

面向计算思维的程序设计系列教材

Visual Basic 项目教程

主编 薛红梅 张永强



科学出版社

面向计算思维的程序设计系列教材

Visual Basic 项目教程

主 编 薛红梅 张永强

副主编 申艳光 刘志敏 王彬丽

孔外平 马丽艳 王瑞林

科学出版社

北 京

内 容 简 介

当前高等学校计算机基础教学的发展方向是将计算思维作为程序设计课程的主线,培养学生如何像计算机科学家一样思考并解决问题的能力。

本书通过把计算思维的要素、方法融入问题和项目,采用“项目目标—项目分析—项目实施—知识进阶—项目交流”教学五部曲的项目化教学模式,用项目引领教学内容,集成基于项目学习和探究式学习的一体化主动学习方法,强调了理论与实践相结合,以实际应用为目标,突出了对学生基本技能、实际操作能力、工程师职业能力和计算思维能力的培养。

本书共分7章,涉及认识计算思维、程序与算法、程序与数据以及软件工程的基本知识、Visual Basic 集成开发环境与基本概念、程序设计基础、用户界面设计、菜单与工具栏、文件、图像操作、数据库应用等内容。

本书可作为大中专院校教材及各类计算机技术培训教材,并可供不同层次的 Visual Basic 程序设计人员学习和参考。

图书在版编目(CIP)数据

Visual Basic 项目教程/薛红梅,张永强主编. —北京:科学出版社,2015.2

面向计算思维的程序设计系列教材

ISBN 978-7-03-043239-1

I. ①V… II. ①薛… ②张… III. ①BASIC 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 022577 号

责任编辑:于海云 / 责任校对:袁省省
责任印制:霍兵 / 封面设计:迷底书装

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

三河市骏杰印刷有限公司 印制

科学出版社发行 各地新华书店经销

*

2015年2月第一版 开本:16开(787×1092)

2015年2月第一次印刷 印张:14

字数:331 000

定价:30.00元

(如有印装质量问题,我社负责调换)

前 言

程序设计是对学生进行思维训练的一个最直接、最具操作性的平台。程序设计课程对高校大学生来说不仅是职业技能的培养，也体现着创造性思维的信息素质培养过程，重视引导学生实现问题求解思维方式的转换——培养学生的计算思维能力。

CDIO (Conceive-Design-Implement-Operate, 构思-设计-实施-操作/运行) 改革, 以一体化和实用方式回应了工程教育的历史和未来的挑战, 使学生知道如何在现代团队环境下构思、设计、实施及运行复杂且具有高附加值的工程产品、过程和系统。自 2000 年起, 世界范围内以 MIT 为首的几十所大学操作实施了 CDIO 模式, 迄今已取得显著成效, 深受学生欢迎, 得到了产业界的高度评价。

建设符合我国实际需求的适应大工程理念和 CDIO 工程新的教育模式的教材体系是本书改革的核心内容之一, 这将有助于课程体系和教学内容更加合理和科学, 有助于学生以主动的、实践的方式学习和获取工程能力, 包括个人的科学和技术知识、终身学习能力、交流和团队工作能力, 以及在社会及企业环境下建造产品和系统的能力。

CDIO 大纲的第二部分为个人职业技能和特质。大纲中指出, 工程师应该具备的三种思维模式是工程思维、科学思维、系统思维。其中科学思维包括三种: 以观察和归纳自然规律为特征的实证思维; 以推理和演绎为特征的逻辑思维; 以抽象化和自动化为特征的计算思维。因此, 计算思维的培养将大大有利于提高工程师的科学思维能力, 符合 CDIO 理念的要求。

计算思维的概念最早是 2006 年 3 月由美国卡内基·梅隆大学周以真 (Wing) 教授在 *Communication of the ACM* 上发表并定义的。她指出, 计算思维是每个人的基本技能, 不仅仅属于计算机科学家。每个人在培养解析能力时不仅应掌握阅读、写作和算术, 还要学会计算思维。著名的计算机科学家——1972 年图灵奖得主 Dijkstra 说: “我们所使用的工具影响着我们的思维方式和思维习惯, 从而也将深刻地影响着我们的思维能力。”

无论计算机教育工作者, 还是计算机普通用户在学习和使用计算机的过程中, 应该着眼于“悟”和“融”: 感悟和凝练计算科学思维模式, 并将其融入可持续发展的计算机应用中, 这是作为工程人才不可或缺的基于信息技术的行为能力。

计算思维是计算机和软件工程学科的灵魂, 程序设计课程的教学重点不在于培养学生如何解决某些实际问题的能力, 而在于培养学生全面思考问题、分析问题的思维模式, 提高学生动手和动脑的实践能力, 促进学生主动学习和探究意识的培养, 提升学生的创新能力和综合素质。让学生了解和掌握如何充分利用计算机技术, 对现实世界中的问题进行抽象和形式化, 达到人类求解问题的目的。

基于以上教育理念, 本书特色如下。

(1) 融入 CDIO 理念, 采用新的教学五部曲。

本书采用以项目实例“学生管理系统”为导向的教学模式, 集成基于项目学习和探究式学习的一体化主动学习方法。采用“项目目标—项目分析—项目实现—知识进阶—项目交流”教学五部曲的项目化教学模式, 用项目引领教学内容, 强调了理论与实践相结合, 突出对学生基本技能、实际操作能力及工程师职业能力的培养, 符合学生思维的构建方式。通过项目设计激发学生的学习兴趣, 培养获取知识 (自主学习)、共享知识 (团队合作)、运用知识 (解决问题)、总结知识

(技术创新)和传播知识(沟通交流)的能力与素质,训练其职业道德修养和社会责任意识,提高学生的认知能力,从而为学生提供真实世界的学习经验。

(2) 围绕现代工程师应具备的素质要求,多方位多角度地培养学生的工程能力和计算思维能力。

本书利用“想想议议”、“思考与探索”、“知识进阶”、“项目交流”、“角色模拟”、“思辩题”、“能力拓展与训练”、“我的问题卡片”等栏目多方位多角度地培养学生工程能力,包括终身学习能力、团队工作和交流能力、在社会及企业环境下建造产品的系统能力等。

“想想议议”是一些启发性较强、难度不太大的问题,旨在培养学生善于观察、勤于思考、乐于讨论的良好学习习惯和品质。有一些“想想议议”问题与社会、生活和科技发展紧密联系,旨在培养学生解决实际问题的能力。

“思考与探索”是面向计算思维的对于知识的一种解析,旨在培养学生的计算思维能力和善于观察、勤于思考、乐于探索的良好学习习惯和品质。

“知识进阶”和“项目交流”中包括一些思维密度较大、思维要求较高的问题和要求,旨在培养学生的系统思维能力、发散思维能力、创新思维能力、沟通能力、自信心、适应变化的能力以及团队协作创新的工作理念。

“思辩题”旨在培养学生的批判性和创造性思维。

“角色模拟”主要是通过模拟工程师与真实世界之间的互动,旨在培养学生的工程实践应用能力,培养学生在团队中有效合作、有效沟通、有效管理的能力,提高学生应用工程知识的能力和解决真实世界问题的能力。

“能力拓展与训练”有利于激发学生的自主探究性,在拓展创作中实现自我价值,并培养主动学习、经验学习和终身学习的能力。

“我的问题卡片”旨在培养学生主动学习、自主学习的能力和积极的态度。

(3) 贴近学生生活,倡导“快乐学习”理念。

本书精选贴近学生生活具有趣味性和实用性的项目实例“学生管理系统”,按照教学规律和学生的认知特点将知识点融于项目实例中。

(4) 强调人文素质培养。

本书每章后附有“你我共勉”栏目,旨在培养学生良好的学习态度和职业道德。

总之,本书在适度的基础知识与理论体系覆盖下,突出了工程教育的教学方法论,主要特色在于采用创新教学方法和学习环境,为学生提供真实世界的学习经验,力求达到CDIO改革的总体目标。

本书共分7章,内容包括绪论、Visual Basic 概述、程序设计基础、用户界面设计、文件、图形操作和数据库的应用。

本书由薛红梅、张永强任主编,申艳光、刘志敏、王彬丽、孔外平、马丽艳、王瑞林任副主编,统稿工作由薛红梅、刘志敏、王彬丽和孔外平完成。各章编写分工如下:第1章由张永强编写,第2章由薛红梅编写,第3章由申艳光编写,第4章由刘志敏编写,第5章由王瑞林编写,第6章由马丽艳编写,第7章由王彬丽编写。本书在编写过程中得到了河北工程大学领导和计算机基础教研室全体教师的大力支持,在此表示深深的敬意和感谢。

限于作者的水平及时间仓促,书中难免存在不足之处,恳请读者批评和指正,以使其更臻完善。

本书有配套的实验实训教材,同时提供电子课件和项目素材,可以登录科学出版社网站下载。与本书内容相关的视频,读者可以登录中国大学视频公开课官方网站“爱课程”网(<http://www.icourses.cn>),学习河北工程大学的“心连‘芯’的思维之旅”课程。

编者

2015年1月

目 录

前言

| | |
|--------------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 认识计算思维 | 1 |
| 1.2 程序与算法 | 2 |
| 1.2.1 算法的概念与分类 | 2 |
| 1.2.2 数学建模 | 3 |
| 1.2.3 算法的描述 | 5 |
| 1.2.4 算法的实现——程序设计语言 | 7 |
| 1.3 程序与数据 | 9 |
| 1.3.1 内存数据的基本组织和管理方式——数据结构 | 9 |
| 1.3.2 外存数据的基本组织和管理方式——文件系统和数据库 | 13 |
| 1.4 软件与软件工程 | 15 |
| 1.4.1 什么是软件 | 15 |
| 1.4.2 软件工程的观念 | 16 |
| 1.4.3 软件生命周期——以学生管理系统为例 | 17 |
| 1.4.4 软件工程方法 | 21 |
| 第 2 章 Visual Basic 概述 | 28 |
| 2.1 Visual Basic 6.0 简介 | 28 |
| 2.1.1 Visual Basic 6.0 的特点 | 28 |
| 2.1.2 Visual Basic 6.0 的启动 | 29 |
| 2.2 认识 VB 的集成开发环境 | 30 |
| 2.3 Visual Basic 基本概念 | 34 |
| 2.3.1 对象和类 | 34 |
| 2.3.2 对象的属性、方法和事件 | 34 |
| 2.3.3 工程和工程文件 | 37 |
| 2.4 创建 Visual Basic 应用程序的基本步骤 | 38 |
| 2.4.1 创建应用程序界面 | 38 |
| 2.4.2 设置对象的属性 | 39 |
| 2.4.3 编写过程代码 | 39 |
| 2.4.4 保存工程 | 39 |
| 2.4.5 执行程序 | 40 |
| 2.5 项目 系统登录窗口和主窗口 | 40 |
| 2.5.1 窗体 | 43 |
| 2.5.2 控件分类 | 46 |
| 2.5.3 控件的基本操作 | 47 |
| 2.5.4 命令按钮 | 48 |

| | | |
|------------|---------------------|------------|
| 2.5.5 | 文本框 | 49 |
| 2.5.6 | 标签 | 50 |
| 第3章 | 程序设计基础 | 53 |
| 3.1 | 项目 学生成绩输入 | 53 |
| 3.1.1 | 程序的编码基础 | 55 |
| 3.1.2 | 数据类型 | 56 |
| 3.1.3 | 变量 | 57 |
| 3.1.4 | 常量 | 59 |
| 3.1.5 | 变量的作用域 | 59 |
| 3.1.6 | 运算符与表达式 | 62 |
| 3.1.7 | 常用内部函数 | 65 |
| 3.1.8 | 顺序结构 | 69 |
| 3.2 | 项目 学生成绩评定 | 74 |
| 3.2.1 | 选择结构 | 75 |
| 3.2.2 | 判定结构嵌套 | 79 |
| 3.2.3 | 条件函数 | 80 |
| 3.3 | 项目 学生成绩统计 | 81 |
| 3.3.1 | 循环结构 | 81 |
| 3.3.2 | 循环嵌套 | 86 |
| 3.4 | 项目 学生成绩统计分析 | 87 |
| 3.4.1 | 数组 | 89 |
| 3.4.2 | 自定义数据类型 | 93 |
| 3.5 | 项目 小助手——组队方法 | 94 |
| 3.5.1 | 过程 | 95 |
| 3.5.2 | 参数传递 | 98 |
| 3.6 | 常用算法设计和算法分析 | 101 |
| 3.6.1 | 常用算法设计 | 101 |
| 3.6.2 | 算法分析 | 114 |
| 3.7 | 知识进阶 | 117 |
| 3.7.1 | 编码约定 | 117 |
| 3.7.2 | 结构化编码 | 118 |
| 第4章 | 用户界面设计 | 122 |
| 4.1 | 用户界面设计原则 | 122 |
| 4.2 | 项目 档案管理之信息录入 | 124 |
| 4.2.1 | 复选框 | 127 |
| 4.2.2 | 单选按钮 | 128 |
| 4.2.3 | 框架 | 129 |
| 4.2.4 | 列表框 | 129 |
| 4.2.5 | 组合框 | 131 |
| 4.2.6 | 水平滚动条和垂直滚动条 | 132 |
| 4.2.7 | 定时器控件 | 133 |

| | | |
|--------------|--------------------------|------------|
| 4.2.8 | 图片框 | 133 |
| 4.2.9 | 图像框 | 134 |
| 4.3 | 项目 档案管理之菜单设计 | 135 |
| 4.3.1 | 菜单的基本概念 | 136 |
| 4.3.2 | 菜单编辑器 | 137 |
| 4.3.3 | 菜单的制作 | 137 |
| 4.3.4 | 弹出式菜单 | 138 |
| 4.4 | 项目 档案管理之工具栏设计 | 139 |
| 4.4.1 | 添加工具栏控件组 | 141 |
| 4.4.2 | 添加 ImageList 控件 | 142 |
| 4.4.3 | 工具栏添加按钮 | 142 |
| 4.4.4 | 为工具栏编写代码 | 143 |
| 4.5 | 知识进阶 | 143 |
| 4.5.1 | 界面所涉及的元素 | 143 |
| 4.5.2 | 界面属性 | 143 |
| 4.5.3 | 统一管理 VB 控件的界面属性 | 145 |
| 第 5 章 | 文件 | 148 |
| 5.1 | 项目 档案管理之信息存储 | 148 |
| 5.1.1 | 文件的基本概念和结构 | 152 |
| 5.1.2 | 文件的分类 | 152 |
| 5.1.3 | 顺序文件的操作 | 152 |
| 5.1.4 | 随机文件的操作 | 156 |
| 5.1.5 | 二进制文件的操作 | 159 |
| 5.1.6 | 文件系统控件 | 159 |
| 5.2 | 知识进阶 | 162 |
| 5.2.1 | 通用对话框 | 162 |
| 5.2.2 | 常用的文件操作语句、函数和属性 | 164 |
| 第 6 章 | 图形操作 | 168 |
| 6.1 | 项目 绘图板 | 168 |
| 6.1.1 | 坐标系统概述 | 172 |
| 6.1.2 | 使用 Visual Basic 作图 | 174 |
| 6.1.3 | 鼠标事件 | 178 |
| 6.1.4 | 键盘事件 | 180 |
| 6.2 | 知识进阶 | 182 |
| 6.2.1 | 打印机 | 182 |
| 6.2.2 | 屏幕 | 183 |
| 6.2.3 | 剪贴板 | 184 |
| 第 7 章 | 数据库的应用 | 187 |
| 7.1 | 项目 学生管理系统数据库设计 | 187 |
| 7.1.1 | 数据库中使用的术语 | 193 |
| 7.1.2 | 数据库的建立 | 194 |

| | | |
|-------|---------------|-----|
| 7.1.3 | Data 控件 | 196 |
| 7.1.4 | 使用代码管理数据库 | 199 |
| 7.1.5 | ADO 控件 | 203 |
| 7.1.6 | 数据绑定控件 | 206 |
| 7.1.7 | SQL | 207 |
| 7.2 | 知识进阶 | 212 |
| 7.2.1 | 创建报表 | 212 |
| 7.2.2 | 将报表导出到 HTML 上 | 213 |
| | 参考文献 | 216 |

第1章 绪 论

现实世界中，必须通过人类的思维将问题进行形式化、程序化和机械化后，变成计算思维，才能利用计算机来进行问题求解。问题求解的核心是算法和系统设计。程序方式是人类使用计算机的高级方式，程序反映了人类求解问题的思维和方法。

1.1 认识计算思维

近年来，移动通信、普适计算、物联网、云计算、大数据这些新概念和新技术的出现，在社会经济、人文科学、自然科学的许多领域引发了一系列革命性的突破，极大地改变了人们对计算和计算机的认识。无处不在、无事不用的计算思维成为人们认识和解决问题的能力之一。

2006年3月，美国卡内基·梅隆大学计算机系主任周以真(Wing)在美国计算机权威期刊*Communication of the ACM*上发表并定义了计算思维(Computational Thinking)。她认为：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的涵盖计算机科学领域的一系列思维活动。她指出，计算思维是每个人的基本技能，不仅仅属于计算机科学家。每个人在培养解析能力时不仅要掌握阅读、写作和算术(Reading, Writing and Arithmetic)，还要学会计算思维。这种思维方式对于学生从事任何事业都是有益的。

计算方法和模型给了人们勇气处理那些原本无法由任何个人独自完成的问题求解和系统设计。计算思维直面机器智能的不解之谜：什么人能比计算机做得更好？什么计算机能比人类做得更好？

“人类的特性恰恰就是自由的有意识的活动。”——马克思。自古以来，所有的教育都是为了人类的发展。人之发展，首在思维，一个人的科学思维能力的养成，必然伴随着创新能力的提高。工程师应该具备的三种思维模式是工程思维、科学思维和系统思维。而其中科学思维可以分为三种：以观察和归纳自然(包括人类社会活动)规律为特征的实证思维；以推理和演绎为特征的逻辑思维；以抽象化和自动化为特征的计算思维。

计算思维综合了数学思维(求解问题的方法)、工程思维(设计、评价大型复杂系统)和科学思维(理解可计算性、智能、心理和人类行为)。

计算思维就是把一个看起来困难的问题重新阐述成一个人们知道怎样解决的问题，采用的方法有约简、嵌入、转化和仿真等。

计算思维是一种递归思维，它是并行处理过程，把代码译成数据又把数据译成代码。它评价一个程序时，不仅仅依据其准确性和效率，还有美学的考量，而对于系统的设计，还考虑简洁性和优雅性。

计算思维采用了抽象和分解的方法来迎战浩大复杂的任务，选择合适的方式陈述一个问题，或者对一个问题的相关方面进行建模使其易于处理。

计算思维是通过冗余、容错、纠错的方式，在最坏情况下进行预防、保护和恢复的一种思维。计算思维利用启发式推理来寻求解答。它就是在不确定情况下的规划、学习和调度。它就是搜索、搜索、再搜索，最后得到的是一系列网页、一个赢得游戏的策略，或者一个反例。计算思维利用

海量的数据来加快计算。它就是在时间和空间之间，在处理能力和存储容量之间的权衡。考虑日常生活中的事例：当一位学生早晨去学校时，他把当天需要的东西放进书包，这就是预置和缓存；当一个孩子弄丢他的手套时，你建议他沿走过的路回寻，这就是回溯；在什么时候你停止租用滑雪板而为自己买一对呢？这就是在线算法；在超市付账时你应当去排哪个队呢？这就是多服务器系统的性能模型；为什么停电时你的电话仍然可用？这就是失败的无关性和设计的冗余性。

人们已见证了计算思维在其他学科中的影响。例如，计算生物学正在改变着生物学家的思考方式。类似地，计算博弈理论正改变着经济学家的思考方式，纳米计算改变着化学家的思考方式，量子计算改变着物理学家的思考方式。计算思维将成为每一个人的技能，它是人类除了理论思维、实验思维以外，应具备的第三种思维方式。

思考与探索

符号化、计算化、自动化思维，以组合、抽象和递归为特征的程序及其构造思维是计算技术与计算系统的重要思维。计算思维能力训练不仅使人们理解计算机的实现机制和约束、建立计算意识、形成计算能力，有利于发明和创新，而且有利于提高信息素养，也就是处理计算机问题时应有的思维方法、表达形式和行为习惯，从而更有效地利用计算机。

计算思维的抽象体现在使用符号代替实际问题中的各种变量，计算思维的自动化则体现在程序的机械式执行，而实现自动化则依赖于完备的算法。

计算思维建立在计算机的能力和限制上，所以用计算机解决问题时，既要充分利用计算机的计算和存储能力，又不能超出计算机的能力范围。

1.2 程序与算法

1976年，瑞士苏黎士联邦工业大学的科学家 Wirth (Pascal 语言的发明者，1984年图灵奖获得者)发表了专著，其中提出公式“程序=算法+数据结构”(Algorithms+Data Structures=Programs)，这一公式的关键是指出了程序是由算法和数据结构有机结合构成的。程序是完成某一任务的指令或语句的有序集合；数据是程序处理的对象和结果。就像写文章，文章=材料+构思，构思是文章的灵魂，同样算法是程序的灵魂，也是计算的灵魂，在计算思维中占据重要地位。

1.2.1 算法的概念与分类

做任何事情都有一定的步骤。例如，学生考大学，首先要填报名单，交报名费，拿准考证，然后参加全国高考，得到录取通知书，到指定大学报到。又如，网上预订火车票步骤如下：①登录中国铁路客户服务中心(12306网站)，下载根证书并安装到计算机上；②到网站上注册个人信息，注册完毕，到信箱里单击链接激活注册用户；③进行车票查询；④进入订票页面，提交订单，通过网上银行进行支付；⑤凭乘车人有效二代居民身份证原件到全国火车站的任意售票窗口、铁路客票代售点或车站自动售票机上办理取票手续。

1. 算法的定义

人们从事各种工作和活动，都必须事先想好需要进行的步骤，这种为解决一个确定类问题而采取的方法和步骤称为算法(Algorithm)。算法规定了任务执行或问题求解的一系列步骤。例如，菜谱是做菜的“算法”；歌谱是一首歌曲的“算法”；洗衣机说明书是洗衣机使用的“算法”等。

算法不仅是计算机科学的一个分支，更是计算机科学的核心。计算机算法能够帮助人类解决很多问题，例如，找出人类 DNA 中所有 100000 种基因，确定构成人类 DNA 的 30 亿种化学基对的序列；快速地访问和检索互联网数据；电子商务活动中各种信息的加密及签名；制造业中各种资源的有效分配；确定地图中两地之间的最短路径；各种数学和几何计算(矩阵、方程、集合)。

2. 算法的特征

一个算法应该具有以下五个重要的特征。

(1) 确切性。算法每一个步骤必须具有确切的定义，不能有二义性。

(2) 可行性。算法中执行的任何计算步骤都可以被分解为基本的可执行的操作步骤，即每个计算步骤都可以在有限时间内完成(也称为有效性)。

(3) 输入项。一个算法有 0 个或多个输入，以刻画运算对象的初始情况，所谓 0 个输入是指算法本身设定了初始条件。

(4) 输出项。一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

(5) 有穷性。一个算法必须保证执行有限步后结束。

例如，操作系统是一个在无限循环中执行的程序，因而不是一个算法。但操作系统的各种任务可看成单独的问题，每一个问题由操作系统中的一个子程序通过特定的算法来实现。该子程序得到输出结果后便终止。

3. 算法的分类

按照算法所使用的技术领域，算法可大致分为基本算法、数据结构算法、数论与代数算法、计算几何的算法、图论的算法、动态规划以及数值分析、加密算法、排序算法、检索算法、随机化算法、并行算法、厄米变形模型和随机森林算法。

按照算法的形式，算法可分为以下三种。

(1) 生活算法：完成某一项工作的方法和步骤。

(2) 数学算法：对一类计算问题的机械的统一的求解方法，如求一元二次方程的解，以及求圆面积、立方体的体积等。

(3) 计算机算法：对运用计算思维设计的问题求解方案的精确描述。例如，人们玩扑克牌的时候，如果要求同花色的牌放在一起而且从小到大排序，人们一般都会边摸牌边把每张牌依次插入合适的位置，等把牌摸完了，牌的顺序也排好了。这是摸牌的过程，也是一种算法。计算机学科就把这个生活算法转化成了计算机算法，称为插入排序算法。

1.2.2 数学建模

问题求解中的计算思维的过程为：首先建立问题的模型，然后根据模型设计相应的算法。

数学建模是运用数学的语言和方法，通过抽象、简化建立对问题进行精确描述和定义的数学模型。简单来说，就是抽象出问题，并用数学语言进行形式化描述。

一些表面上看似非数值的问题，进行数学形式化后，就可以方便地进行算法设计。

如果研究的问题是特殊的，例如，今天所做的事情的顺序，因为每天不一样，就没有必要建立模型。如果研究的问题具有一般性，就有必要体现模型的抽象性质，为这类事件建立数学模型。模型是一类问题的解题步骤，即一类问题的算法。广义的算法就是事情的次序，**算法提供一种解决问题的通用方法。**

【例 1.2.1】 国际会议排座位问题。

现要举行一个国际会议，有 7 个人分别用 a 、 b 、 c 、 d 、 e 、 f 、 g 表示。已知下列事实： a 会讲英语； b 会讲英语和汉语； c 会讲英语、意大利语和俄语； d 会讲日语和汉语； e 会讲德语和意大利语； f 会讲法语、日语和俄语； g 会讲法语和德语。

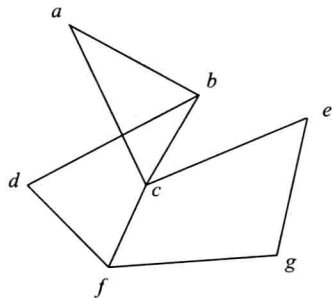


图 1.2.1 用数学语言来表示的问题模型

试问：这 7 个人应如何排座位，才能使每个人都能和他身边的人顺利地沟通交谈？

问题分析：可以尝试将这个问题转化为图的形式，建立一个图的模型，将每个人抽象为一个节点，人和人的关系用节点间的关系——边来表示。于是得到节点集合 $V = \{a, b, c, d, e, f, g\}$ 。对于任意的两点，若有共同语言，就在它们之间连一条无向边，可得边的集合 $E = \{ab, ac, bd, bc, df, cf, ce, fg, eg\}$ ，图 $G = \{V, E\}$ ，如图 1.2.1 所示。

这时问题转化为在图 G 中找到一条哈密顿回路的问题。

哈密顿图是一个无向图，由天文学家哈密顿提出。哈密顿回路 (Hamiltonian Path) 是指从图中的任意一点出发，经过图中每一个节点一次。这样便从图中得出， $abdfgeca$ 是一条哈密顿回路，照此顺序排座位即可满足问题要求。

【例 1.2.2】 警察抓小偷的问题。

警察抓了 a 、 b 、 c 、 d 四名偷窃嫌疑犯，其中只有一人是小偷，审问记录如下。

a 说：“我不是小偷。”

b 说：“ c 是小偷。”

c 说：“小偷肯定是 d 。”

d 说：“ c 在冤枉人。”

已知四个人中三人说的是真话，一人说的是假话，到底谁是小偷？

问题分析：依次假设每个人是小偷的情况，然后一一代入那四句话，依次检验已知条件“四个人中三人说的是真话，一人说的是假话”是否成立，如果成立，那么对应的假设成立，小偷找到。

计算机算法设计如下。

(1) 将 a 、 b 、 c 、 d 四个人编号为 1、2、3、4。

(2) 用变量 x 存放小偷的编号。

(3) 依次将 $x=1, x=2, x=3, x=4$ 代入问题系统，检验“四个人中三人说的是真话，一人说的是假话”是否成立。

问题系统如下。

(1) a 说：“我不是小偷。”

(2) b 说：“ c 是小偷。”

(3) c 说：“小偷肯定是 d 。”

(4) d 说：“ c 在冤枉人。”

(5) 四个人中三人说的是真话，一人说的是假话。

分别翻译成计算机的形式化语言如下。

(1) a 说： $x \neq 1$ 。

(2) b 说： $x=3$ 。

(3) c 说： $x=4$ 。

(4) d 说: $x \neq 4$ 。

(5) 四个逻辑式的值相加, 和为 $1+1+1+0=3$ 。

这时候就便于计算机理解了。

数学建模的实质是: 提取操作对象 \rightarrow 找出对象间的关系 \rightarrow 用数学语言进行描述。

思考与探索

抽象是一种重要的思维方式。计算机科学中的抽象包括数据抽象和控制抽象。

从科学研究到艺术创造都离不开抽象。毕加索认为, 画家的职责不是借助具体形象来反映现实, 而是创造抽象的形式来表现科学的真实。

1.2.3 算法的描述

算法的描述方式主要有以下几种。

1. 自然语言

自然语言是人们日常所用的语言, 这是其优点。但自然语言描述算法的缺点也有很多: 自然语言的歧义性易导致算法执行的不确定性; 自然语言语句一般太长导致算法的描述太长; 当算法中循环和分支较多时就很难清晰表示; 不便翻译成程序设计语言。因此, 人们又设计出流程图等图形工具来描述算法。

【例 1.2.3】 已知圆半径, 计算圆面积。

可以用自然语言表达出以下算法步骤。

(1) 输入圆半径 r 。

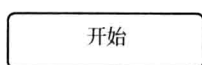
(2) 计算 $S=3.14 \times r \times r$ 。

(3) 输出 S 。

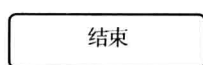
2. 流程图

程序流程图简洁、直观、无二义性, 是描述程序的常用工具, 一般采用美国国家标准学会规定的一组图形符号, 如图 1.2.2 所示。

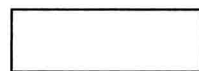
对于十分复杂难解的问题, 框图可以画得粗略一些、抽象一些, 首先表达出解决问题的轮廓, 然后细化。流程图也存在缺点: 使设计人员过早考虑算法控制流程, 而不考虑全局结构, 不利于逐步求精; 随意性太强, 结构化不明显; 不易表示数据结构; 层次感不明显。



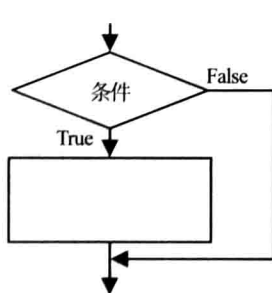
(a) 开始框——用于流程的开始



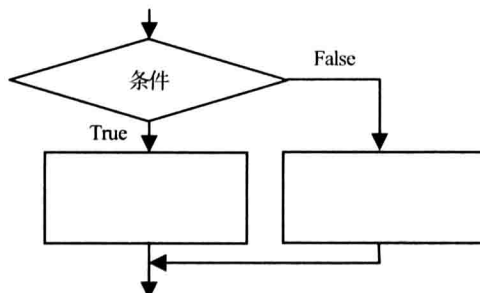
(b) 结束框——用于流程的结束



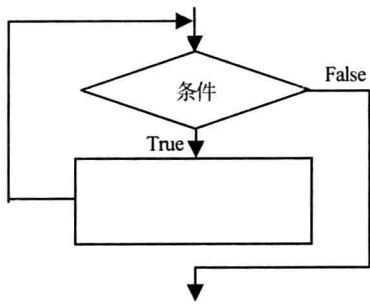
(c) 功能框——用来完成计算等功能



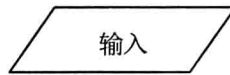
(d) 单分支判断框——用于解决单分支问题



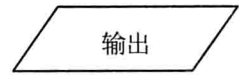
(e) 双分支判断框——用于解决双分支问题



(f) 循环框——用于解决需要反复执行的问题



(g) 输入框——向程序输入数据



(h) 输出框——程序向外输出信息

图 1.2.2 程序流程图常用图形元素

【例 1.2.4】 用流程图表示例 1.2.3 的算法。

算法用自然语言表示如下。

- (1) 输入圆半径 r 。
- (2) 计算 $S=3.14 \times r \times r$ 。
- (3) 输出 S 。

用流程图表示的算法如图 1.2.3 所示。

【例 1.2.5】 计算 $1+2+3+\dots+n$, n 由键盘输入。

分析：这是一个累加的过程，每次循环累加一个整数值，整数的取值范围为 $1\sim n$ ，需要使用循环结构。

用流程图表示的算法如图 1.2.4 所示。

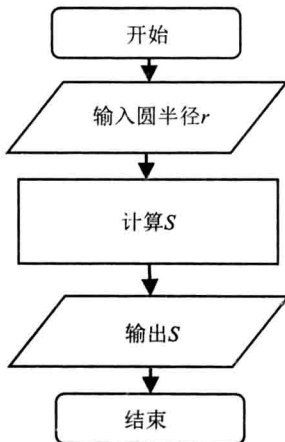


图 1.2.3 程序流程图表示的算法

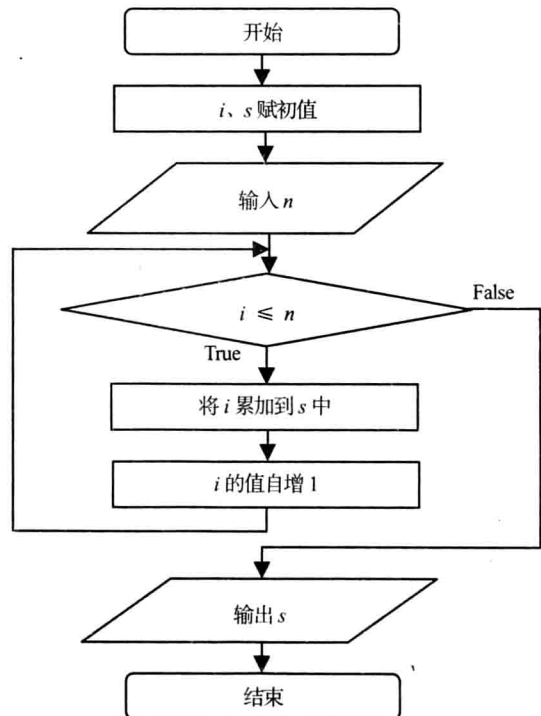


图 1.2.4 程序流程图表示的累加算法

3. 盒图 (N-S 图)

盒图层次感强、嵌套明确；支持自顶向下、逐步求精的设计方法；容易转换成高级语言，但不易扩充和修改，不易描述大型复杂算法。N-S 图中基本控制结构的表示符号如图 1.2.5 所示。

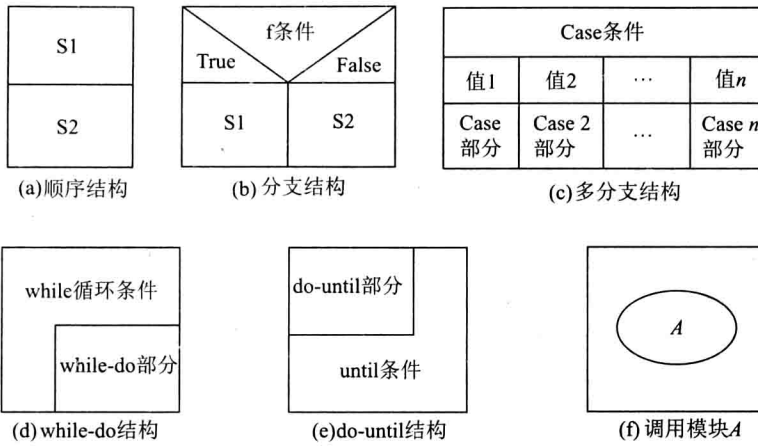


图 1.2.5 N-S 图中基本控制结构的表示符号

4. 伪代码

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法的工具。它不用图形符号，书写方便，语法结构有一定的随意性，目前还没有一个通用的伪代码语法标准。

常用的伪代码是用简化后的高级语言来进行编写的，如类 C、类 C++、类 Pascal 等。

5. 程序设计语言

以上算法的描述方式都是为了方便人与人交流，但最终算法是要在计算机上实现的，所以用程序设计语言进行算法的描述，并进行合理的数据组织，就构成了计算机可执行的程序。

与人类社会使用语言交流相似，人要与计算机交流，必须使用计算机语言。于是人们模仿人类的自然语言，人工设计出一种形式化的语言——程序设计高级语言。

1.2.4 算法的实现——程序设计语言

如果需要把用流程图、自然语言等方式描述的算法在计算机上执行，还需要用某种计算机语言表示出来，即用程序设计语言把算法翻译成机器能够理解的可执行程序。

1. 程序设计语言的分类

目前，程序设计语言按照与计算机硬件的联系程度可分为三类：机器语言、汇编语言、高级语言。

1) 机器语言 (Machine Language)

机器语言是计算机硬件系统能够直接识别的不需翻译的计算机语言。机器语言中的每一条语句实际上是一条二进制形式的指令代码，由操作码和操作数组成。操作码指出进行什么操作；操作数指出参与操作的数或在内存中的地址。用机器语言编写程序工作量大，难于记忆和使用，但执行速度快。它的二进制指令代码通常随 CPU 型号的不同而不同，不能通用，因而说它是面向机器的一种低级语言。通常不用机器语言直接编写程序。

2) 汇编语言 (Assemble Language)

汇编语言是为特定计算机或计算机系列设计的。汇编语言用助记符代替操作码，用地址符号代替操作数。由于采用这种“符号化”的做法，所以汇编语言也称为符号语言。用汇编语言编写的程序称为汇编语言源程序。汇编语言程序比机器语言程序易读、易检查、易修改，同时又保持

了机器语言程序执行速度快、占用存储空间少的优点。汇编语言也是一种面向机器的低级语言，不具备通用性和可移植性。

3) 高级语言(High Level Language)

高级语言是由各种意义的词和数学公式按照一定的语法规则组成的，它更容易阅读、理解和修改，编程效率高。高级语言不是面向机器的，而是面向问题的，与具体机器无关，具有很强的通用性和可移植性。高级语言的种类很多，有面向过程的语言，如 Fortran、BASIC、Pascal、C 等；有面向对象的语言，如 C++、Java 等。

不同的高级语言有不同的特点和应用范围。Fortran 语言是 1954 年提出的，是最早出现的一种高级语言，适用于科学和工程计算；BASIC 语言是初学者的语言，简单易学，人机对话功能强；Pascal 语言是结构化程序语言，适用于教学、科学计算、数据处理和系统软件开发，目前逐步被 C 语言所取代；C 语言程序简练、功能强大，适用于系统软件、数值计算、数据处理等，是目前高级语言中使用最多的语言之一；C++、C# 等面向对象的程序设计语言，给非计算机专业的用户在 Windows 环境下开发软件带来了福音；Java 语言是一种基于 C++ 的跨平台分布式程序设计语言。

上述通用语言仍然都是过程化语言。编码的时候，要详细描述问题求解的过程，告诉计算机每一步应该“怎样做”。为了把程序员从繁重的编码工作中解放出来，还必须寻求进一步提高编码效率的新语言，这就是甚高级语言或第 4 代语言(4GL)产生的背景。对于 4GL，迄今仍没有统一的定义。一般认为，3GL 是过程化的语言，目的在于高效地实现各种算法；4GL 则是非过程化的语言，目的在于直接实现各类应用系统。前者面向过程，需要描述“怎样做”；后者面向应用，只需说明“做什么”。

2. 语言处理程序

程序设计语言能够把算法翻译成机器能够理解的可执行程序。这里将计算机不能直接执行的非机器语言源程序翻译成能直接执行的机器语言的语言翻译程序称为**语言处理程序**。

(1) 源程序。用各种程序设计语言编写的程序称为源程序，计算机不能直接识别和执行。

(2) 目标程序。源程序必须由相应的汇编程序或编译程序翻译成机器能够识别的机器指令代码，计算机才能执行，这正是语言处理程序所要完成的任务。翻译后的机器语言程序称为目标程序。

(3) 汇编程序。将汇编语言源程序翻译成机器语言程序的翻译程序称为汇编程序。

(4) 编译方式和解释方式。编译方式是将高级语言源程序通过编译程序翻译成机器语言目标代码；解释方式是对高级语言源程序进行逐句解释，解释一句就执行一句，但不产生机器语言目标代码。例如，BASIC 语言大都是按这种方式处理的。大部分高级语言都采用编译方式。

3. 程序设计语言的基本控制结构

1996 年，计算机科学家 Boehm 和 Jacopini 提出并从数学的角度证明，任何一个算法都能以三种基本控制结构表示，即顺序结构、选择结构和循环结构。

1) 顺序结构

顺序结构是一类最基本、最简单的结构，其形式是：执行语句 1，然后执行语句 2……

顺序结构的特点：程序按照语句在代码中出现的顺序逐条执行；顺序结构中的每一条语句都被执行，而且只能被执行一次。就像将一颗颗珠子串成项链，也好像一层一层地爬上楼梯……

2) 选择结构

选择结构又称为分支结构，包括单分支和多分支结构，它根据判定条件的真假来确定应该执行哪一条分支的语句序列。