

高等学校计算机基础教育教材精选

# Python程序设计基础

鲁凌云 等编著

清华大学出版社

高等学校计算机基础教育教材精选

# Python程序设计基础

鲁凌云 等编著

清华大学出版社  
北京

## 内 容 简 介

本书是 Python 程序设计的入门书籍,将 Python 程序设计分为三大模块:面向过程的 Python 程序设计,面向对象的 Python 程序设计,以及 Python 程序设计综合实践题、模拟题和习题解析。面向过程的 Python 程序设计模块,介绍 Python 语言的入门基础知识,主要包括 Python 语言的技术起源、Python 语言的数据类型与表达式、Python 语言的流程控制语句,以及 Python 语言的函数设计方法。面向对象的 Python 程序设计模块,介绍 Python 的面向对象技术,包括继承、接口、封装的概念及实现,利用 turtle 库绘制图形,利用 NumPy 进行科学计算。Python 程序设计综合实践题、模拟题和习题解析模块,通过两个案例综合了前两个模块的重要知识点,通过两套模拟试卷夯实 Python 基础知识及核心技术,通过对每章的习题进行剖析,让读者尽快理解和掌握知识要点。本书所有案例都是基于 Python 3.0 版本以上的。

本书适合作为高等学校计算机通识类课程的教材,也可作为 Python 程序设计爱好者的入门书籍。

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

Python 程序设计基础/鲁凌云等编著. —北京:清华大学出版社,2019

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-52418-2

I. ①P… II. ①鲁… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 042149 号

责任编辑:龙启铭

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:沈 露

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

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

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:13.75 字 数:321 千字

版 次:2019 年 5 月第 1 版 印 次:2019 年 5 月第 1 次印刷

定 价:39.00 元

---

产品编号:064241-01

# 前言

Python 程序设计基础

Python 语言以其“语法简单、句式清晰、高效实现”等特点逐渐成为当今世界最流行的编程语言之一。随着我国不断加强人工智能(Artificial Intelligence, AI)发展战略, Python 语言已被广泛应用于人工智能产品的研发、行业大数据分析等各个领域。掌握必要的 Python 语言已成为新世纪人才具备的基础素质之一。本书是 Python 语言的入门教材,期望能够为初学者打下良好基础,为初学者开启一扇探索 Python 语言与行业有效结合的大门。本书具有以下特点。

## 1. 定位准确

本书主要是为非计算机专业学生进行 Python 程序设计学习而编写的,考虑到这部分学生的程序设计基础比较薄弱,因此,本书的学习目标主要是将程序设计与本专业相结合,通过大量示例讲述程序设计语言中的奥妙。

## 2. 注重实践

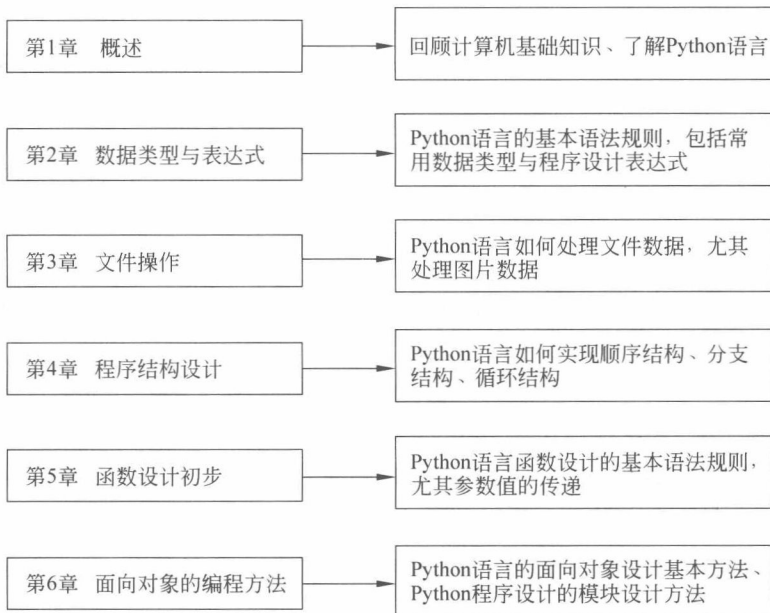
对非计算机专业的学生,在学习了必要的程序设计语法和规则后,更关注的是 Python 语言的实际应用,因此本书注重介绍使用 Python 语言编写程序来解决专业中可能遇到的工程问题。

## 3. 便于自学

本书由浅入深,通过大量的示例讲解,便于学生根据示例的提示,独立完成 Python 语言程序的编写与调试,有助于培养学生独立解决问题的能力,据此激发学生与本专业知识相结合的创新能力。

本书的篇章结构如下图所示。

本书由北京交通大学计算机学院鲁凌云任主编,诸强任副主编,张英俊、王瑞平两位教师负责了部分章节的编写。同时,也特别感谢北京交通大学熊柯教授、高博副教授、北京服装学院刘正东教授参与书稿的编写并提出宝贵建议。其中,鲁凌云负责编写第 1 章和第 2 章,诸强负责编写第 3 章和第 5 章,张英俊负责编写第 4 章,王瑞平、刘正东负责编写第 6 章。



与本书配套的教学资源，可登录“中国大学 MOOC(爱课程)”网站获取，链接如下：

<https://www.icourse163.org/course/NJTU-1003368009>

由于本书是为非计算机专业学生编写的教材，难易程度是否适合，还需要时间的检验，也欢迎广大读者对本书的各类错误批评指正。

编 者  
2019年3月

# 目录

<b>第 1 章 概述</b> .....	1
1.1 计算机工作原理 .....	1
1.1.1 计算机起源 .....	1
1.1.2 二进制数 .....	3
1.1.3 计算机内部结构 .....	4
1.2 计算机语言 .....	6
1.2.1 指令 .....	6
1.2.2 计算机语言发展史简介 .....	6
1.2.3 程序设计中的“变量”与“变量值” .....	7
1.3 Python 语言 .....	8
1.4 第一个 Python 程序 .....	9
本章小结 .....	10
习题 .....	10
<b>第 2 章 数据类型与表达式</b> .....	13
2.1 数据类型 .....	13
2.1.1 数值类型 .....	14
2.1.2 字符串类型 .....	17
2.1.3 布尔类型 .....	21
2.1.4 列表类型 .....	22
2.1.5 字典类型 .....	25
2.1.6 元组类型 .....	29
2.2 访问不同类型的数据 .....	31
2.2.1 Python 语言常用符号 .....	31
2.2.2 序列的操作 .....	36
2.2.3 指定函数对序列的操作 .....	38
2.2.4 字典遍历 .....	40

2.3	表达式与运算符	42
2.3.1	算术符号与算术表达式	42
2.3.2	关系符号与关系表达式	43
2.3.3	逻辑符号与逻辑表达式	43
2.3.4	位运算符与位运算	43
2.3.5	运算符的优先级	44
2.4	变量赋值与输出	45
2.4.1	直接赋值	45
2.4.2	input()输入方式	46
2.4.3	eval()函数	46
2.4.4	format()输出方式	47
	本章小结	49
	习题	49
<b>第3章</b>	<b>文件操作</b>	<b>51</b>
3.1	认识文件	51
3.1.1	文件名	51
3.1.2	文件类型	52
3.1.3	文件位置	52
3.2	文件的操作	53
3.2.1	文件的打开与关闭	53
3.2.2	读取文件	54
3.2.3	写入文件	57
3.3	图像文件和网络文件	59
3.3.1	图像文件的读写	59
3.3.2	图像文件的处理	60
3.3.3	网络文件的读写	66
	本章小结	72
	习题	73
<b>第4章</b>	<b>程序结构设计</b>	<b>75</b>
4.1	程序的基本结构	75
4.1.1	Python 程序结构概述	75
4.1.2	算法概述	76
4.1.3	算法的特点	76
4.1.4	算法的表示	77
4.1.5	程序的三种基本结构	80
4.2	程序设计中的表达式	83

4.2.1	Python 语言的关系表达式 .....	83
4.2.2	Python 语言的逻辑表达式 .....	83
4.3	分支语句 .....	84
4.3.1	单分支结构: if 语句 .....	84
4.3.2	二分支结构: if-else 语句 .....	86
4.3.3	多分支结构: if-elif-else 语句 .....	87
4.4	循环控制语句 .....	88
4.4.1	for 语句 .....	88
4.4.2	while 语句 .....	91
4.4.3	break 和 continue .....	92
4.4.4	程序的异常处理语句 .....	94
4.5	控制结构综合案例 .....	96
	本章小结 .....	99
	习题 .....	99
<b>第 5 章</b>	<b>函数设计初步 .....</b>	<b>102</b>
5.1	函数定义 .....	102
5.1.1	程序设计函数的起源 .....	102
5.1.2	函数的定义 .....	102
5.1.3	匿名函数 .....	104
5.2	函数的参数传递 .....	105
5.2.1	按照位置传递参数 .....	105
5.2.2	按照关键字传递参数 .....	105
5.2.3	按照默认值传递参数 .....	106
5.2.4	可变数量的参数传递 .....	106
5.3	函数的返回值 .....	110
5.3.1	返回布尔值和列表的函数 .....	110
5.3.2	无返回值的函数 .....	111
5.3.3	返回多值的函数 .....	112
5.4	变量的作用域 .....	113
5.5	递归 .....	115
5.5.1	递归的定义 .....	115
5.5.2	递归实例 .....	117
	本章小结 .....	118
	习题 .....	119
<b>第 6 章</b>	<b>面向对象的编程方法 .....</b>	<b>123</b>
6.1	面向对象基础知识 .....	123



6.1.1 对象与面向对象·····	123
6.1.2 类·····	124
6.1.3 面向对象的程序设计·····	126
6.2 利用 turtle 库绘制图形·····	131
6.3 Python 科学计算·····	135
6.3.1 NumPy 处理数据·····	136
6.3.2 Matplotlib 绘制图表·····	141
6.3.3 SciPy 数值计算库·····	147
本章小结·····	156
习题·····	156
<b>第 7 章 综合训练题</b> ·····	<b>159</b>
模拟题一·····	161
模拟题二·····	165
<b>附录 各章参考答案及解析</b> ·····	<b>169</b>

# 第 1 章 概 述

## 1.1 计算机工作原理

### 1.1.1 计算机起源

中文名词“计算机”来源于英文 computer。英文单词 teacher(教师)、worker(工人)都是指人,透过 computer 英文单词的表面意思,还可以理解为“能够计算的人”。computer 这个词大约在 19 世纪 40 年代提出,也许那个时候的科学家有一种美好的愿景,希望有某种机器能够模仿人的大脑,像人的大脑一样分析问题、处理问题。这里我们有必要先了解一下大脑的工作过程,例如,计算“ $8+6\div 2=?$ ”,如图 1-1 所示。人的眼睛看到这个数学表达式,经过分析,会通过嘴说出答案 11。首先,将这个表达式记在大脑中,再经过脑神经元的思考,结合数学知识,先算出“ $6\div 2=3$ ”这一中间结果,然后再给出“ $8+3=11$ ”这一最终结果,11 这一结果将会保存在某个神经元。我们可以把这一结果通过声音告诉其他人,也可以写在纸上告诉别人,还可以记在脑海中不告诉其他人。通过做这一简单运算题,就可发现一条规律:首先,通过眼睛等感觉器官将捕捉的信息输送给大脑并存储起来,然后,对这一信息进行加工处理,最后,由大脑控制人把最终结果以某种方式表达出来。

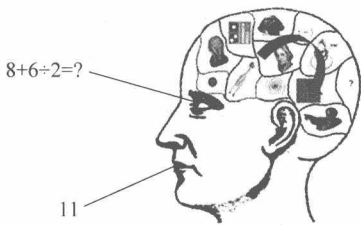


图 1-1 大脑的工作过程

是否可以设计一种机器来模拟或代替人脑的部分功能? 计算机之父阿兰·图灵、控制论之父诺伯特·维纳和数字计算机鼻祖冯·诺依曼从不同的角度证明了这一假设的可能性与可行性。而且他们还从不同的视角预言,计算机可以模拟甚至超越人脑的某些能力(例如判断、推理、记忆等能力)。

阿兰·图灵是一名数学家(见图 1-2)。他发表了一篇经典论文《论可计算数及其在判定问题中的应用》。阿兰·图灵提出了“程序控制”思想,阐明了“有一种机器,也能像人脑一样执行指令序列”,并且给出了一种计算模型,即著名的图灵机(Turing Machine)模型。这种假想的机器由一个控制器和一个两端无限长的工作带组成,工作带被划分成一个个大小相同的方格,方格内记载着给定字母表上的符号。控制器带有读写头并能在工作带上按要求左右移动。随着控制器的移动,其上的读写头可读出方格上的符号,也能改

写方格上的符号。这种机器能进行多种运算,并可用于证明一些著名的定理。这就是最早给出的通用计算机模型。

诺伯特·维纳(见图 1-3)从控制与通信的角度论述计算机的实现问题。在《控制论》一书中,维纳认为计算机的设计应该遵循下述原则。



图 1-2 阿兰·图灵



图 1-3 诺伯特·维纳

- (1) 计算机的核心部分,加法和乘法装置应当是数字式的。
- (2) 加法和乘法采用二进制,实现这种方式比十进制更经济,可以由电子管完成。
- (3) 从把数据放入机器到取得结果,全部运算序列应自动进行,中间没有人的干预。
- (4) 运行中涉及的逻辑判断应由机器自身完成。
- (5) 机器应包括存储数据和控制命令的装置。

维纳建议美国政府研制数字计算机,得到采纳后,成立以冯·诺依曼负责的电子数字计算机研究课题组。1946年,世界上第一台电子计算机 ENIAC 问世。今天的计算机也被称为冯·诺依曼计算机。当时,研究和开发 ENIAC 计算机的目的是为军事服务,主要是为了计算弹道和火力表。随着 ENIAC 的诞生,人类拉开了计算机发展的序幕。

冯·诺依曼在实现计算机的过程中,也汲取了图灵和维纳的思想。运算器采用了二进制方法实现,电子器件采用电子管。ENIAC 由 18800 个电子管组成(如图 1-4 所示),利用电子管实现了计算机的核心部件 ALU,每秒钟可完成 5000 次加减法或 400 次乘法运算。

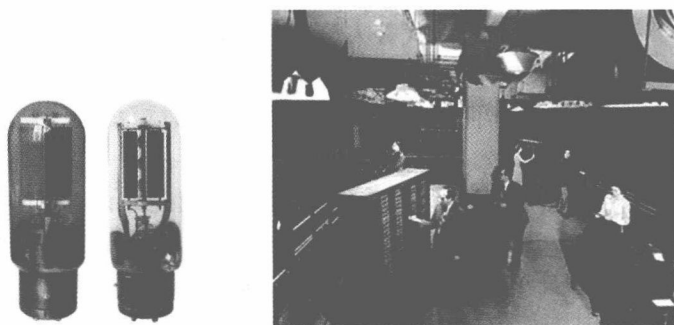


图 1-4 电子管与世界上第一台计算机 ENIAC

但是电子管体积大,耗电多,容易损坏,例如,ENIAC耗电大约 150kW。因此,随着电子技术的不断发展,计算机的核心部件逐渐用晶体管、集成电路等电子元器件替代。从第一台计算机诞生至今,计算机都是由电子器件实现,因此,我们目前使用的绝大多数计算机称为“电子计算机”。

随着科技的突飞猛进,人类对计算机的认识不断深入,除了“电子计算机”,未来可能还会出现各种物理形式的计算机。目前,科研工作者正不断探索“量子计算机”“生物计算机”“光子计算机”等。

### 1.1.2 二进制数

计算机是电子设备,利用计算机可以处理数字信号的数据,也可以处理模拟信号的数据。数字信号与模拟信号的区别如图 1-5(a)和图 1-5 (b)所示。我们不难发现,数字信号的最大特点是容易实现,只要我们事先设定一个阈值,大于这个阈值,我们认为这个信号表示“1”,否则认为是“0”。利用电子元器件较容易实现数字计算机,而实现模拟计算机需要较高精确度的电路结构,是比较困难的,因此“模拟电子计算机”在计算机的发展史上稍纵即逝,很快被淘汰。我们目前使用的绝大多数计算机都是“数字电子计算机”。

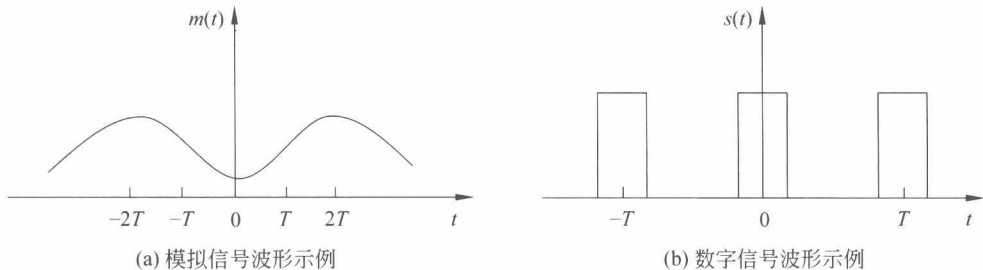


图 1-5 模拟信号与数字信号

数字计算机的电路只有两种工作状态,即开或关。为了表示这两种工作状态,常常用 1 表示“开”,用 0 表示“关”,即计算机硬件用一系列的高低电平表示“0”或“1”。在计算机内部,由于只能处理“0”或“1”组合的信息,因此无论是用户数据,还是控制这些数据的“命令”,都需要用二进制数表示。

二进制数只有两个数字“0”或“1”,按照“逢二进一”的原则计数,即每位计满 2 时向高位进 1。例如,二进制的数值“100”对应十进制的数值“4”,十进制的数值“10”对应二进制的数值“1010”。一连串的二进制数,例如“1000110010001100”,不但可以表示数值,也可以表示计算符号、图片的像素。那么,计算机存储二进制数的最小单位就是“比特”(bit),简称为 b。字节(Byte)是计算机处理数据的基本单位,简称为 B。比特与字节之间的关系如图 1-6 所示。

计算机利用“字节”的方式存储指令和数据。计算机“指令”就是指挥计算机工作的命令,一系列按一定顺序排列的指令就构成了程序。数据好像是“士兵”,而指令更像是“指挥官”。数据的一举一动,都要服从“指挥官”的命令。不同的指令占用的字节数不同;不



图 1-6 比特与字节之间的关系示意图

同类型的数据,所占的字节数也不同。例如,对于 Python 指令“a=3”,其中“3”是数据,计算机内部将会分配 2 字节存储整数 3;对于 Python 指令“a='C'”,其中“C”是数据,计算机内部将会分配 1 字节存储字符 C;对于 Python 指令“a=3.58”,其中“3.58”是数据,计算机内部可能会分配 4 字节存储小数 3.58,如图 1-7(a)~图 1-7(c)所示。给变量分配的字节数目的大小,与编译器、操作系统等编译环境有关,不同的编译环境可能会为同一数据类型分配不同的字节数。

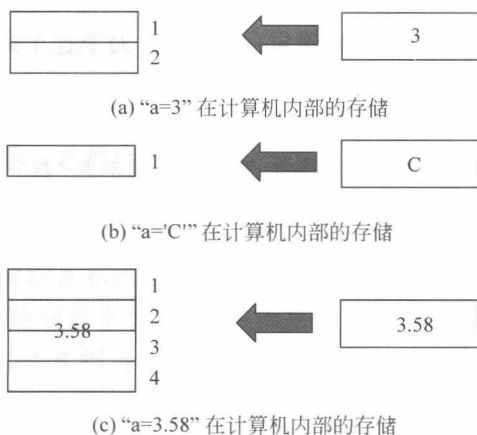


图 1-7

### 1.1.3 计算机内部结构

计算机是如何利用存储的指令和数据工作的?这就涉及计算机内部结构。目前,大多数的计算机结构延续的还是冯·诺依曼设计的结构。冯·诺依曼(如图 1-8)提出了“将计算机要处理的程序和数据先放在存储器中,在计算机的运算过程中,由存储器按事先编好的程序,快速地提供给微处理器进行处理,在处理过程中不需要用户干预”的原理,而计算机之所以能够高效工作,就是基于存储程序和程序控制这个原理。



图 1-8 冯·诺依曼

计算机主要包括五部分:输入设备、输出设备、存储器、运算器、控制器,如图 1-9 所示。输入设备就像人的眼睛和耳朵一样,负责传递各类数据。输出设备就像人的嘴一样,负责传递大脑思考分析后的数据。存储器、运算器和控制器配合起来工作,就像人的大脑一样,可以分析、判断问题。运算器在处理问题时,如果

问题较复杂,需要一条条分析、推理或是判断。控制器负责发送各类存取数据的命令。我们不难发现,“数据什么时候进入计算机”“什么时候数据被处理”“什么时候把处理结果显示出来”,所有操作都与计算机的控制器密切相关。冯·诺依曼设计计算机的过程中,把“运算器”“控制器”“寄存器”封装成计算机的核心器件,即中央处理器(Central Processing Unit,CPU)。

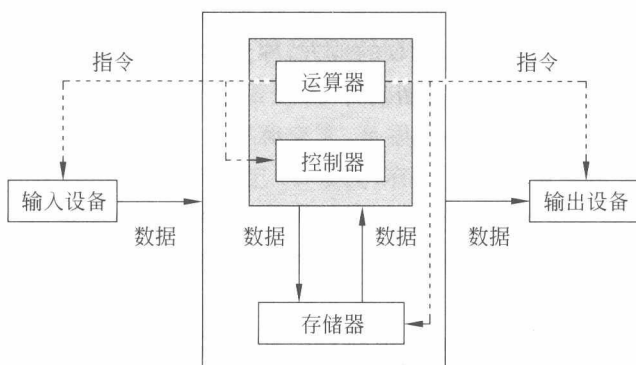


图 1-9 冯·诺依曼体系结构

## 1. 运算器

运算器是任何计算机的核心设备之一,其作用就是用来进行算术运算和逻辑运算,是计算机的主体。在控制器的命令下,运算器接收要运算的数据,完成程序指令指定的算术运算或逻辑运算。

## 2. 控制器

控制器是计算机的指令控制中心,用来分析指令、协调 I/O 操作和内存访问。控制器从存储器中逐条取出指令、分析指令,然后根据指令要求完成相应操作,产生一系列控制命令,使计算机各部分自动、连续并协调地工作,作为一个有机的整体,实现数据和程序的输入、运算并输出结果。

## 3. 存储器

存储器是用来存储程序、数据、运算的中间结果及最后结果的设备,计算机中的各种信息都要存放在存储设备中。根据存储设备在计算机中处于的不同位置,可分为主存储器(也称为内存储器,简称为内存)和辅助存储器(也称为外存储器,简称为外存)。从存储介质构成原理角度,可分为磁表面存储器、半导体存储器、光介质和磁光介质存储器等。

在计算机内部,直接与 CPU 交换信息的存储器称为内存,用于存放计算机运行期间所需的信息,如指令、数据等。外存是内存的延伸,其主要作用是长期存放计算机工作时所需要的系统文件、应用程序、用户程序、文档和数据等。当 CPU 需要执行某部分程序和数据时,由外存调入内存以供 CPU 访问。可见,外存用于长期保存数据和扩大存储系统容量,主要有磁盘、磁带或 U 盘。由于 U 盘具有存储容量大、价格低廉、性能好等特点,

已成为目前微型计算机最常用的移动存储设备。

如果说 CPU 是计算机非常重要的一个部件,那么存储器就应该是计算机特别重要的一个部件。输入到计算机的数据放在什么地方? 处理这些数据的命令被放在哪里? 数据处理后的结果又放在什么地方? 这些问题都需要在计算机设计过程中认真考虑。存储器的任务艰巨,工程师们把存储器分成了三种不同的类型,承担着不同的存储任务。一种是暂时存储器,常常称为计算机的内存;一种是 CPU 内部的存储器,称为寄存器(register);一种是长期保存数据和命令的地方,称为外存。当计算机关机时,内存不会长期保存数据,因此需要把数据存放在外存中。

#### 4. 输入设备

输入设备是用来完成数据输入功能的部件,即向计算机输送程序、数据以及各种信息的设备。常用的输入设备有键盘、鼠标、扫描仪、U 盘、磁盘和触摸屏等。

#### 5. 输出设备

输出设备是用来长期保存数据的,即将计算机工作的中间结果或最终的处理结果从内存传送到外存的设备。常用的输出设备有显示器、打印机、绘图仪、U 盘以及磁盘等。

## 1.2 计算机语言

### 1.2.1 指令

指令是指挥计算机完成某个操作的命令,发出的指令要能被计算机的输入设备、输出设备、存储器、运算器、控制器理解并执行。设计好的计算机硬件是为了更好地执行“指令”。为解决某个问题而设计的一系列有序指令的集合称为程序,而程序是用某种“计算机语言”描述的。计算机的工作过程就是根据程序处理数据,得到结果。若干段程序和相关的文档集合就构成了软件。软件可实现比较完整的一个功能,如图 1-10 所示。

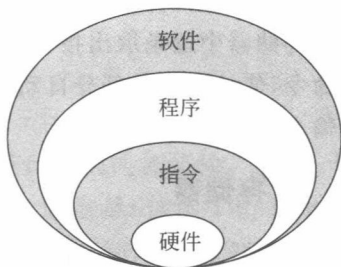


图 1-10 计算机软件层次结构

### 1.2.2 计算机语言发展史简介

计算机语言是人与计算机之间交流的语言。第一台计算机 ENIAC 直接识别并执行人类编写的指令,人类使用“机器语言”与 ENIAC 进行交流。机器语言很难掌握,不具备通用性,而且,机器语言是整个计算机体系中最深奥的接口,是硬件和软件直接的中间件。

汇编语言的出现大大提高了人们控制计算机的热情,利用汇编语言可以解决很多问题。汇编是一种利用一些特定助记符表示指令的语言。与机器语言相比,汇编语言比较

容易掌握。但计算机的硬件只能处理二进制数,汇编语言最终要转换成二进制。“0”和“1”的世界,人类是不熟悉的,因此需要编译或者解释工作。我们把汇编语言翻译成机器语言的过程称为编译。但是,只有程序员对计算机的工作流程很熟悉,才能使用汇编语言编写程序,对程序员的要求较高,从某种程度上也限制了计算机的普及和应用。

为了让更多的人更容易控制计算机,产生了高级的计算机语言。伴随着人类进步和计算机技术的进一步发展,计算机的各类高级语言正帮助人们解决各式各样的问题。高级语言是一种更接近人们对求解过程或问题描述的计算机语言。这种语言允许用英文编写程序,程序中所使用的运算符或者表达式,都与我们日常用的数学表达式十分相似。高级语言容易学习,通用性强,便于推广和交流,是很理想的一种程序设计语言。高级语言发展于20世纪50年代中叶到20世纪70年代,有些流行的高级语言已经被大多数计算机厂家采用,固化在计算机的内存里。例如,C、FORTRAN、C++、Java、Python等都属于高级语言。相对于高级语言,机器语言和汇编语言又称为低级语言。

### 1.2.3 程序设计中的“变量”与“变量值”

利用计算机的某种语言可以实现某种功能,称为“程序设计”。程序设计的目的是帮助处理数据,并得到理想的结果。那么,输入的数据是千变万化的,就像一道数学题“两个数相加的得数是多少?”可以设计为表达式“ $Z=X+Y$ ”,未知数 $X$ 和 $Y$ 可以填写很多种可能的数据。根据 $X$ 和 $Y$ 的值,可以得到 $Z$ 值。计算机中的变量(variable)是一个抽象概念,可以理解为允许存放数据的空间。计算机程序设计中的表达式“ $Z=X+Y$ ”涉及3个变量,即计算机要准备“3块空间”来存放数据。

我们这个世界,从沙粒到太阳,从原生物到人,都处于永恒的产生和消灭中,处于不断的流动中,处于无休止的运动和变化中。当然,这其中包括了计算机,计算机如何“流动”,很大程度依靠计算机指令。在计算机的内部,如果要存储一些数据,就要准备存放数据的空间。当声明一个变量,计算机就会分配空间来存放数据。变量可以通过“变量名”访问。在表达式“ $Z=X+Y$ ”中, $Z$ 、 $X$ 、 $Y$ 为变量名。放在变量里面的数据可以是数值、字符或一串字符等。

变量里面能放什么值,还要看计算机语言的规则。例如,计算机的C语言要求在使用变量前,必须先声明变量类型,也就是说,类型不同,给变量分配的存储空间也不同。但Python语言很简单,用户把什么类型的数据存入计算机的空间,变量就是什么类型的。在程序中使用变量,一般需要三个步骤。

(1) 声明变量:就是给这块能够存放变量值的存储空间起名。但是划分多大空间需要看存储值的类型,即数据类型。在C语言中,一定要先声明变量,但Python却可以省略了这一步。这是Python语言与C语言的不同之处,Python不需要声明变量的类型。C语言是事先为变量定义好一个数据类型,根据这个数据类型的特点分配存储空间。例如,Python程序中如果定义 $a=5$ ,那么 $a$ 就是整型数值;如果定义 $a=4.5$ ,那么 $a$ 就是浮点型数值;如果定义 $a='Beijing'$ ,那么 $a$ 就是字符串类型数值。

(2) 变量赋值:使用赋值符号“=”,相当于从右到左的标识符“←”,也可以这样理



解,把右侧的数值给左侧的变量。例如, $a=3$ 就是把整数 3 存放在变量  $a$  中。还可以利用函数给变量赋值,例如, $a=input()$ 把从键盘输入的字符存放在变量  $a$  中。这里, $input()$ 是一个标准函数。

(3) 使用变量:计算机中存放了数据,利用设计的程序使用和控制数据。例如,对于程序语句  $a=3$  和  $a=10-a$ , $a$  的初始值是 3,运行语句  $a=10-a$  后,变量  $a$  的数值为 7。因此,也可以理解为,变量值可根据不同的控制语句发生改变。

**【例 1-1】** 一个罐子装了 10 颗糖,每天吃 1 颗糖。3 天以后,糖的数目是多少? 描述计算机处理这个过程如图 1-11 所示。

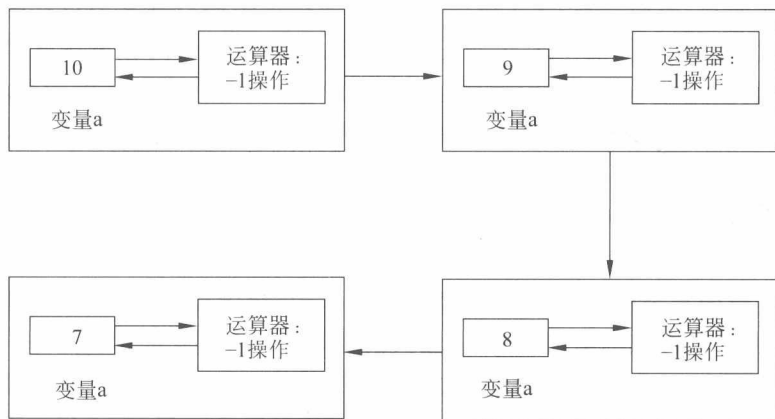


图 1-11 变量的存储值变化情况

### 【分析】

① 假设一个罐子相当于变量  $a$ ,运行表达式  $a=a-1$ 。最开始存放一个整数 10,第一天,减 1 后,将结果 9 仍然存放在变量  $a$  中。第二天,9 减 1 后,将结果 8 仍然存放在变量  $a$  中。第三天,8 减 1 后,将结果 7 仍然存放在变量  $a$  中。

② 最后变量  $a$  内存中存放的数值是 7。

## 1.3 Python 语言

Python 语言属于计算机高级语言,它功能强大,对程序设计初学者来说易于掌握。Python 语言是由荷兰人 Guido Van Rossum 于 1989 年创造。他用英国喜剧 *Monty Python and the Flying Circus* 中的“Python”命名了这门语言。Python 作为一门程序设计语言主要具有如下特点。

### 1. Python 语言能够减少代码量

Python 语言是当今世界最灵活和易用的编程语言。Python 可减少很多编写、调试和维护工作。使用 Python 进行程序开发,可以数倍地提升编码效率,不到 C、C++ 或 Java 一半的代码行数,将大幅度减少开发过程和维护阶段的工作量。与其他语言比较,