然系统和社会系统都可视为一个动态演化系统，演化伴随着物质、能量和信息的交换，这种交换可以映射为符号变换，使之能用计算机实现离散的符号处理。当动态演化系统抽象为离散符号系统后，就可以采用形式化的规范来描述，通过建立模型、设计算法和开发软件来揭示演化的规律，实时控制系统的演化并自动执行。

(3) 人类行为理解中的计算思维。计算思维是基于可计算的手段，以量化的方式进行的思维过程，是能满足信息时代新的社会动力学和人类动力学要求的思维方法。在人类的物理世界、精神世界和人工世界中，计算思维是建设人工世界所需要的主要思维方式。

计算思维的研究包含狭义计算思维和广义计算思维两个方面：立足计算机科学本身，研究该学科中涉及的构造性思维称为狭义计算思维，即研究怎么把问题的求解过程映射成计算机程序的方法；借助计算机科学概念进行问题求解、系统设计、人类行为理解等涵盖所有人类活动（包括反过来指导计算机科学）的一系列思维活动则被称为广义计算思维，它是狭义计算思维的延伸和拓展。

### 3.1.3 计算思维的本质

当人们利用计算机处理或求解一个具体的实际问题时，其思维过程通常按以下步骤进行。

(1) 分析问题。在求解问题时，首先是分析问题，理解求解问题的目的，建立正确的数据模型，并确定利用计算机求解时需要提供哪些输入信息，需要输出哪些信息等。

(2) 制订计划。根据问题性质，选择适合的算法，制订求解的可行性方案。在此过程中，要考虑如何充分发挥计算机高速运算和计算机按照程序自动执行的优势。

(3) 执行计划。计算机按照程序步骤和所提供的输入数据进行计算，然后输出运行结果。

(4) 检验结果。检验和分析程序运行结果是否正确，如何改进和提高。

从上述步骤可以看出，利用计算机求解问题是一个人-机结合的联合方式，既要发挥人的特长——抽象，又要发挥计算机的特长——自动化，可以将其概括为“两 A”——抽象 (abstraction) 和自动化 (automation)，它既是利用计算机求解问题的思维方法，也是计算思维的本质。

#### 1. 抽象

在计算机科学中，抽象是一种被广泛使用的思维方法，也是利用计算机求解问题的第一步。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。与数学和物理科学相比，计算思维中的抽象更为广泛。数学抽象时抛开现实事物的物理、化学和生物学等特性，仅保留其量的关系和空间的形式。而计算思维中的抽象除了具有数学抽象的特点，还要确定合适的抽象对象和选择合适的抽象方法，并考虑如何实现的问题。例如，文件是对输入输出设备的抽象；

虚拟内存是对程序存储器的抽象；进程是对一个正在运行的程序的抽象；虚拟机是对整个计算机的抽象。

现实世界的事物及其规律是一个多层次的系统，因此，揭示自然规律的抽象必然也是一个多层次的系统。在问题抽象过程中，通常分为以下3个层次。

第一层抽象——计算理论。它是信息处理机的抽象，信息处理机的工作特性是映射，即把一种信息变换成另一种信息。本层所关注的问题是计算的本质是什么？什么是可计算的？计算机的基本能力和局限性是什么？

第二层抽象——信息处理。它是信息表示与算法的抽象，因而涉及输入、输出信息的选择，以及把一种信息变换成另一种信息的算法选择。本层所关注的问题是能否计算和计算的复杂度。

第三层抽象——硬件系统。它是信息表示和算法的物理实现的抽象，因而涉及计算机硬件的功能特性。其关注的问题是，物理上如何实现信息表示和算法复杂度，是否理论上不可计算。

从概念上讲，抽象的计算理论与现实的机器硬件之间没有直接关系，但算法是计算理论与机器硬件有关的中介。从现实上看，3个层次相互关联，任何问题的计算都是相互协调的结果。

## 2. 自动化

在计算思维中，“抽象”对应“建模”，“自动化”对应“模拟”，抽象是手段，自动化是目的。也就是说，计算思维中的抽象，最终要能够按照程序一步步地自动执行。这一过程是一种映射，即通过计算机语言，把客观世界的实体（问题空间对象）映射成计算机中的实体（解空间对象）。

抽象和自动化的目的是能够一步步地自动执行抽象形成的数据模型，以求解问题、设计系统和理解人类行为。从实施角度讲，计算思维的本质可以从多个层面体现，通常可划分为3个层面。

(1) 机器层面：协议（抽象）和编码器/转换器等（自动化），解决机器与机器之间的交互问题。

(2) 人机层面：语言（抽象）和译码器/执行器（自动化），解决人与机器之间的交互问题。

(3) 业务层面：模型（抽象）和引擎/执行系统（自动化），解决业务与计算之间的交互问题。

抽象层次是计算思维中一个极为重要的概念，使人们可以根据不同的抽象层次，进而有选择地忽视某些细节，最终控制系统的复杂性。为了确保能机械地自动化，就需要在抽象过程中进行精确、严格的符号标记和建模，同时要求软件系统提供各种不同抽象层次之间的翻译工具。

### 3.1.4 计算思维的特性

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、人类行为理解等涵盖计算机科学的一系列思维活动，并且在思维活动中体现计算思维的概念特性和计算思维的问题特性。

#### 1. 计算思维的概念特性

计算思维是人类的基本思维方式，从方法论的角度讲，计算思维方式体现出7个概念特性。

(1) 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方法和途径，但绝非试图要让人像计算机那样思考。计算机之所以能求解问题，是因为人类将计算思维的思

想赋予了计算机。例如，用计算机求解方程是将求解思想赋予计算机后，它才能进行求解计算。

(2) 计算思维是概念化思维，不是程序化的思维。计算思维像计算机科学家那样，在抽象的多个层次上思考问题，它远远超出了计算机编程，计算机科学不等于计算机编程。

(3) 计算思维是数学和工程相互融合的思维，而不是数学性的思维。计算机科学本质上源自数学思维，其形式化基础是构建于数学之上，但因为受计算设备的限制迫使计算机科学家必须进行工程思考。数学思维和工程思维的相互融合，体现抽象、理论和设计的学科形态。

(4) 计算思维是思想，而不是物品。计算思维突显问题方法和计算概念，被人们用来求解问题、管理日常生活、与他人交流和活动。例如，计算机能逻辑推理，它是人类智慧的结晶。

(5) 计算思维是一种基础技能，而不是机械技能。计算思维是现代社会中每个人都必须掌握的，刻板的技能意味着机械地重复，但计算思维不是这类机械重复的技能，而是一种创新的能力。

(6) 计算思维是一种理念，而不是表现形式。计算思维是一种引导计算机教育家、研究者和实践者的前沿理念，并且面向所有人和所有领域，能融入人类的各种活动中，而不是停留和表现在形式上。计算思维是解决问题的有效工具，在所有学科、所有专业中都能得到应用。

(7) 计算思维是一种思维方法，而不是一种思维模式。计算思维可以由人或计算机执行，例如递归、迭代，黎曼积分，人和机器都可以计算，但人的计算速度无法与计算机相比，借助于计算机的“超算”能力，人类就能够用智慧去解决那些在计算机时代之前不敢尝试的问题，实现“只有想不到，没有做不到”的境界。

## 2. 计算思维的问题特性

计算思维通常表现为人们在问题求解、系统设计以及人类行为理解的过程中，对抽象、算法、数据及其组织、程序、自动化等概念和方法潜意识地应用，周以真教授将其概括为 7 个问题方法。

(1) 计算思维利用化繁为简、化难为易，通过约简、嵌入、转化和仿真等方法，把一个看来困难的问题重新阐释成一个我们知道怎样解决问题的思维方式。

(2) 计算思维利用递归思维、并行处理，既能把代码译成数据，又能把数据译成代码，是一种多维分析推广的类型检查方法。

(3) 计算思维是采用抽象和分解来控制庞杂的任务，或进行巨大复杂系统设计的方法，因而是一种基于关注点分离 (Separation of Concerns, SoC) 的方法。

(4) 计算思维是选择合适的方式去陈述一个问题，或对一个问题的相关方面进行建模，使其易于处理的思维方法。

(5) 计算思维是按照预防、保护以及通过冗余、容错和纠错的方式，从最坏情况进行系统恢复的一种思维方法。

(6) 计算思维是利用启发式推理寻求解答，在不确定情况下的规划、学习和调度的思维方法。

(7) 计算思维是利用海量数据加快计算，在时间和空间之间、处理能力和存储容量之间进行权衡折中的思维方法。

正是计算思维的这些概念特性和问题特性，使我们在问题求解和系统设计过程中，突破传统的思维方式，从而使一些复杂问题迎刃而解，计算机学科就是在问题求解的实践中发展起来的。

## 3.2 计算思维的问题求解

在计算机科学发展过程中，人们一直在不断探索问题求解的基本策略和思维方式，计算思维就是实行问题求解最为有效的思维方式。问题求解涉及数学建模、基本策略和过程抽象。

### 3.2.1 问题求解的数学建模

数学建模 (mathematical modeling) 源于计算机科学技术的飞速发展，大量的实际问题需要用计算机来解决，而计算机与实际问题之间需用数学模型 (mathematical model) 来沟通，在此背景下数学建模应运而生。数学建模是 20 世纪 70 年代初诞生的一门新兴学科，是一种用数学仿真 (mathematical simulation) 或数学模拟 (mathematical analogue) 方法建立的数学模型。具体来说，数学建模是针对参照某种事物系统的特征或数量依存关系，用数学符号对实际问题本质属性的抽象、刻画和定义，概括地或近似地表述出系统的数据结构。因此，数学建模方式是一种数学的思维方式，数学建模过程是一种计算思维的过程，即从实际问题中抽象、提炼、简化出数学模型的过程。

#### 1. 模型的基本类型

模型是对研究对象通过抽象、归纳、演绎、类比等方法，用适当形式描述的简洁表达方式。不同类型问题抽象形成的模型具有不同的形式，适应计算机求解的模型可概括为以下 4 种形式。

(1) 计算模型 (computational model) 是指定量或定性地描述系统各变量之间的相互关系或因果关系的模型。计算模型可以是一个数学公式或一组代数方程、微分方程、积分方程、统计学方程，通常用数学语言描述问题的算法特征，或用符号、图形、表格等形式描述问题的结构特征。

(2) 仿真模型 (simulation model) 泛指实物仿真模型或数字仿真模型，这里是指对现实系统经过抽象和简化所形成的系统模型，属于数字仿真模型。通过对系统模型进行仿真计算，分析仿真数据及其图形，可以预测实际系统基本性能的优劣，从而对系统进行改进和参数优化。

(3) 结构模型 (structural model) 是指反映系统的结构特点和因果关系的模型，结构模型大部分采用抽象简单的图形表示，是研究复杂系统的有效手段。计算机科学领域中的结构模型有图灵机模型、数据库概念模型、网络协议层次模型、云计算模型等。

(4) 思维模型 (thinking model) 是指用简单易懂的语言、图形、表格、符号等形式来表达人们思考和解决问题的形式，例如数据库系统中的概念模型和逻辑模型属于思维模型。思维模型具有不同的结构形式，如判断结构、推理结构、证明结构等，可以把思维模型看作解决实际问题的一种思想策略。

不同的现实世界，具有不同的计算模型。面对不同的计算模型，需要采用不同的求解方法。

#### 2. 建立数学模型的方法、步骤

建立数学模型是为了便于利用计算机来实现问题求解。建立数学模型是实现问题求解中十分关键的一步，也是十分困难的一步。在建立数学模型时，通常要经历以下 4 个步骤。

(1) 模型描述 (model description)。根据问题背景和对象特征，研究变化规律，确定总变量及其相互关系，进而用数学符号 (形式) 语言进行描述，即把实际问题转化为数学问题。

(2) 模型构想 (model conception)。根据对象的特征和建模目的，用数学思维方法提出假设

和数学抽象 (mathematical abstraction) 简化问题, 用形式化语言刻画各变量和常量之间的数学关系, 例如用  $s=r^2\times\pi$  描述圆面积  $s$  与圆的半径  $r$  之间的函数关系。其中,  $s$ 、 $r$  为变量,  $\pi$  为常量。

(3) 模型建立 (model building)。利用数学工具把数学问题转化为一个代数问题, 并且把问题分支归结到解一个方程式, 建立问题整体的数学结构, 形成能为算法描述提供依据的数学模型。

(4) 模型验证 (model verification)。对所建立的模型进行合理性、准确性、可靠性、适应性分析, 然后选取与所建模型有关的数据进行验证, 判别模型的优劣。在验证过程中, 不仅要充分考虑到现实问题的边界和极端情况, 还要考虑数值求解的误差性和稳定性问题。

【提示】数学建模涉及多方面的数学知识, 如计算方法、微分方程、模糊数学、数理统计、图论、最优化方法等。数学建模应具备分析综合能力、抽象概括能力、想象洞察能力、运用数学工具的能力、验证数学模型的能力等, 3.2.3 节中例 3-2 是确定性数值求解的数学建模典型实例。

### 3.2.2 问题求解的基本策略

现实世界中的计算问题往往复杂多样, 要实现问题求解, 必须具有适应各类问题求解的基本方法。问题求解策略是指利用计算机求解问题的基本方法步骤, 可以概括为以下 7 个方面。

#### 1. 构建模型

构建模型 (construct model) 就是建立所求问题的数学模型, 也称为数学建模。它是问题求解的第一步, 也是最为重要的一步, 数学建模的本质是挖掘数据之间的关系和数据的变化规律, 这些“规律”往往隐藏在数据之间而难以发现。所以在数学建模时, 如果能够在繁杂的数据中找到有价值的规律并加以合理应用, 往往可使问题获得简化, 便于利用计算机实现求解。

#### 2. 问题界定

在构建模型和问题求解过程中, 必须理解问题对象的性质、目的、要求等, 只有准确地界定了问题的本质特征, 才能采取解决问题的有效措施、方法和步骤; 否则, 就可能劳而无获, 甚至南辕北辙。因此, 对问题界定 (problem definition) 是极为关键的。然而, 许多问题并非“一目了然”, 在问题界定过程中一定要认真观察、分析、思考、研究问题的实质, 不要被问题的表象所迷惑。

计算机科学中所处理的问题可分为两类: 一类是求解问题; 另一类是求证问题。求解问题的目的是获得问题的准确结果; 求证问题的目的是去判定某一个结论是正确的或是错误的, 即去证明它或去否定它。计算机科学中的证明大多为构造性证明, 详细内容已在 2.3.2 节中讨论过。

#### 3. 寻找条件

在界定问题性质后, 尽力寻找解决问题的必要条件, 以缩小问题求解范围。当遇到一道难题时, 以“简化问题, 变难为易”为原则, 尝试从最简单的特殊情况入手, 找出有助于简化问题、变难为易的条件, 逐渐深入, 最终分析、归纳解题的一般规律。在一些需要进行搜索求解的问题中, 可以采用深度优先搜索和广度优先搜索, 如果问题的搜索范围太大, 减少搜索量最有效的手段是“剪枝” (删除一些对结果没有影响的问题分支), 即建立一些限制条件, 缩小搜索的范围。

**【例 3-1】** 若公鸡每只 3 元，母鸡每只 5 元，小鸡三只 1 元，求 100 元钱买 100 只鸡有多少种方案？这是著名的“百钱买百鸡”问题，是一个典型的不定式方程求解问题。

**【解析】** 用不定式方程求解，设公鸡为  $x$ ，母鸡为  $y$ ，小鸡为  $z$ ，可列出如下联立方程：

$$x + y + z = 100$$

$$3x + 5y + z/3 = 100$$

虽然两个方程式不可能解出 3 个确定的未知数，但利用计算机的高速运算对上述有限集合中  $1 \leq x \leq 33$ 、 $1 \leq y \leq 20$ 、 $3 \leq z < 100$  的  $x$ 、 $y$ 、 $z$  的各种组合值进行试算（称为枚举算法），只要结果符合两个表达式的值为 100，就记录为一种方案。然而，实际工作中遇到的问题往往错综复杂，可以尝试从多个侧面分析、寻找必要条件，根据各部分的本质特征，利用各方面的必要条件予以综合分析和处理。

#### 4. 问题分解

在解决复杂问题的过程中，常常将复杂问题进行分解，其分解的基本策略是采用“等价”和“分治”方法将系统分解成多个模块，在求得各模块解后进行综合，从而使复杂问题能简单实现。

(1) 利用“等价”关系进行分解。就是把一个复杂系统看作简单问题的集合，因而可以按“等价关系”把一个大系统分解为若干子系统。分解的原则是各子系统具有某种共同属性，使合并后仍保持原有属性状态。有些问题通过分解处理，还能使原系统的性能得到进一步提高。例如数据库中的模式分解，不仅消除了数据冗余和操作异常，还能实现更高一级范式规范。

(2) 利用“分治”思想进行分解。就是把一个复杂的问题分解成若干简单的问题，然后逐个解决，这种分而治之的思想在计算机科学技术中得到广泛应用。例如，算法研究中的“分治算法”和“动态规划”，程序设计中的“模块化”和“子程序”等，就是有效解决复杂问题的常用方法。有些问题通过分治处理，能使系统的效率得到提高，例如利用子程序，能提高编程和执行效率。

#### 5. 对象离散

电子数字计算机是一个离散型系统，它所处理的对象必须是离散型或离散过的数字化数据。因此，在应用中必须将连续型的信息对象离散化。将连续型问题离散化的方法有以下 4 种。

(1) 模拟信息离散化是指将随时间变化而变化的物理量（如血压、气温、体温、电流等）转换为数字量，离散方法是通过模-数转换，将模拟信息转换为数字化信息。

(2) 媒体信息离散化是指将图像、视频、音频等数据信息，通过采样（sample）、量化（quantify）和编码（encode），将多媒体信息转换为数字化信息。

(3) 被积函数离散化是指将那些找不到被积函数或函数表达式的复杂问题，利用定积分近似计算方法得到原问题的解析表达式，通过编制程序，便能利用计算机获得计算结果。

(4) 微分方程离散化是指将描述连续系统的微分方程，通过数值方法（如 Runge-Kutta 法），把微分方程的分析解变为微分方程的数值解。微分方程离散化是计算机数字仿真的核心内容。

#### 6. 确定算法

求解一个具体问题时必须确定算法（determine algorithm）。通常有多种算法可供选择，选择

的标准是算法的正确性、可靠性、简单性和复杂性。然后分析算法，确定算法的复杂度，选择算法的描述方法。

(1) 选择算法。在实行问题求解时需要观察问题的性质和结构，选择相适应的算法。常用算法有枚举法、递推法、迭代法、递归法、分治法、回溯法、贪心法、动态规划等，都有适宜的问题对象，如动态规划适宜解决的问题需要有最优子结构和重复子问题。一旦观察出问题的性质和结构，就可以利用现有的算法去解决它。计算机中涉及的常用算法及其应用在第 7 章中介绍。

(2) 分析算法。获得了求解问题的算法并不等于问题可解，问题是否可解还取决于算法的复杂性，即算法所需要的时间和空间在数量级上能否被接受。2.4 节中介绍了计算学科的典型问题，对于难以计算的复杂问题可采用“等价”“剪枝”“分治”等方法进行简化或近似处理。

(3) 描述算法。算法的描述形式有数学模型、数据表格、结构图形、伪代码、程序流程图等。通过算法描述，规划问题求解和自动执行的步骤。问题求解的程序方法在第 6 章和第 7 章讨论。

### 7. 程序设计

从构建模型到确定算法，这一切工作都是为了利用计算机实行问题求解的手段，而程序设计(programming)则是实现问题求解的目标，也是计算思维本质(抽象与自动化)的真实体现。计算机的本质是程序的机器，所有需要求解的问题都要编成程序，通过机器的高速运算获得结果。

## 3.2.3 问题求解的过程抽象

计算思维是人类思维与计算机功能的综合，因此在问题求解的过程中，既体现出人类求解问题的过程抽象，又体现出计算机求解问题的过程抽象，即体现计算思维的本质——抽象和自动化。

### 1. 人类求解问题的过程抽象

问题求解是一个非常复杂的思维活动过程，是对客观世界的抽象，人的思维越丰富，问题求解越完美。可以将问题求解的思维活动过程概括为 5 个步骤(可视层次阶段)，如图 3-1 所示。



图 3-1 人类求解问题的过程抽象

(1) 发现问题。这是一切研究的起源，人类的许多重大发明和创造都是从发现问题开始的，只有发现了有价值的问题，才会研究它，直到解决问题，而发现问题比解决问题更重要。那么，怎样发现问题，能否发现和提出重大的、有社会价值的问题，取决于多方面的因素。

① 依赖于各人对事件的态度。人对事件的态度越认真，积极性越高，责任感越强，则越容易从现象中捕捉到有价值的问题，这个问题或许是被其他人忽略的“问题”。

② 依赖于思维活动的积极性。不善于思考、观察和缺乏创新意识的人很难发现问题，只有勤于思考、留心观察和善于钻研的人，才能从细微而平凡的事件或活动中发现有重要意义的“问题”。

③ 依赖于求知欲和兴趣爱好。有好奇心和求知欲强烈、兴趣爱好广泛的人往往不满足现实，

而是热心探索和思索，因而往往能从通常事件或活动中发现一些重要“规律”或“奥秘”。

④ 依赖于已有的知识和经验。具有丰富阅历和渊博知识的人容易引起由此及彼、由表及里的分析和类比，由此发现和提出深刻而有价值的“问题”。

(2) 明确问题。在发现问题的基础上，进行问题分析→找出主要矛盾→抓住问题关键→确定问题范围→明确解决问题方案。明确问题是进行问题抽象和构想数学模型的过程，也是一个非常复杂的过程。它不仅与一个人的知识和经验有关，而且与一个人的思维方式和潜在智能有关。

(3) 提出假设。假设的提出依赖于一定的条件，例如已有的知识经验、直观的感性材料、尝试性的实际操作、语言的表达和重复、创造性构想等，都对其产生重要影响。

(4) 检验假设。对所提出的假设是否切实可行，需要进一步检验。检验的方法主要有两种：一种是直接检验法，按照假设条件进行实验，再依据实验结果判断假设的真伪；另一种是间接检验法，根据个人掌握的科学知识，通过智力活动进行论证检验，但最终仍需接受实践的检验。

(5) 问题实施。根据检验假设，找出解决问题的方法、原则、途径，制订解决问题的实施细节并予以实现。这一系列步骤和实施细节是确保问题正确求解的有效措施。

**【例 3-2】** 卫星发射时，要使卫星进入轨道，火箭所需的速度是多少？

**【解析】** 卫星发射是一个极为复杂的过程，按照问题求解策略和计算思维，求解步骤如下。

(1) 发现问题。把研制好的卫星发射到太空中，并使卫星能正常运行，便是要解决的问题。

(2) 明确问题。火箭是一个复杂系统，为了简化问题，这里只从动力系统及整体结构上分析。

(3) 提出假设。为了简化问题，需要对问题进行界定，即把问题线性化，因而进行如下假设：

① 地球是固定于空间中的一个均匀的球体，其质量集中于球心（根据地球为均匀球体假设）。

② 其他星球对卫星的引力忽略不计，并且火箭的引擎足够大。

③ 卫星轨道是过地球中心某一平面上的圆，卫星在轨道上以地球引力作为向心力绕地球做平面圆周运动。卫星绕地球的运行轨迹如图 3-2 所示。

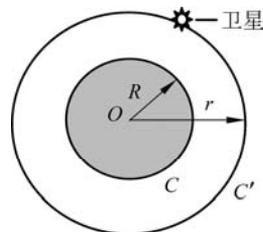


图 3-2 卫星绕地球的运行轨迹

设地球半径为  $R$ ，中心点是  $O$ ，曲线  $C$  为地区表面，卫星轨道为  $C'$ ，轨道半径为  $r$ ，卫星质量为  $m$ 。根据牛顿定律，地球对卫星的引力为

$$F=Gm/r^2 \quad (3-1)$$

其中  $G$  为引力常数，可根据卫星在地面的重量计算：

$$Gm/R^2=mg, \quad G=gR^2 \quad (3-2)$$

将式 (3-2) 代入式 (3-1) 得

$$F=mg \cdot (R/r)^2 \quad (3-3)$$

由假设③，卫星所受到的引力（做匀速运动的向心力），则有

$$F=mv^2/r \quad (3-4)$$

从而得出火箭所需的速度为

$$v=R \cdot (g/r)^{1/2} \quad (3-5)$$

(4) 检验模型。取  $g=9.8\text{m/s}^2$ ,  $R=6400\text{km}$ , 可算出卫星离地面高度分别为 100km、200km、400km、600km、800km 和 1000km 时, 其速度应分别为 7.86km/s、7.80km/s、7.69km/s、7.58km/s、7.47km/s 及 7.371km/s。

(5) 问题实施。在完成上述步骤, 理论分析和实践检测一切正常无误的情况下, 方可实施, 即通过数学计算, 求出最终结果。

【提示】 虽然本例不能概括所有问题求解的思维过程, 但足以说明问题求解中的计算思维方法和步骤, 即人类分析问题和解决问题的思维方法和步骤都是建立在过程抽象之上的。

### 2. 计算机求解问题的过程抽象

现实世界中的计算问题是由实体及实体间的相互关系构成的, 人们把现实世界中的实体称为问题空间或问题域对象, 把计算机中的实体称为解空间对象。为了实现求解自动化, 必须把问题域的对象转换成便于计算机求解的解空间对象, 计算机求解问题的过程抽象如图 3-3 所示。

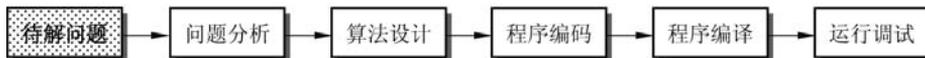


图 3-3 计算机求解问题的过程抽象

- (1) 问题分析: 问题求解的第一步是分析、理解、抽象和归纳, 以获取求解问题的思路。
- (2) 算法设计: 根据问题求解的需要组织数据, 建立起相应的数学模型或数据结构。
- (3) 程序编码: 根据求解问题的数学模型或数据结构, 利用程序设计语言进行编程。
- (4) 程序编译: 用高级语言编写的源程序必须通过编译, 形成扩展名为 EXE 的可执行文件。
- (5) 运行调试: 运行可执行程序, 此时程序设计者必须对运行结果进行检测, 不仅需要验证运行结果是否正确, 而且需要测试各种极端情况下的运行情况。

【例 3-3】 已知五边形的边及对角线的长度, 要求按照程序设计方法, 编写求如图 3-4 所示五边形面积的源程序。

【解析】 参照问题求解的基本策略和问题求解的思维过程, 求五边形面积的步骤如下。

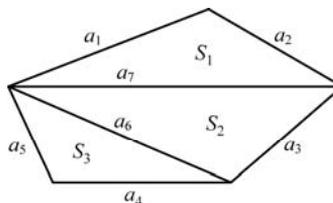


图 3-4 求五边形的面积

(1) 问题分析。求如图 3-4 所示五边形的面积, 实际上就是求出 3 个三角形  $S_1$ 、 $S_2$ 、 $S_3$  的面积之和。为了编写程序简便、高效起见, 可将计算三角形面积定义成函数, 然后在主程序中通过 3 次调用, 再进行相加, 便可得到五边形的面积。

(2) 算法设计。对于数值求解, 必须建立数学模型。对于本例, 通常可分为 3 步。第一步是获取数学模型。求三角形面积的常用计算公式有多种, 最常见的有以下几种。

- ① 已知三角形底  $a$ 、高  $h$ , 则  $S = a \times h / 2$ 。
- ② 已知三角形三边分别为  $a$ 、 $b$ 、 $c$ , 则  $s = (a + b + c) / 2$ ,  $S = \sqrt{s(s - a)(s - b)(s - c)}$ 。
- ③ 已知三角形两边  $a$ 、 $b$  和这两边夹角  $\varphi$ , 则  $S = \frac{1}{2}ab\sin\varphi$ 。
- ④ 设三角形三边分别为  $a$ 、 $b$ 、 $c$ , 内切圆半径为  $r$ , 则  $S = (a + b + c) \frac{r}{2}$ 。
- ⑤ 设三角形三边分别为  $a$ 、 $b$ 、 $c$ , 外接圆半径为  $R$ , 则  $S = ab \frac{c}{4R}$ 。

根据给出的参数, 可以选择公式② (即海伦公式)。只要输入三角形的 3 条边长, 就能计算

并输出该三角形的面积。

第二步是设计算法框架。针对如图 3-4 所示的五边形，设计求出五边形的面积的具体计算方法，直到能用具体的程序设计语言表达为止。为此，根据问题分析，设计计算步骤如下：

- ① 输入  $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ 。
- ② 计算  $S=ts(a_1, a_2, a_7)+ts(a_3, a_6, a_7)+ts(a_4, a_5, a_6)$ 。
- ③ 输出  $S$ 。

第三步是算法设计求精。对各计算步骤进行细化求精，即用算法的表示方法按照问题求解过程进行算法描述。

对算法设计框架的第①步算法求精：input( $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ )；

对算法设计框架的第②步算法求精：设三角形的 3 边长为  $a, b, c$ ，则求三角形面积 area。

Step1: 输入三条边的边长  $a, b, c$ 。

Step2: 计算  $p=(a+b+c)/2$ 。

Step3: 计算  $area = \sqrt{p(p-a)(p-b)(p-c)}$ 。

Step4: 输出 area 的值。

(3) 程序编码。设计好一个算法后，利用描述工具，准确、清楚地将所设计的解题步骤进行描述，然后编写成程序代码。编辑源程序就是按照程序设计语言的语法规则编辑程序代码，编辑源程序的过程就是通过计算机提供的编辑工具将程序代码生成文本文件的过程，通过编辑工具形成的程序代码称为源代码 (source code)。用 C 语言描述的求五边形面积的源代码如下：

```
#include<stdio.h>
#include<math.h>
float ts(float a, float b, float c);          //函数调用声明
main()
{ float a1, a2, a3, a4, a5, a6, a7, s;
  printf("请输入五边形的 7 条边长: \n");
  scanf("%f, %f, %f, %f, %f, %f, %f", &a1, &a2, &a3, &a4, &a5, &a6, &a7);
  s=ts(a1, a2, a7)+ts(a3, a6, a7)+ts(a4+a5+a6); //调用 3 次求三角形面积函数,相加得出总面积
  printf("五边形面积 s=%f\n",s);
}
{ float ts(float a, float b, float c)      //求三角形面积的函数
  float p, area;
  p=(a+b+c)/2;                             //数据必须满足 a+b>c 且 b+c>a、a+c>b 的条件
  area=sqrt(p*(p-a)*(p-b)*(p-c));         //求三角形面积函数公式
  return area;
}
```

(4) 程序编译。无论用何种高级语言编写的源程序，都必须编译成扩展名为 EXE 的文件，计算机才能执行。现在的编译程序大多是将源程序编辑和编译集中在一个环境中，通过编译程序对源代码进行调试编译后，便形成可执行程序 (文件)。运行该程序，便可求出任意五边形的面积。

(5) 运行调试。运行可执行文件便可得出计算结果，程序设计者必须对运行结果进行验证、检测和调试。对于本例来说，按照上述计算步骤，只要输入三角形的 3 条边长，就能计算并输出该三角形的面积。然而，该算法虽然正确，但没有考虑到如果输入的 3 个数满足  $a+b \leq c$  且  $b+c \leq a, a+c \leq b$  时，不能构成一个三角形的情况，因而程序缺乏健壮性。为此，可将该算法改

进如下。

Step1: 输入三条边的边长  $a$ 、 $b$ 、 $c$ 。

Step2: 如果  $a+b>c$  且  $b+c>a$  且  $a+c>b$ , 那么执行 Step3; 否则, 输出提示信息“数据输入不合理”并结束程序。

Step3: 计算  $p=(a+b+c)/2$ 。

Step4: 计算  $area = \sqrt{p(p-a)(p-b)(p-c)}$ 。

Step5: 输出  $area$  的值。

这样便考虑到了输入不合理数据的处理, 满足了对算法健壮性的要求。此外, 还需考察各种极端情况下的运行情况。例如,  $z=y/x$ , 当  $y$  和  $x$  均取实型数时求出  $z$  的值是正确的。但如果类型设置不当, 出现  $x=0$  或  $x$  趋向于 0 就无法求出  $z$  的值, 这说明程序中存在漏洞, 需采取防护措施。

### 3.3 计算思维与计算机学科

计算思维的核心是求解问题、设计系统和理解人类行为, 计算思维的理论基础是计算科学。虽然计算思维本身并不是计算机的专属, 即使没有计算机, 计算思维也会逐步发展, 但计算机的出现给计算思维的发展带来了根本性的变化, 使得两者相互促进, 从而加速了彼此的发展进程。

#### 3.3.1 计算思维本质与学科形态的关系

计算思维之所以与计算学科相互促进, 其关键就体现在计算思维本质(抽象、自动化)与计算学科形态(抽象、理论、自动化)的内在关联。那么, 它们两者之间有何关联? 为了描述方便起见, 这里以关系数据库应用系统设计为例, 揭示计算学科形态与计算思维的关系, 数据库应用系统设计的学科形态如图 3-5 所示。

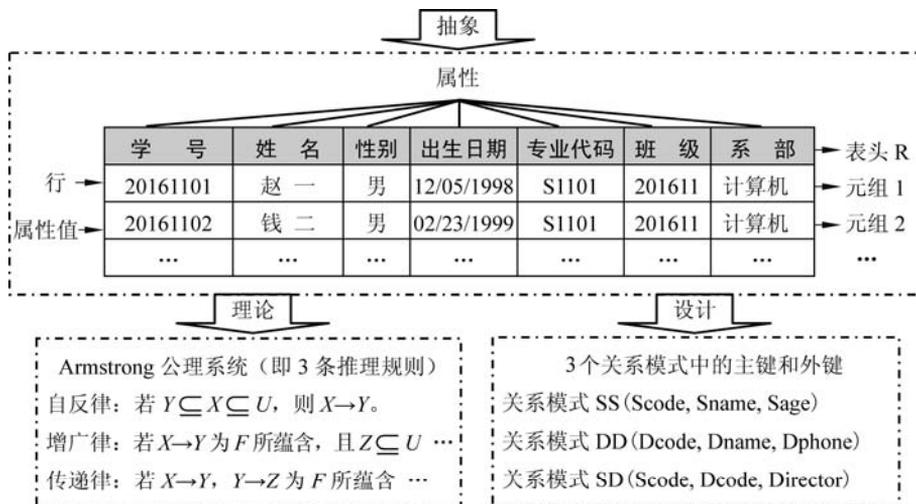


图 3-5 数据库应用系统设计的学科形态

## 1. 抽象

抽象是指从具体事物中发现其本质特征和方法的过程，是感性认识世界的手段，是人们认识世界和解释世界的方法论。例如，利用计算机实行教学信息管理，通常要经过以下步骤。

第一步，录取学生的基本信息。描述一个学生的基本信息，可以包括姓名、性别、出生日期、所学专业、班级、系部。然而，这些信息项仅适合某一个学生，一个班级中常常有同名同姓的情况，如果设置“学号”，便有了信息的“唯一性”。

第二步，以表格形式记录与教学管理没有直接关联的学生信息，可以包括学号、姓名、性别、出生日期、所学专业、班级、系部。

第三步，为了便于对数据记录的操作和科学管理，必须研究“学生信息表”的基本结构及其数据信息的基本特征。例如，把每个学生的信息称为“记录”，把每条记录称为“行”，把行中的各项称为“属性”，把属性的集合称为“表头”。

第四步，在上述基础上进行提炼和概括，形成描述学生基本特征的结构——二维表。

由此可见，在计算机科学中，可以把抽象的过程概括为发现→提取→命名→表达。

(1) 发现 (find) 是指对客观事物进行观察和分析，从共性中找差异，从差异中找共性，从中发现一些内在联系和规律。例如，图 3-5 中的“学号”就是通过对学生基本信息分析的结果。

(2) 提取 (draw) 是指对所观察或待研究的事物中的要素进行区分，图 3-5 中的每行数据是实行学生信息管理必备的信息，每列数据是具有相同类型的数据，是必备关联信息的最小集合。

(3) 命名 (name) 是指对每个需要区分的要素进行恰当的命名，以反映区分的结果。它体现出抽象是“现象事物的概念化”，图 3-5 中将行命名为“元组”，将列命名为“属性”等。

(4) 表达 (express) 是指以适当的形式呈现“提取”和“命名”的要素及其之间的关系，形成“抽象”的结果，如对学生基本信息抽象的结果是如图 3-5 所示的二维表。通过该二维表，可以考查学生信息的基本规律。如果将其抽象结果用数学形式进行表达，便可为理论研究提供数学依据。

## 2. 理论

理论是客观世界在人类意识中的反映和用于改造现实的知识系统。理论需要论证，例如，算法、编程语言、编译系统都要验证其正确性、有效性、安全性、完备性、复杂性等，而所有这些验证，都需要理论的支撑。因此，可以把理论的基本特征概括为定义→公理→定理→证明。

(1) 定义 (definition) 是指对概念的严密化描述。例如，埃德加·考特 (E. F. Codd) 定义了“域”的概念，域是指具有相同类型值的集合。然后定义了“元组”的概念，元组是二维关系表中的“行”，继而定义“笛卡儿积”为所有可能的元组，并且用数学集合与关系概念来定义二维数据表。

(2) 公理 (axiom) 是指依据理性的不证自明的基本事实，经过长期反复实践不需要再证明的基本命题。例如，数据库函数依赖中的阿姆斯特朗 (Armstrong) 公理系统就是著名的推理规则。

(3) 定理 (theorem) 是指经过逻辑推理证明为真的陈述，其中证明的依据是定义、公理或其他已被证明的定理，包括条件和结论两部分，即在条件成立的情况下有结论为真的事实。

(4) 证明 (proof) 是指通过数学理论对所提出的定理或公理进行验证，是由公理和定理推导出某些命题真假的过程，证明有很多种数学方法，如穷举法、反证法、归纳法、构造法等。

### 3. 设计

计算机科学中的设计是构建一个应用系统的过程，是技术、原理在计算机系统中实现的过程。尽管设计的形态和内容多种多样，但其基本特征可以概括为形式→构造→自动化。

(1) 形式 (form) 是指被研究对象的呈现形式。例如，开发一个数据库应用系统，它呈现的是数据模式，并且必须以符号化的形式表达才便于编程；如果是数字仿真，必须给出适合编程的数学模型。

(2) 构造 (structure) 包括“构”和“造”，其中“构”是指被研究对象各种要素之间的组合关系与框架；“造”是指实现各种要素之间的组合关系与框架的过程。计算学科中包括算法构造、抽象层面构造和操作对象构造，例如关系数据模式中的主键和外键便是操作对象构造。

(3) 自动化 (automatization) 是指通过软件、硬件、网络等自动化系统，让机器按照程序指令自动执行。

〔提示〕学科形态是问题求解的基本过程，也是学科研究的价值体现。对计算学科而言，设计是构造计算系统改造世界的手段，如果没有设计，任何系统都难以实现——设计价值；理论是发现世界规律的手段，如果没有理论指导，则无法保证设计的严密性、可靠性、正确性——理论价值；抽象是感性认识世界的手段，也是理论和设计的前提，如果没有抽象，理论和设计都无法达成目标——抽象价值。因此，学科形态体现了学科研究的抽象价值、理论价值和设计价值。

#### 4. 计算学科形态与计算思维本质的比较

计算学科的 3 个形态特征比较如下。抽象 (发现→提取→命名→表达)、理论 (定义→公理→定理→证明)、设计 (形式→构造→自动化) 与计算思维的本质 (抽象与自动化) 两者之间具有交织关系，共同点都是基于“抽象”；学科形态中的“理论”与“设计”实际上是实行“自动化”的手段和步骤。因此，计算学科形态与计算思维本质的目标是一致的，其宗旨是“寻找通用的方法，处理类似的问题”，在表现形式上是“采用不同的形式，解决同样的问题”。

### 3.3.2 计算思维在计算机学科中的体现

从上面的论述可知，计算思维涵盖了反映计算机科学的一系列思维活动，它把一个看起来求解困难的问题阐述成我们知道怎样解决的问题。计算思维在计算机科学中的体现是全方位的，下面从 7 个方面描述计算机科学中的计算思维，并分别体现在后述各章的教学内容中。

#### 1. “1 和 0”的计算思维

计算机科学是研究计算过程的科学，计算过程是通过操作数字符号变换信息或状态的过程，在其变换的过程中涉及语义层面的改变。具体来说，就是用计算机处理现实世界中的各类问题时，必须将现实世界的事物转换为适合计算机处理的表示形式，这就是所谓 1 和 0 的计算思维。

用 1 和 0 来描述事物状态的思维，在计算机学科中不仅有着极为重要的作用，而且是计算机科学理论的“基石”。例如，1 和 0 在电子器件中可以用来表示电位的“有”和“无”；在开关电路中可以用来表示开关的“闭”和“开”；在数据编码中可以用来表示运算符号的-和+；在逻辑学中可以用来表示事件的“真”和“假”等。通过 1 和 0 的有机组合，便能完整地描述事物的特征或现实世界中的自然现象。例如，1 和 0 的组合，在指令系统中可以表示“地址”“操作码”“操作数”；中国古代用两仪（—表示 1，--表示 0）构成了八卦图，并利用八卦图来推演自然规律（春、夏、秋、冬）和自然现象（阴、晴、圆、缺）等。

正是由于“1和0”的计算思维，从而演化形成了计算机科学中数据表示形式化、语义符号化、信息集合化、模型构造化的计算思维，因而使得“1和0”的计算思维成为计算机科学的理论基础。审视这些思想，对深度理解计算思维以及利用计算思维去改造客观世界有着极为重要的意义。

【提示】语义符号化是人类社会发展的标志，并且社会越进步，符号越丰富，含义越深刻。符号具有时代性、社会性、权威性、抽象性等特征，被用来标记事物特征和揭示自然科学规律。

## 2. “计算系统”的计算思维

计算系统是对计算机的抽象，是由硬件和软件组成且能高效工作的完整系统。而计算系统的计算思维是指如何用系统科学的理论和观点来构建一个能实现科学计算和数据处理的计算系统。

【例3-4】根据冯·诺依曼结构计算机的原理，描述构建一个完整的计算系统的计算思维。

这里涉及两个方面的问题：一是计算系统的结构组成；二是如何实现科学计算或数据处理。一个完整的计算系统包括硬件系统和软件系统，硬件系统是计算系统的物理支撑，它由内存存储器、处理器、输入输出设备、外存储器等组成；软件系统由编辑软件、翻译软件、操作系统、其他各类软件组成。其中，操作系统为用户操作使用计算机提供支撑，并实行对内存存储器、处理器、输入输出设备、磁盘文件进行管理。计算系统实现数值计算或信息处理的过程实际上就是把现实世界转换为计算机世界的过程。首先将现实世界抽象成适合计算系统处理的数据模型（即数学建模）；然后进行程序设计，包括算法设计、程序设计方法（面向过程或面向对象）、程序设计语言（面向过程或面向对象）、编制程序（用程序语言编写算法实现的步骤）、程序翻译（将语言程序翻译形成由0和1组成的可执行程序），然后送入硬件系统，进行数值计算或数据处理。构建一个完整计算系统的计算思维如图3-6所示。

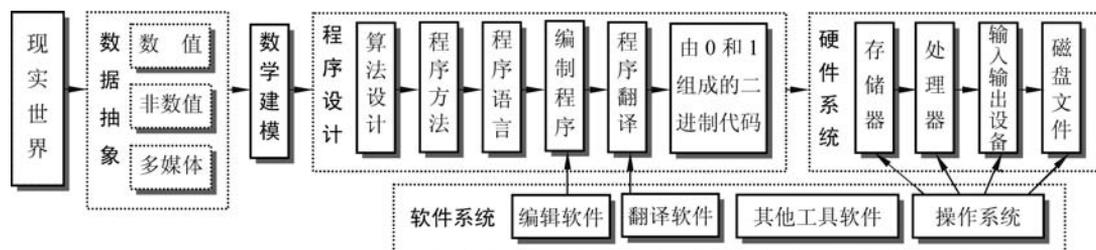


图3-6 构建一个完整计算系统的计算思维

【提示】现实世界所反映出的数据类型可以概括为数值数据、非数值数据和多媒体数据等，不同的数据类型构建出不同的数学模型形式。对于多媒体数据的处理，既可以通过硬件来实现，也可以使用多媒体软件（Audition、Photoshop、Premiere）进行声音、图像、视频编辑、处理和播放等。

## 3. “问题求解”的计算思维

问题求解是计算思维的核心，既体现出人类求解问题的过程抽象，又体现出计算机求解问题的过程抽象。抽象形成数学模型，编程实现自动执行，即体现出计算思维的本质——抽象与自动化。

【例3-5】根据计算系统的工作过程，描述利用计算系统实现问题求解的计算思维。

利用计算机实现问题求解的基本步骤可以概括为问题分析、数学建模、算法设计、程序实

现，整个过程可形成“反馈”结构。利用计算机实行问题求解的计算思维如图 3-7 所示。

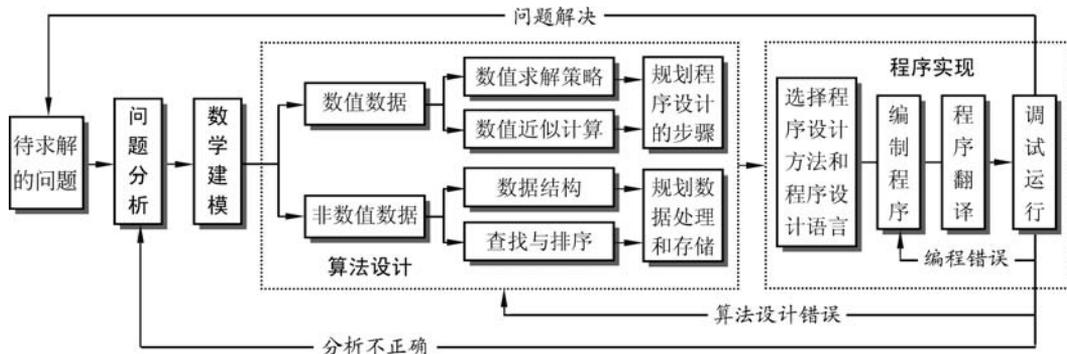


图 3-7 问题求解的计算思维

【提示】 在建立数据模型时要考虑问题求解的可能性和有效性，因而需要选择最有效的算法，还要反复测试，验证算法的适应性和算法的正确性。详细内容在第 7 章中介绍。

### 4. “程序设计”的计算思维

计算机程序是求解问题的“流程”或“顺序”，程序设计思想在计算机科学中是一种极其重要的计算思维，按照 Neumann 原理，计算机解题流程的一般过程如图 3-8 所示。

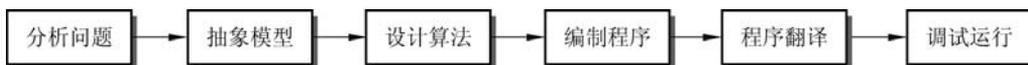


图 3-8 计算机解题流程的一般过程

【例 3-6】 以计算  $a+|b|$  为例，描述利用程序设计实现问题求解的计算思维。

为了使计算机能快速求解问题，首先进行问题分析，把要解决的问题抽象成数学模型，确定问题求解步骤（算法设计）；然后按照处理步骤（算法）编成程序，使计算机把复杂的控制机制变得有序可循；最后进行程序编译、调试和运行，以实现自动、高速地完成解题任务。

(1) 分析问题。针对给出的实际问题，理解未知量、数据类型、数据条件、边界条件等。

(2) 抽象模型。根据问题分析，将其抽象为便于编程计算的数学模型。对于计算  $a+|b|$ ，其数学模型为

$$a+|b| = \begin{cases} a+b, & b \geq 0 \\ a-b, & b < 0 \end{cases}$$

(3) 设计算法。根据给出的数学模型确定计算步骤，并且用程序流程图描述，如图 3-9 所示。该流程是实现数值计算或数据处理“自动化”的根本依据所在。

(4) 编写程序。根据算法流程图用计算机语言编写问题求解的算法程序，它是对被处理数据和算法过程进行详细描述。

对于  $a+|b|$ ，当  $b \geq 0$  时， $a+|b|=a+b$ ；当  $b < 0$  时， $a+|b|=a-b$ 。如果用 C 语言编写算法程序，则为

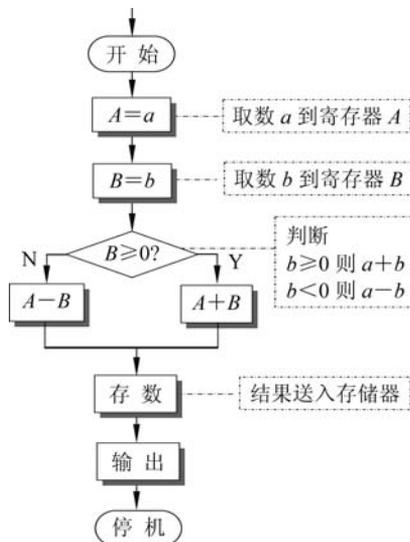


图 3-9 程序设计解题过程流程图

```
#include<stdio.h>
main()
```

```

{ float a, b, c;
  scanf("%f, %f", &a, &b);
  if(b>=0)
    c=a+b;
  else
    c=a-b;
  printf("%d\\", c);
}

```

(5) 程序编译。用程序设计语言编写的程序称为源程序，源程序实际上是各种符号的集合，必须通过编译程序，将其翻译成由 0 和 1 组成的代码程序计算机才能识别。如果所编写的源程序没有语法错误，则通过编译形成可执行程序；如果源程序存在语法错误，那么修改后再编译。

(6) 调试运行。运行可执行程序，便能显示运算结果。然而，由于算法设计考虑不周或数据类型定义等问题，会出现错误结果，此时需要反复测试和调试，其调试过程与图 3-7 所示一致。

〔提示〕 可以把例 3-6 看作例 3-5 的一个实例。无论是问题求解还是程序设计，两者都涉及问题求解的可行性和算法的复杂性分析，详细内容在第 6 章和第 7 章中介绍。

### 5. “递归算法”的计算思维

递归 (recursion) 是一种用有限的步骤描述实现近似无限工作的方法，是计算机学科领域中一种重要的计算思维模式，既是抽象表达的一种手段，也是问题求解的重要方法。

【例 3-7】 利用“和尚讲故事”典型实例，描述运用“递归”实现问题求解的计算思维。

“从前有座山，山上有座庙，庙里有个老和尚，正在给小和尚讲故事！故事讲什么呢？（从前有座山，山上有座庙，庙里有个老和尚，正在给小和尚讲故事！故事讲什么呢？（从前有座山，山上有座庙，庙里有个老和尚，正在给小和尚讲故事！故事讲什么呢？（从前……））”。

这里，反复讲故事可以看成反复调动自身，其关键是如何停下来，否则就没有任何意义了。递归算法求解是借助于数学上的递推方法，根据特定法则或公式对一个或多个前面的元素进行运算而得到后续元素，以确定一系列元素的方法。求阶乘的算法可以定义成以下 3 种函数形式：

$$n! = \begin{cases} 1 \\ n \times (n-1) \times (n-2) \times \dots \times 1 \end{cases} \quad \text{或} \quad n! = \begin{cases} 1 \\ n \times (n-1)! \end{cases} \quad \text{或} \quad f(n) = \begin{cases} 1 & n \leq 1 \\ n \times f(n-1) & n > 1 \end{cases}$$

当  $n \leq 1$  时， $f(n)=1$ ，是递归结束条件；当  $n > 1$  时， $n \times f(n-1)$  是对递归形式的完整描述。

〔提示〕 与递归算法相近似的是迭代算法，它们是问题求解中一种极为重要的计算思维，在计算科学中有着极为重要的地位，关于迭代和递归的算法思想及其编程方法在第 7 章中介绍。

### 6. “信息管理”的计算思维

在当今信息时代，人类社会的数据规模和种类正以前所未有的速度增长，传统的手工管理方式已无法实现对大规模数据的有效管理和高效利用。因此，适应大规模复杂数据信息管理的数据库技术应运而生。为了实现数据库信息管理，首先是对现实世界中涉及规模数据管理和处理的复杂性问题进行调查分析，得到较为准确、细致的用户需求；然后将其抽象为信息世界的概念模型，再将概念模型转换为逻辑（记录）模型，由此确定数据库的物理结构，实现数据库信息管理。

【例 3-8】 根据数据库原理和应用系统开发技术，描述实现计算机信息管理的计算思维。

数据库技术可以解决在大数据的规模效应下的数据存储、管理、分析以及知识发现等问题。