

高等学校计算机基础教育教材精选

大学计算机基础

(第3版)

李 瞰 毛晓光 刘万伟 编著
陈立前 周竞文 周海芳



清华大学出版社

高等学校计算机基础教育教材精选

大学计算机基础

(第3版)

李 瞰 毛晓光 刘万伟
陈立前 周竞文 周海芳 编著

清华大学出版社
北 京

内 容 简 介

本书兼顾计算机科学基础知识和计算思维,以计算思维能力培养为主线,串联信息表示、计算机系统、操作系统、网络、多媒体技术、科学计算及新方向等内容,选择 Python 作为计算实践的语言,内容偏重于如何将计算思维应用于计算机科学等领域以解决问题。实践内容将在授课内容的基础上进行拓展,并要求运用 Python 及相关的配套库进行问题求解练习。希望通过应用问题求解的学习和实践,培养读者在理解计算机系统的基础上,主动在各自专业学习中利用计算思维的方法和技能,进行问题求解的能力和习惯。学完本书后,希望读者能动手解决具有一定难度的实际问题。

本书适合作为高等学校计算机基础课程的教材,也可作为计算机培训、计算机等级考试和计算机初学者的参考书。本书可与《大学计算机基础实验教程(第2版)》配合使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

大学计算机基础/李瞰等编著. —3版. —北京:清华大学出版社,2018

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-50981-3

I. ①大… II. ①李… III. ①电子计算机—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2018)第 179399 号

责任编辑:白立军

封面设计:常雪影

责任校对:徐俊伟

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.25 字 数:455 千字

版 次:2012 年 8 月第 1 版 2018 年 9 月第 3 版 印 次:2018 年 9 月第 1 次印刷

印 数:1~18000

定 价:49.00 元

产品编号:079737-01

前言

大学计算机基础(第3版)

人要成功融入社会所必备的思维能力,是由其所处时代能够获得的工具决定的。计算机是信息社会的必备工具之一,如何有效利用计算机分析和解决问题,将与阅读、写作和算术一样,成为 21 世纪每个人的基本技能,而不仅仅属于计算机专业人员。计算机正在对人们的生活、工作,甚至思维产生深刻的影响。

“大学计算机基础”是大学本科教育的第一门计算机公共基础课程,它的改革越来越受到人们的关注。本课程的主要目的是从使用计算机、理解计算机系统和计算思维 3 个方面培养学生的计算机应用能力。从 2008 年开始,以“计算思维”的培养为主线开展计算科学通识教育,逐渐成为国内外计算机基础教育界的共识。

基于这种认识,第 2 版教材进行了较大幅度的修改,增加了计算思维所占的比重。指导思想是兼顾计算机基础知识和计算思维,选择 Python 作为实践语言,将信息表示与处理、计算机系统、网络、数据库、多媒体等知识既作为教学内容,又作为计算思维求解问题的研究对象,加以实践,教材内容更偏重于如何将计算思维应用于各领域求解问题。最终,通过这种问题求解的学习和实践,希望学生在理解计算机基础知识的同时,能主动在各自专业学习中利用计算思维的方法和技能,进行问题求解,能动手解决具有一定难度的实际问题。

经过一年的实践,综合各方面的反馈,对第 2 版教材进行了改版。

(1) 对计算思维的内容进行了重新梳理,在第 2 版中将计算思维、计算机问题求解的内容统一到计算思维的几个核心概念下。

(2) 对 Python 基础知识进行了重新组织,更新了案例。

(3) 将原来分布在两章的计算机硬件系统与操作系统合并成为一章,连贯性更好。

(4) 合并了信息表示和多媒体技术基础,从信息角度,把字符、数值、图像、声音等等对待和处理。

(5) 新增了一章科学计算的内容,以体现 Computing in Science 的理念,即结合高等数学和计算思维,使学生在大学入学之初就能解决现实世界复杂规模的问题。

第 3 版教材包含 9 章内容,大致可分为如下部分: 计算与社会(第 1 章);Python 简介(第 2 章);计算思维(第 3 章);信息、编码及数据表示(第 4 章);计算机系统、计算机网络及应用,数据库技术应用基础(第 5、6、7 章);科学计算(第 8 章);以及计算机发展新技术(第 9 章)。

本书内容涉及计算机专业多门课程的知识,概念庞杂,术语繁多。表面上看,章与章之间的联系松散。对于初学者来说,学好这门课程不容易,融会贯通就更加困难了。如何把握全书的脉络?建议以“信息表示和信息处理”“计算思维与计算机问题求解”作为理解章节内容联系的两条主要线索。

计算机系统是信息处理的工具,而信息处理依赖于某种形式的信息表示。本书中主要介绍了用0-1符号串表示数值信息、字符信息、声音信息和图像信息的方法。介绍了以文件和数据库形式组织信息的技术。介绍了计算机系统处理信息的工作原理。每一个计算机系统功能都涉及某类或某几类信息,每一个计算机系统功能都可以转换为信息处理过程。读者应该思考:这些信息是怎样表示的?为什么要使用这种表示方法?计算机系统的功能由哪些信息处理过程组成?这些处理过程包含哪些步骤?处理步骤是如何(自动)实现的?

在理解信息表示和信息处理的基础上,学习计算思维,是为了更好地发挥计算机的作用,解决具体问题。读者在学习计算思维时,应该考虑:以计算机基础知识为研究内容,用计算思维如何思考问题和解决问题?如何类比其他领域的问题?碰到具体问题时,可以思考该问题是否有计算的解?解是什么?如何实现解?如何让计算机帮助求解?

同时,这两条线索又是统一的,本书的案例以计算机系统本身作为对象,展示了用计算思维与计算机问题求解来研究计算机系统的方法。为读者将计算思维扩展到其他学科领域做了良好的示范。

如果这些问题都明晰了,对融会贯通全书内容有很大帮助。

本书适用于计算机专业和非计算机专业一年级新生,不要求有计算机程序设计经验,并且也不是以程序设计为主要内容,而是要求学生专注于理解计算思维求解问题的方法和技能。一些Python语言基础知识的介绍,是帮助读者阅读和理解教材中给出的Python程序。希望读者能在理解的基础上,对这些程序进行小修改,来实践自己的问题求解方法。同时建议与本书配套的《大学计算机基础实验教程(第2版)》配合使用,效果更好。

本书的第1~7章主要由李墩编写或在前两版基础上进行了更新,第8章由刘万伟编写,第9章由陈立前编写。各章案例及新增内容由李墩、毛晓光、刘万伟、周海芳、周竞文等编写。全书由李墩、毛晓光负责统稿。王志英、宁洪、陈怀义、王保恒等教授对本书的编写给予了许多指导,陈立前、周竞文为本书的文字整理和校对做了大量工作。此外,本书还参考了很多文献资料和网络素材,在此一并表示衷心的感谢。

本书的写作集体根据多年的教学实践,在内容的甄选、全书组织形式等方面既借鉴了同类书的成功经验,也做出了自己的努力。但是改进的空间还很大,热切希望广大读者能够予以斧正。

编 者

2018年7月

目录

大学计算机基础(第3版)

第1章 计算与社会	1
1.1 计算概论	1
1.2 计算装置发展简史	4
1.2.1 机械式计算装置	4
1.2.2 图灵机和图灵	7
1.2.3 现代电子计算机	10
1.2.4 计算机的发展趋势	12
1.3 计算技术的应用	14
1.4 信息化社会与人	19
1.5 计算思维概论	21
1.6 小结	23
1.7 习题	23
第2章 Python 简介	25
2.1 引言	25
2.2 Python 基本元素	26
2.2.1 对象、表达式和数值类型	27
2.2.2 变量和赋值	28
2.2.3 str 类型与输入	29
2.3 内置数据结构	31
2.3.1 列表	31
2.3.2 元组	33
2.3.3 字典	34
2.4 控制语句	35
2.4.1 分支语句	35
2.4.2 循环	36
2.5 函数	37

2.6	使用模块.....	40
2.7	面向对象基础.....	41
2.8	Python 编程示例——打印月历	45
2.9	小结.....	49
2.10	习题	49
第 3 章	计算思维	52
3.1	概述.....	53
3.2	逻辑思维与算法思维.....	56
3.2.1	逻辑思维	56
3.2.2	算法思维	59
3.2.3	小结	61
3.3	问题求解策略.....	62
3.3.1	基本步骤	62
3.3.2	分解法	63
3.3.3	模式与归纳	65
3.3.4	小结	68
3.4	抽象与建模.....	68
3.4.1	抽象	68
3.4.2	建模	71
3.5	评价解决方案.....	74
3.5.1	解是否正确	74
3.5.2	解的效率如何	76
3.5.3	小结	77
3.6	算法、数据结构与程序	78
3.6.1	算法设计常用策略	78
3.6.2	算法的描述	79
3.6.3	算法示例	81
3.6.4	数据结构	86
3.6.5	程序设计语言	87
3.7	“捉狐狸”问题求解示例.....	90
3.8	小结.....	94
3.9	习题.....	94
第 4 章	信息、编码及数据表示	96
4.1	信息论基础.....	96
4.2	编码及其解释.....	99
4.3	数值的数字化	103

4.4	计算机数值表示	105
4.4.1	计算机码制	105
4.4.2	定点数和浮点数	108
4.5	字符的数字化	110
4.5.1	汉字编码	111
4.5.2	Unicode 码	112
4.6	声音的数字化	113
4.7	图像的数字化	115
4.8	信息处理示例	119
4.8.1	数据压缩示例及 Python 实现	119
4.8.2	生成图像验证码及 Python 实现	122
4.8.3	Python 绘制分形图形	123
4.9	小结	126
4.10	习题	126
第 5 章	计算机系统	130
5.1	概述	131
5.2	计算机硬件系统	133
5.2.1	中央处理器	134
5.2.2	存储系统	138
5.2.3	总线	141
5.2.4	输入输出系统	142
5.3	操作系统	143
5.3.1	概述	144
5.3.2	进程管理	145
5.3.3	存储管理	149
5.3.4	文件管理	150
5.3.5	设备管理	152
5.3.6	用户接口	153
5.3.7	操作系统的加载	155
5.4	Python 构建冯·诺依曼体系结构模拟器	156
5.5	利用 Python 使用操作系统	159
5.5.1	利用 Python 查看进程信息	159
5.5.2	利用 Python 查看系统存储信息	161
5.5.3	Python 文件操作	163
5.6	小结	165
5.7	习题	165

第 6 章 计算机网络及应用..... 168

6.1 计算机网络基础 168

6.1.1 计算机网络的发展历史..... 169

6.1.2 计算机网络的分类..... 171

6.1.3 计算机网络体系结构与协议..... 172

6.1.4 计算机网络传输介质及设备..... 177

6.2 Internet 基础 179

6.2.1 Internet 概述 179

6.2.2 TCP/IP 协议 182

6.2.3 Python TCP/IP 网络编程 186

6.3 Internet 应用 190

6.3.1 万维网..... 190

6.3.2 电子邮件..... 192

6.3.3 文件传输..... 194

6.3.4 搜索引擎..... 196

6.3.5 Python 编程示例 196

6.4 无线网络 198

6.5 物联网 201

6.6 小结 202

6.7 习题 203

第 7 章 数据库技术应用基础 204

7.1 概述 204

7.1.1 数据管理发展简史..... 206

7.1.2 数据库的基本概念..... 206

7.1.3 数据库技术管理数据的主要特征..... 208

7.1.4 数据库的应用..... 209

7.2 数据模型 211

7.2.1 概念模型..... 212

7.2.2 逻辑模型..... 216

7.2.3 E-R 模型到关系模型的转化..... 221

7.3 数据库管理系统 222

7.3.1 数据库管理系统的功能..... 222

7.3.2 常见数据库管理系统软件..... 223

7.4 Python 数据库程序设计示例 225

7.5 Python 数据分析示例 226

7.6 小结 231

VI

大学计算机基础(第 3 版)

7.7 习题	232
第 8 章 科学计算	233
8.1 泰勒级数	234
8.1.1 泰勒级数的主项	234
8.1.2 余项及误差	236
8.2 插值及拟合	238
8.2.1 拉格朗日插值	238
8.2.2 牛顿插值	239
8.2.3 埃尔米特插值	241
8.2.4 函数拟合	242
8.3 数值微积分	244
8.3.1 数值微分	244
8.3.2 数值积分	247
8.4 非线性方程数值解	249
8.4.1 二分法求根	249
8.4.2 函数迭代法求根	250
8.4.3 牛顿迭代法求根	251
8.5 线性方程组求解	252
8.5.1 直接法求解	252
8.5.2 迭代法求解	255
8.6 符号计算	257
8.7 小结	262
8.8 习题	262
第 9 章 计算机发展新技术	264
9.1 高性能计算	265
9.1.1 高性能计算的含义及意义	265
9.1.2 高性能计算的关键技术	266
9.1.3 高性能计算的典型应用	270
9.1.4 高性能计算的发展挑战	270
9.1.5 Python 高性能编程——计算 π	272
9.2 云计算与大数据	274
9.2.1 云计算	274
9.2.2 大数据	276
9.3 人工智能	278
9.3.1 人工智能的基本概念与发展历程	278
9.3.2 搜索	280

9.3.3	知识表示与推理	281
9.3.4	机器学习	283
9.3.5	智能控制	285
9.3.6	Python 机器学习示例——预测外卖配送时间	286
9.4	新型计算技术	289
9.4.1	量子计算	289
9.4.2	光计算	290
9.4.3	生物计算	291
9.5	小结	292
9.6	习题	292
参考文献		294

【学习内容】

本章作为本书的引言,主要知识点如下。

- (1) 计算的概念及内涵。
- (2) 算法的概念。
- (3) 计算装置的发展简史。
- (4) 计算技术的应用。
- (5) 计算机文化相关内容。
- (6) 计算思维概论。

【学习目标】

通过本章的学习,读者应该掌握以下内容。

- (1) 理解计算的概念及内涵。
- (2) 理解算法的基本概念。
- (3) 了解计算装置的发展规律。
- (4) 了解图灵机的工作原理及其意义。
- (5) 了解计算技术的应用领域及作用。
- (6) 了解计算技术对工作、生活的影响。
- (7) 了解信息化社会对人的素质要求,了解相关社会和法律问题。
- (8) 了解计算思维的基本概念。

计算机系统是通用的、计算能力强大的工具,在社会生活的各个方面都有广泛的应用。欲深入、有效使用计算机,必须理解计算的相关概念。本章介绍计算和算法的相关概念,简述计算装置的发展历程,介绍计算机技术的应用,以及信息化社会对人的素质和技能(计算思维)要求。

1.1 计算概论

从小学开始,“计算”这个词就不断出现在日常生活、数学作业中,例如,对“苹果 18 元一千克,算一下买 3 千克苹果要多少钱?”这个问题,一般可用两种方法进行解答:一是

3 个 18 相加,二是 18 乘以 3。对第一种方法,通常列出竖式,个位与个位对齐,十位与十位对齐,然后将个位上的 3 个 8 相加,得到 24,直接在结果的个位写上 4,2 进位到十位,与 3 个 1 相加得到 5,结果为 54。对第二种方法,也可列竖式,首先将 18 的个位 8 与 3 相乘,得到 24,将 4 写到结果的个位上,2 进位到十位,十位的 1 与 3 相乘得 3,与进位的 2 相加得 5,结果也为 54。当然,这样的问题也可直接用计算器求解,输入 18×3 就能得到结果。从这个例子,可以看出计算的一些特性,给出如下定义:

计算指的是在某计算装置上,根据已知条件,从某一个初始点开始,在完成一组良好定义的操作序列后,得到预期结果的过程。

对这个定义,有以下两点需要注意。

(1) 计算的过程可由人或某种计算装置执行。

(2) 同一个计算可由不同的技术实现。

在人类历史上,计算的作用受到了人脑运算速度和手工记录计算结果的制约,使得能通过计算解决的问题规模非常小。相对于制约计算的人的因素,计算机(Computer)非常擅长于做(也只能做)两件事情:运算和记住运算的结果。随着计算机的出现,以及计算机运算速度的不断提高,能通过计算解决的问题越来越多,问题规模越来越大,即越来越多的问题被证明存在计算的解(Computational Solution)。所谓有计算的解,指的是对某个问题,能通过定义一组操作序列,按照该操作序列行为得到该问题的解。

一般来说,知识可分为陈述性的或过程性的。陈述性知识(Declarative Knowledge)是对事实的描述。例如,“ x 的平方根是一个数 y ,使得 $y \times y = x$ ”。但是,从平方根的描述,无法知道如何去求某数的平方根。而过程性知识(Imperative Knowledge)描述的是“如何做”,或演绎信息的动作序列。例如,古希腊数学家希罗第一次给出了一种计算平方根的方法,描述如下。

- (1) 对给定的数 x ,猜测其平方根为 g 。
- (2) 如果 $g \times g$ 足够逼近 x ,停止,并报告 g 就是 x 的平方根。
- (3) 否则,用 g 和 x/g 的平均值作为新的猜测。
- (4) 将新的猜测仍记为 g ,重复上述过程,直到 $g \times g$ 足够逼近 x 。

例如,用上面的方法求 49 的平方根,计算过程如下。

- (1) 猜测 49 的平方根为 6,即 g 为 6。
- (2) $6 \times 6 = 36$ 不够逼近 49。
- (3) 令 $g = (6 + 49/6)/2 \approx 7.08333$ 。
- (4) $7.08333 \times 7.08333 \approx 50.17$,不够逼近 49。
- (5) 令 $g = (7.08333 + 49/7.08333)/2 \approx 7.00049$ 。
- (6) $7.00049 \times 7.00049 \approx 49.007$,已足够逼近 49,停止,并称 7.00049 足够近似于 49 的平方根。

希罗求平方根的方法是由一组简单动作的序列,以及规定每一个动作何时执行的控制构成的。这就是计算定义中所指的“一组良好定义的操作序列”,又称为算法(Algorithm)。

算法是求解问题类的、机械的、统一的方法,它由有限个步骤组成,对于问题类中的每

个给定的具体问题,机械地执行这些步骤就可以得到问题的解答。

可以用两数加法的运算方法来理解算法的概念。数的个数是无限的,所以可能要做的加法也是无限次的。但是无论做多少次加法,做加法的方法是不会变的。所以,做加法的方法是一种运用有限的规则应对无限可能情况的方法。算法正是这样一种方法,它是用来解决一类问题的。

与菜谱类似——按照这些步骤就能做出这道菜——可将算法理解为遵循这些步骤,就能解决你的问题。利用一组良好定义的序列来解决问题的思路可上溯到古希腊、波斯和中国古代。例如,古希腊数学家欧几里得在公元前3世纪,就提出了寻求两个正整数的最大公约数的“辗转相除”算法,该算法被人们认为是史上第一个算法。Algorithm一词来源于波斯学者 Muhammand ibn Musa al-Khwarizmi 的名字,他定义了加、减、乘、除等运算的过程,按上述定义,这些过程即为算法。

算法通常具有如下五大特征。

- (1) 输入:一个算法必须有零个或零个以上输入量,用于描述要解决的问题。
- (2) 输出:一个算法应有一个或一个以上输出量,输出量是算法计算的结果。
- (3) 明确性:算法的每个步骤都必须精确地定义,拟执行动作的每一步都必须严格地、无歧义地描述清楚,以保证算法的实际执行结果精确地符合要求或期望。
- (4) 有限性:算法必须在有限个步骤内终止。
- (5) 有效性:又称为可行性或能行性,是指算法的所有运算必须是充分基本的,因而原则上人们可以使用笔和纸在有限时间内精确地完成它们。

算法描述了对数据进行加工处理的顺序和方法,从上面希罗算法的例子可以看出,动作难以严格按照所给顺序一个一个地进行,不可避免地会遇到需要进行选择或不断重复的情况。通常使用顺序结构、选择结构和循环结构3种控制结构来组织算法中的动作。

(1) 顺序结构:算法的各个动作严格按它们的先后顺序依次执行,前一个动作执行完毕后,顺序执行紧跟在它后面的动作步骤。

(2) 选择结构:提供了一种根据判断的不同结果,分别执行不同的后续操作的控制机制。

(3) 循环结构:通常包括循环控制条件和循环体。循环控制条件描述了循环反复执行的条件,而循环体则描述了每次循环如何对数据进行处理的动作(序列)。

已经证明,任何算法都可用这3种结构描述,即这3种结构是组织算法动作的最小集合。

算法规定的动作序列可由人或机器来执行。以求平方根方法为例,当人来执行时,首先,要能理解所描述的各动作的含义:第1步的“猜测”,第2步的“乘”和“足够逼近”,第3步的“ $(g+x/g)/2$ ”等运算,以及第4步“重复”等的含义。然后,在理解这些含义的基础上能做相应的动作,即能完成“猜测某个数并用符号 g 表示”“乘法运算”“加法运算”“除法运算”,以及“回到第2步开始执行”等动作。最后,要能够根据动作序列,自动化、机械地完成序列的执行,这包含两个方面的机制:一是记住了(在脑海中或在纸上)当前执行的动作以及知道下一步该执行哪一个动作,二是记住了(在脑海中或在纸上)中间结果(如不同时刻 g 的值)。

与此类比,如果能设计一台机器,该机器能像人一样“理解”动作的含义,“执行”相应的动作(即能实现乘、除、加、比较和重复等操作),能记住正在执行的和下一步要执行的动作序列,以及能记住相应的中间结果,那么,就能用这台机器代替人来进行求平方根的运算。

人类发明计算装置(包括计算机)最初目的是使加、减、乘、除等基本算术运算能够自动化,以便让其承担工程问题中枯燥、烦琐和大规模的计算,从而减轻人类的脑力劳动强度。但是,计算机的作用不仅限于此。在人类研究和开发计算装置和计算技术的进程中,逐步赋予了计算机逻辑操作能力,计算机不但能进行算术运算,而且能进行逻辑判断,使其可以根据问题的性质,执行不同的运算。人类发明精致的编码符号系统,用于将各种信息形式数字化,使计算机不仅仅能够处理数值信息,而且能够处理更广泛存在的其他形态的信息,如文本、图形、图像、声音和视频等。

1.2 计算装置发展简史

劳动创造工具,而工具又拓展了人类探索自然的深度和广度。计算机是人类对计算装置的不懈努力追求的最好回报。从原始的结绳记事、手动计算、机械式计算到电动计算,计算装置的发展经历了漫长过程。现代电子计算机的出现,才使计算装置有了飞速的发展。科学技术的进步促进了计算装置的一代代更新。计算装置的发展不仅得益于组成计算装置的元器件技术的发展,而且得益于对计算本质的认识的提高。

1.2.1 机械式计算装置

人类发明工具辅助处理信息的历史可追溯到远古。在古代中国,原始部落人就发明了“结绳记事”的方法,即将一根绳子打结来记载曾经发生的事件。绳结的形状、位置、数目和颜色等属性可以不同,因而可以表示不同的事件。在一根绳子上,由下到上可以打多个绳结,每个绳结对应一个事件,绳结的顺序表示事件发生的顺序。绳结的颜色则可表示事件的性质,如红色表示幸事,白色表示不幸之事。绳结是一种记忆工具,通过观察绳结能使人联想曾经发生的事情。这种工具虽然简单,却有利于记忆的材料的基本特性。第一,可区分绳结的不同状态,即绳结的形状、位置、颜色和数目是可区分的;第二,可以人为设定绳结的状态,即可以打出不同的绳结;第三,一旦给定绳结的某种状态,在自然条件下,可以保持很长一段时间,除非外力将其改变。

算盘是古代中国发明的一种有效的计算工具。汉代已出现用珠子进行计算的方法,东汉的《数术记遗》一书有过记载。出现“算盘”术语者,以宋代《谢察微算经》为最早。可以确定最迟到宋代,横梁和穿档的算盘就已经出现。至元代,算盘的使用已十分流行。宋元代之间的刘因就写有《算盘》一诗。明代关于算盘的记载更多,如《瀛涯胜览》《九章详注比类算法大全》等。明朝初期,中国算盘流传到日本,其后又流传到俄国,又从俄国传至西欧各国,对近代文明产生了很大的影响。15 世纪中期,《鲁班木经》中载有制造算盘的规

格。算盘的材质以木头为主,其他有竹、铜、铁、玉、景泰蓝、象牙、骨等。算盘小者的可藏入口袋,大者要人抬。

算盘的结构如图 1-1 所示,它的四周有一个框架,框架中间嵌入数根纵杆,纵杆称为档,纵杆上串有数颗珠子。由一个横梁将珠子分隔为上下两个区域,分别称为梁上区和梁下区。梁上区中的珠子称为上珠,梁下区的珠子称为下珠。在一根纵杆上上珠或一颗,或两颗,下珠或 4 颗,或 5 颗。

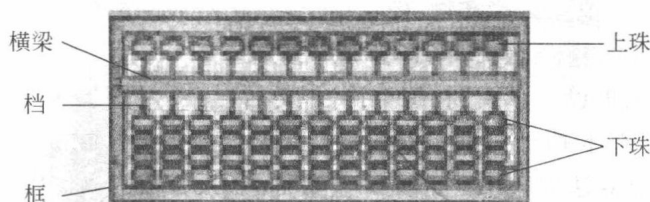


图 1-1 算盘

算盘珠子的数目和位置表示十进制数。下面以两颗上珠,4 颗下珠的算盘为例,说明如何来表示十进制数。一颗下珠表示数 1,两颗下珠表示数 2,...,4 颗下珠表示数 4;一颗上珠表示数 5,两颗上珠表示数 10。因此,算盘每档可表示的最大数是 14。当上珠依次挨在一起并贴着上框,且下珠依次挨在一起并贴着下框时,表示数 0。每往上拨动一个下珠,该档表示的数增加 1;每往下拨动一个上珠,该档表示的数增加 5。算盘中从右往左,档所表示的十进制数由低位向高位递进。一个算盘可同时表示多个十进制数,十进制数之间的区隔由人决定,可在横梁上分别标志为个、十、百、千、万等,这样每位数的位置相对固定;也可仅仅由人记住每位数的位置即可,这时每位数的位置是可变化的。一个算盘能表示多少个十进制数,由档的数目和十进制数的位数决定。

算盘是一种快捷方便的算术运算工具,珠算熟练者,快过用现代计算器进行计算。作为一种计算工具,它有如下特点。

(1) 具有表示数值的一套符号系统,这套符号系统由珠子数目和珠子的位置确定。

(2) 存在高效的运算法则,操作者按照运算法则,拨动珠子,实现快速运算。

(3) 短期记忆。算盘上暂时保存操作数和结果,且保存的数易于复写和改变。

(4) 手工操作,即操作过程没有自动化。虽然每个拨动珠子的操作是机械的,但需要人来完成。运算过程的脑力劳动有所减轻。

因此,算盘的快捷方便来源于高效的运算口诀和简易的拨动珠子操作。同时,它能够节省运算所需要的纸墨资源。

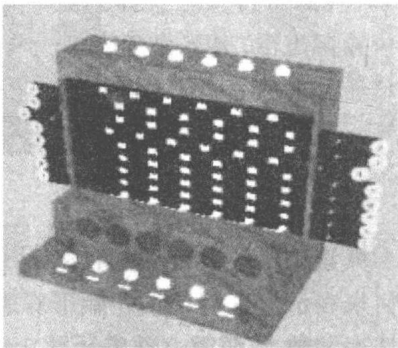
机械式计算装置大约出现在 17 世纪的西方国家。随着机械装置广泛应用于生产劳动中,人们开始设想发明一种机械装置来实现算术运算。最初的计算机械装置是粗糙的,但改变了完全依赖手工进行计算的状态,在计算自动化方面有了重要的起步,开始了初步的低级自动计算的历程。

有据可查的第一台机械计算器,是威尔海姆·舒卡德(Wilhelm Schickard)于 1623 年制造的。舒卡德 1592 年出生在德国的 Herrenberg,分别于 1609 年和 1611 年在 Tübingen 大学获得学士学位和硕士学位。1619 年被任命为图比杰大学的希伯莱语教

授,1631 年为天文学教授。1623 年他建造了一台能够做数学操作的机械装置。在给开普勒(Johannes Kepler)的一封信中,舒卡德如此描述他的机器:

“对于你所做的计算,我已经尝试用机械的方式做了。我设想了一台机器,它由 11 个完全和 6 个不完全的链齿轮组成。给定数值,它能即时和自动做加法、减法、乘法和除法。你将会高兴地看到,这台机器是如何累加一个大小为 10 或 100 的数,并自然将它向左传送,或向右传送,当作减法时它又是如何做相反的事情的。”

后人没有看到舒卡德计算器实物,但在他 1624 年给开普勒的一封信的附件中发现了设计草图。20 世纪,图比杰大学的布努诺·巴罗(Bruno Baron)教授根据设计草图,利用 17 世纪的钟表制造技术,重构了舒卡德计算器,如图 1-2 所示。舒卡德计算器在做加减法时,先在机器中将参加运算的数设置好,经过自动运算,在机器上的读数窗口能读出结果。而做乘法时,先利用乘法表实现乘数的每一位与被乘数的乘积,然后将这些部分乘积加起来,得到最终结果。



通常的机械计算装置都能执行加法和减法,有的还能完成乘法和除法。一般来说,机械计算器有下列特点。

- (1) 以某种机械的方式保存参加运算的数及结果。
- (2) 用齿轮作为自动运算的装置。
- (3) 运算法则固化在机械中,以机械运动实现运算。

查尔斯·巴贝奇(Charles Babbage)及巴贝奇的机器分析机(A analytical Engine)在计算装置发展史上占有重要的地位。巴贝奇生活的那个时代,蒸汽机的广泛使用极大地推动了工业生产的发展,各项工程都需要大量的数学计算,人们借助一种数学计算工具——数学函数表(从简单的加法和减法表,到复杂的对数函数和三角函数表),来提高计算的效率。但绘制数学函数表,需要花费大量的脑力劳动,而且不可避免产生大量的错误。人们开始设想能否发明一种由蒸汽机驱动的计算机器,将冗长枯燥的计算任务转移到机器上。1822 年,巴贝奇基于差分计算原理设计了一台差分机(Difference Engine),该机器可用于计算各种数学函数表。

1830 年,巴贝奇设计了分析机。分析机有 3 个主要部件:齿轮存储器、运算装置和控制装置。巴贝奇设想用穿孔卡片叠(Jacquard's Punched Cards)控制机器的计算过程,包括操作顺序、输入和输出等过程。该控制机制包含了顺序、选择和循环控制等特性。

奥古斯特·艾达·劳莉斯(Augusta Ada Lovelace)为分析机编写了一个程序,用来计算 Bernoulli 数序列。这是世界上为机器编的第一个程序。所以,艾达·劳莉斯也是世界上的第一个程序员。现代人为了纪念艾达·劳莉斯,将一种计算机程序设计语言命名为 Ada。

150 年之后,伦敦科学博物馆依照巴贝奇的设计图纸,用铁、铜和钢等材料制造了一