

学术顾问：马玉慧 李 粤 樊 磊

总 主 编：梁森山 谢作如

副 总 主 编：夏正仁 于方军

本 册 主 编：夏正仁

本册副主编：梁森山

主要编写人员：(按姓氏拼音排序)

陈建林 陈瑶瑶 傅海涛 管 庆 雷 鸣 梁光福 林森焱
刘啸宇 刘正云 陆雅楠 罗 亮 祁荣宾 邱奕盛 申劲红
王怡婷 杨璐璐 应根球 于旭珩 俞 晓 张敬云 张天辉
郑 祥 周 鹏 周 琼 周源远

版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

信息科技·七年级上册/夏正仁主编. —北京：清华大学出版社，2023.9(2024.8重印)
ISBN 978-7-302-62273-4

I. ①信… II. ①夏… III. ①计算机课—初中—教材 IV. ①G634.671

中国国家版本馆 CIP 数据核字(2023)第 007089 号

责任编辑：赵轶华

封面设计：王 静 薛玉斌 张思宇

责任校对：李 梅

责任印制：杨 艳

出版发行：清华大学出版社

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

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-83470000 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

印 装 者：重庆升光电力印务有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：7 字 数：118 千字

版 次：2023 年 10 月第 1 版 印 次：2024 年 8 月第 2 次印刷

定 价：6.47 元

产品编号：092274-02

前言

人类已进入全球化信息时代，信息科技作为现代科学技术领域的重要组成部分，对全球经济、社会和文化发展起着越来越重要的作用。信息科学所蕴含的思维方式改变了人们对虚拟世界和现实世界的认知方式，信息技术衍生出的虚拟世界重塑了人们沟通和交流的时空观念，深刻影响了人们的生活、学习和工作方式。信息素养已成为信息社会公民不可或缺的基本生存技能，无论从哪种意义上来说，信息素养的培育都势在必行。

党的二十大报告指出：全面贯彻党的教育方针，落实立德树人根本任务，培养德智体美劳全面发展的社会主义建设者和接班人。为落实新时代教育根本任务，满足社会发展对全体国民素质和人才培养的新要求，做好义务教育教材与高中新课标教材的衔接，我们编写了本套教材。

本套教材依据《义务教育课程方案（2022年版）》《义务教育信息科技课程标准（2022年版）》《普通高中信息技术课程标准（2017年版2020年修订）》的精神进行编写，全面落实党的育人方针，聚焦信息科技学科核心素养，借鉴了创客教育、STEAM教育、机器人教育多年的教学成果，积极创设真实的活动化、生活化、游戏化学习场景，以主题式项目学习组织知识、实验、活动与实践，旨在培养学生的创新思维和实践能力，提升他们的信息素养。

本套教材从培育与发展中小学生学习素养出发，采用多种策略适应中小学生的学习和认知特点，围绕“数据”“算法”“网络”“信息处理”“信息安全”“人工智能”六条逻辑主线，结合学生身边的事例、应用和真实情境，以项目式学习方式为主线，全面诠释了义务教育阶段学生应该了解和掌握的信息科技学科知识和应该具备的学科思维。

本册教材通过“体验活动”“实验活动”“实践活动”等多种学习活动，将学科知识、科学原理、问题解决方法、学科思维和对社会的影响系统地

融入其中。另外，教材创新性地选用了优质的国产自主知识产权平台、编程环境、开源硬件和配套资源等作为项目实践活动的支撑，从小处、细节入手，培养学生用国产、爱国产的文化自信。教材中所选项目和案例力求做到源于真实问题，在引领学生树立民族自豪感的同时，逐渐使他们养成负责任地使用信息科技解决实际问题的习惯，为今后的学习发展打下坚实的基础。

按照教育部印发的《中小学教材管理办法》等文件的要求，本套教材在具体编写过程中，特别强调中国在信息科技和信息化方面所取得的巨大成就，介绍信息科技在社会发展和解决重大问题中的核心作用，注重并强化信息科技学科的德育价值，引导学生把党、祖国、人民装在心中，强化学生做社会主义建设者和接班人的思想意识。

由于信息科技处在一个高速发展的过程中，中小学信息科技课程和教材也会随着教学实践与改革的深入而不断完善。鉴于编者水平有限，教材中难免存在不足之处，在此我们诚恳地希望广大师生给我们提出宝贵意见和建议，我们将及时予以修订。非常感谢每一位教材编写者所付出的心智与辛劳。

编 者

2024 年 1 月

目 录

第 1 单元 认识计算机	1
第 1 节 计算机的前世今生	4
第 2 节 计算机硬件	14
第 3 节 计算机软件	26
单元小结	37
 第 2 单元 算法及其实现	39
第 1 节 问题与算法	42
第 2 节 人机的互动	50
第 3 节 信息的处理	58
第 4 节 决策的实现	68
第 5 节 循环的艺术	75
第 6 节 指令的复用	85
第 7 节 算法的实践	96
单元小结	103
 附录	105

教材使用说明

本册教材供七年级第一学期使用，共分为 2 个单元，总计 16 学时。

本套教材设计了知识、活动和项目三条主线，这三条主线既彼此独立又相互融合。

(1) 知识主线主要从以下几个方面进行讲解。

【学习导引】结合学生生活实际或教学需求，简要介绍本单元将要学习的内容。

【正文】按照学科内在逻辑，系统性地阐述知识。

【拓展阅读】拓展延伸与本主题相关的信息科技知识，旨在拓宽学生的视野，提高学生探索未知的兴趣。

【概念解释】解释一些特定的专业术语，以帮助学生理解。

【知识回顾】以思维导图的形式呈现知识结构，帮助学生梳理学习内容。

(2) 活动主线主要包括以下几个方面，穿插在知识主线中。

【体验活动】需要让学生了解、但实施难度较高的活动。时间安排比较灵活，尽可能在课内完成，借助现有器材、设备、环境等进行体验，不需要过多的额外准备。

【实验活动】能够让学生探究的活动。相对正式，实验目标清晰体现学科核心素养或者关键知识点。

【实践活动】具有一定复杂度、综合性、学以致用和技术操作活动，是知识内化之后的呈现，是创客式学习或 STEAM 跨学科学习的典型形式。

【问题讨论】提出开放性的问题，引导学生深入思考并开展讨论交流。

(3) 项目主线主要包括以下几个方面。

【项目情景】设置一个具体的项目情景，并在这个情景下提出问题。

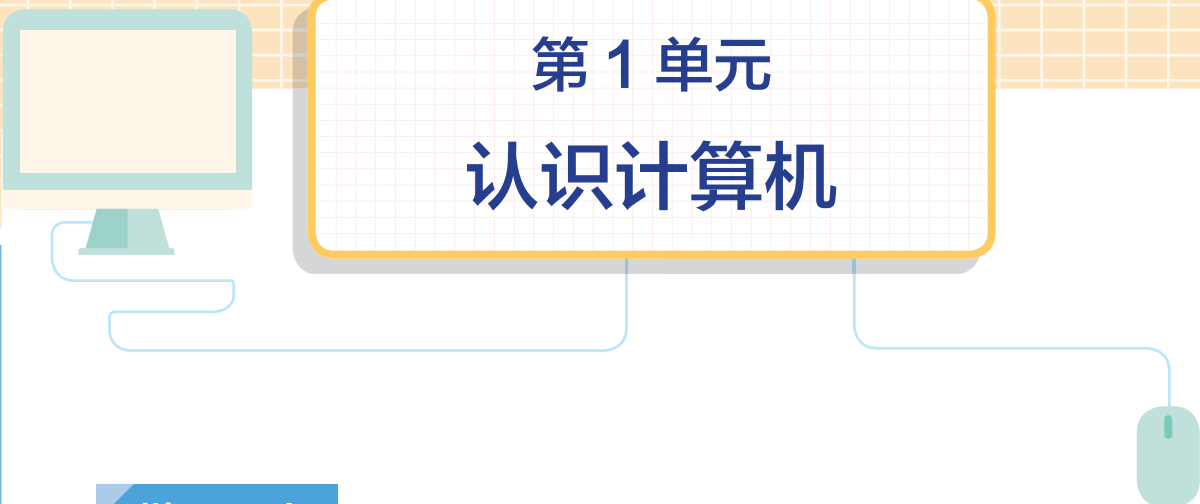
【项目方案】给出本单元项目中要用到的相关知识，并根据项目的目标与任务，对项目进行规划，给出主要的实施步骤，引导学生体验、动手操作或者进行探索实践，列出预期的实践成果。

【项目分工】引导学生根据任务内容进行分工，和与自己有共同想法的同学组成一组，并选出组长，列出每个人的具体分工。

【项目实施】引导学生结合项目内容进行项目实施。

【成果交流与评价】引导学生进行成果交流并学会分享，进行自评与他评。

在学习知识主线之前，先完成【项目情景】【项目方案】【项目分工】；【项目实施】穿插在知识主线中，【成果交流与评价】在单元最后。



第1单元

认识计算机

学习导引

计算机堪称是人类最伟大的发明之一，它在我们日常的学习和生活中发挥着越来越重要的作用，它是我们学习上的良师、工作中的助手、娱乐活动里的玩伴。但对很多人来说，计算机是熟悉而又陌生的小伙伴。虽然天天看到并且经常使用，但是并不了解其内部结构和运行原理。

在这个单元，我们将从身边的计算机开始，了解它的过去、现在与未来，探知它的内部结构与运行原理，以便更好地使用计算机。

项目情景

小清家很早就拥有包括台式机、平板电脑与智能手机等在内的多种类型的计算机，并用它们来开展社交、娱乐、学习与工作等各种活动。小清经常需要自行升级安装相关软件。但在升级安装实践过程中，他遇到了以下问题。

- (1) 不清楚计算机的硬件配置及其性能是否满足新软件的基本要求。
- (2) 不知道下载安装软件的安全途径。
- (3) 不知道有哪些可替代的非商业软件。

.....

让我们和小清一起，给他家的计算机设计一个合理的软件升级方案吧！

项目方案



经过咨询与了解，小清设计了以下方案。

知 识 学 习	实 施 步 骤	预 期 成 果
(1) 学习教材中相关知识 (2) 通过网络了解更多软硬件相关的知识	(1) 分析计算机硬件状况与当前对软件的升级安装需求 (2) 撰写软件升级方案 (3) 软件升级安装实践 (4) 撰写项目报告	(1) 软件升级安装需求 (2) 软件升级限制说明 (3) 软件升级方案 (4) 项目报告

你对小清的项目方案有什么不同的看法或建议？你准备如何设计项目方案？请填写在下表中。

知 识 学 习	实 施 步 骤	预 期 成 果

项目分工

方案设计完成后, 小青发现仅凭一己之力很难完成这个项目, 于是决定邀请对此问题感兴趣的同学一起参与, 并在项目方案中增加了以下表格。



姓名	角色	分 工	任 务
小青	组长	负责提出项目需求, 并主持整个项目实施活动	项目整体规划设计, 项目实施过程的统筹、协调、监督与总结, 制作项目汇报 PPT、对外展示交流等
同学甲	成员	负责分析软件的升级安装需求与升级限制说明	从功能性、安全性与经济性等角度, 分析应用程序的安装需求, 研究原有硬件配置对软件的升级限制情况
同学乙	成员	负责撰写应用程序升级方案与开展实践探究	根据软件的升级安装需求, 研究软件的获取途径与安装方法, 并开展升级安装实践
同学丙	成员	负责撰写项目实施报告	记录、整理项目实施的相关过程, 撰写项目实施报告等

你认为小青项目组的成员构成、分工和任务分配是否合理? 请在下表中填写你的项目分工情况。

姓名	角色	分 工	任 务

第 1 节 计算机的前世今生

本节知识

- ◆ 常见的计算机
- ◆ 计算工具的发展史

本节活动

- ◆ 壁挂式一体机优势的分析
- ◆ “不一样的计算机”的讨论
- ◆ 分享计算机发展史上的经典发明

计算机 (computer) 是人类的伟大发明之一。它能够实现数据计算、信息处理与人工智能等, 将人类从繁重的脑力劳动中解放出来, 并直接促进了信息社会的形成。我们回顾历史会惊讶地发现, 人们研发计算机的初衷竟然只是为了满足自动计算的需求。一个原本简单的计算工具, 却掀起一场让人类天翻地覆变化的信息革命。

一、形形色色的计算机

现今计算机已广泛地应用于各行各业, 人们的工作、学习和生活都越来越依赖计算机。不同的行业、岗位都会有不同的需求, 因此出现了很多形态各异、功能不一的计算机。在我们的日常生活中, 接触较多的通常是以下几类计算机。

1. 台式计算机

这类计算机体积相对较大, 通常需要放置在专用的办公台面上, 所以被形象地称为台式计算机, 简称为台式机。台式计算机虽然出现较早, 但目前仍然有较高的使用率, 无论是教室、机房还是办公室, 到处可见它的身影。

如图 1.1.1 所示, 按照主机与显示器是否分离这一特征, 台式计算机主要分为分体台式机与一体台式机两大类。人们在一体台式机基础上进一步发展出桌面一体机与壁挂式一体机, 以满足人们在不同环境中的使用需求。



图 1.1.1 台式计算机

问题讨论

壁挂式一体机广泛的应用于会议室、教室等各种公共场所。选用这种形式计算机有什么优势？

2. 笔记本电脑

笔记本电脑又称为便携式电脑、手提电脑等。它是一种既方便携带又具有台式计算机特点的计算机，其机身轻巧得像“笔记本”一样，还具备自带电源的特点，非常适用于移动办公。如图 1.1.2 所示，超轻、超薄是笔记本电脑追求的发展方向，但价格也因此变得昂贵。根据用途与使用对象的不同，人们将其细分为上网笔记本、商务笔记本、家用笔记本等。



图 1.1.2 笔记本电脑

3. 手持设备

手持设备又称为移动设备或口袋电脑，它是一种比笔记本电脑还要轻巧的计算机。如图 1.1.3 所示，我们所熟知的平板电脑、智能手机、电子书阅读器、掌上电脑等都属于手持设备。虽然手持设备的性能远不如台式计算机和笔记本电脑，但日益发达的移动互联网可以弥补其性能方面的不足，故手持设备逐渐

成为人们学习、办公与生活的得力助手。据统计，当前全球超过一半的用户通过手持设备接入移动互联网。



图 1.1.3 常见的手持设备

拓展阅读

计算机的分类

计算机的分类方法有很多。按照用途与适用范围的不同，可以分为通用计算机与专用计算机两大类。通用计算机的适用范围广泛，能够计算和处理多种问题，是日常生活中最为常见的计算机，如学习、办公或娱乐时使用的各种计算机。专用计算机则是为解决特定问题而专门设计制造的，用途较为单一，如导弹上专门用来计算和处理弹道数据的计算机和工业上专门用于控制生产过程的计算机等。

通用计算机按其规模与性能的不同可以划分为巨型机、大型机、中型机、小型机、微型机及单片机。“神威·太湖之光”超级计算机就属于巨型机，如图 1.1.4 所示。微型机又称为个人计算机（personal computer, PC），台式计算机、笔记本电脑、手持设备以及卡片式计算机等都属于微型机。单片机又称为微控制器（micro-controller unit, MCU），是集成在一块集成电路芯片上的、小而完善的微型计算机系统，如各种开源硬件主控板等（见图 1.1.5）。

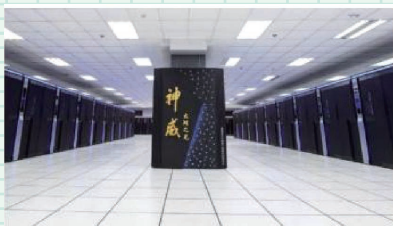


图 1.1.4 神威·太湖之光

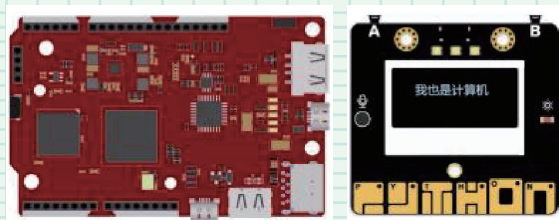


图 1.1.5 开源硬件主控板

问题讨论

观看配套资源包中的视频“不一样的计算机”，讨论下列问题。

- ◆ 与我们熟悉的计算机相比，开源硬件主控板有哪些不同？
- ◆ 开源硬件主控板可以应用于哪些领域？

二、计算工具的发展史

计算工具的演化与人类社会需求有着直接的关系。大约从“结绳记事”时起，人们就开始研究数据的记录与存储。随着人口的增多与生产规模的扩大，人们逐渐对提高计算效率产生了更为迫切的需求，各种计算工具（如算盘和计算尺，见图 1.1.6）应运而生。但这些计算工具效率较低，于是科学家们相继开发出各种能够替代人工计算的自动计算机器。



算盘



计算尺

图 1.1.6 算盘和计算尺

1. 机械式计算机

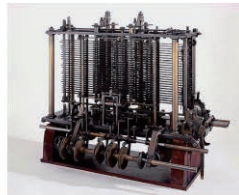
科学家们首先尝试用齿轮这种机械来实现自动计算功能。因为齿轮上的齿数可以用来表示数字，并且可通过复杂的机械转动来实现输入、计算与输出，后人便将这类机器称为机械式计算机。其典型的代表有帕斯卡加法器、莱布尼茨乘法器和巴贝奇差分机等，如图 1.1.7 所示。



帕斯卡加法器



莱布尼茨乘法器



巴贝奇差分机

图 1.1.7 机械式计算机

巴贝奇差分机是机械式计算机的重要代表，是为编制航海和天文方面的数学用表而设计。差分是一种数学运算方法，能用来解决多项式函数的问题。在巴贝奇的设计中，该机器采用十进制数字系统，通过转动手柄来驱动，手柄每转动 4 圈，差分机就会进行一次完整的加法进位运算。最初的差分机可以计算 3 个不同的 5 位数，精度达到 6 位小数。但受限于当时的机械制造水平与其他因素，差分机制造最后以失败告终。

拓展阅读

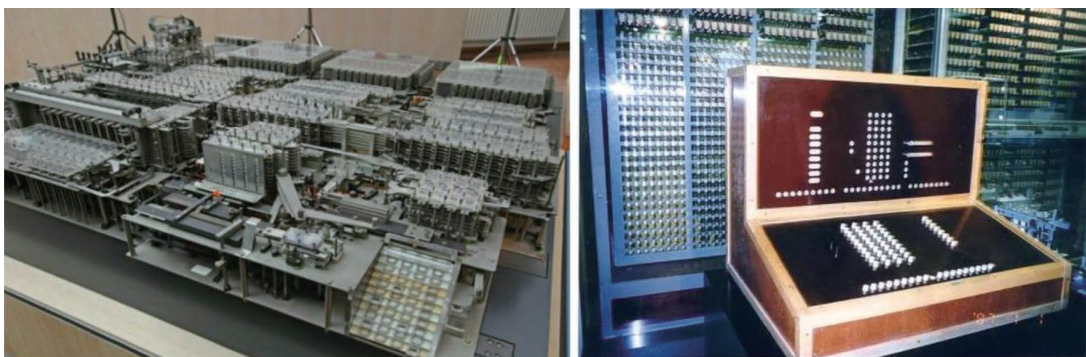
世界上第一名程序员艾达·洛芙莱斯的故事

巴贝奇在制造差分机时，认识了艾达·洛芙莱斯。艾达是英国诗人拜伦的独生女，师从计算机数学基础布尔代数的创始人摩根。她在 1842—1843 年间翻译了意大利数学家费德里科·路易吉阐述分析机的论文，译文里包含了生成伯努利数的算法。该算法被后人认为是世界上第一个正式发表的计算机程序，她也由此被认为是世界上第一名程序员。

艾达在研究差分机时，不仅提出变量与递归的思想，还就此展开丰富的想象。认为这台机器“除了数，还可能操作其他东西”，由于机器上的齿轮可以用来表示更为抽象的符号，因此在理论上也能够操纵语言、谱写音乐。“举例来说，假设在和声和作曲科学中，各种音调声音的基本关系可以用这种方式表达和改编，那么这台机器就能够谱写任意复杂度和长度、精致且科学的乐曲。”

2. 机电式计算机

19 世纪后期，电力的发明与应用将人类带入了电气时代。人们开始运用电动机、继电器来设计计算机，这类计算机称为机电式计算机。1938 年，楚泽设计出的 Z1 计算机是世界上第一台可编程的二进制计算机，他于 1941 年设计出的 Z3 计算机是世界上第一台通用程序控制的机电式计算机，如图 1.1.8 所示。同时期，著名的数学家艾伦·图灵提出了“图灵机”的概念，形成了计算机的理论雏形。



Z1 计算机

Z3 计算机

图 1.1.8 Z1 与 Z3 计算机

拓展阅读

图灵机

1936 年，英国数学家图灵从数学角度提出了一种可行的抽象机器，它就是现代通用计算机的鼻祖——图灵机。如图 1.1.9 所示，图灵机主要由处理盒、读写头与纸带组成。处理盒相当于现代电子计算机的处理器，位于处理盒下方的读写头相当于输入输出设备，而纸带则相当于内存或硬盘。按照图灵的设想，处理盒位于纸带的上方，它能够自由地移动并通过读写头读取纸带上的方格信息。在处理盒内部有已经写好的程序，它能够处理纸带上的信息并将处理的结果输出到纸带上。处理盒不断地调整自己的状态，并读入下一个方格信息，周而复始，直到程序结束。

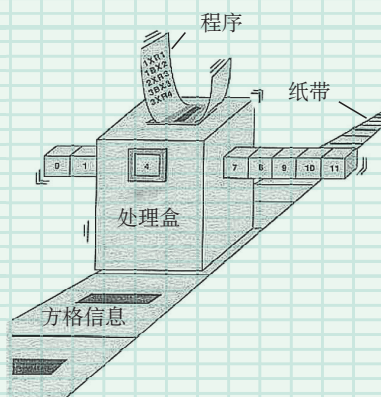


图 1.1.9 图灵机

3. 电子计算机

机械式计算机和机电式计算机都存在计算速度较慢的问题，人们开始用电子元件来研制具有更高运算速度的计算机。1946 年，美国宾夕法尼亚大学研发出了第一台采用数字技术的通用电子计算机 ENIAC（埃尼阿克），它的占地面积为 170m^2 ，质量达 30t ，能进行 5000 次/s 的加法运算，如图 1.1.10 所示。ENIAC 的诞生标志着人类进入了电子计算机时代。

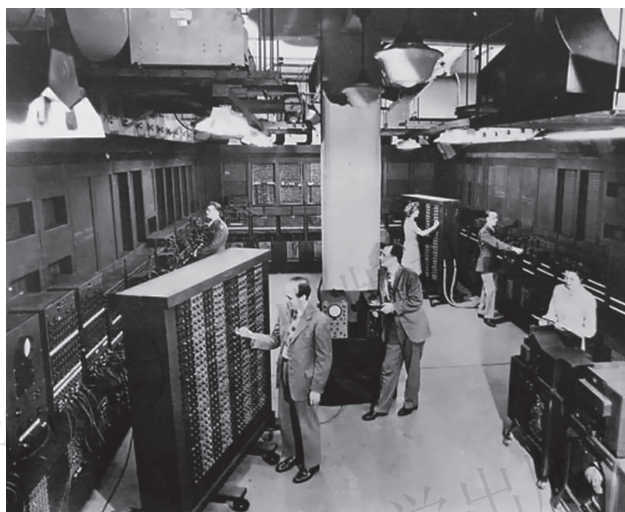


图 1.1.10 ENIAC

电子元件是电子计算机能实现高速运算的关键，因为它非常容易实现“通”与“断”两种状态，就像房间里控制电灯的开关，通电灯亮，断电灯灭。电子计算机的发展主要体现在不断地加快这些“开关”的通断速度、增加“开关”的数量及改善“开关”的材料等方面。另外，电子计算机在工作原理方面也取得了较大的突破，内部采用由 0 和 1 组成的二进制数据来实现信息的存储与运算，其中 0 代表“开关”关闭，1 代表“开关”接通。按照所使用电子器件的不同，人们将电子计算机的发展划分为四个阶段，如表 1.1.1 所示。

表 1.1.1 电子计算机的发展阶段

发展阶段	电子元件
第一代	电子管
第二代	晶体管
第三代	中小规模集成电路
第四代	大规模和超大规模集成电路



实践活动

分享计算机发展史上的经典发明

从机械式计算机到电子计算机，期间诞生了很多形式多样、性能不一的计算机产品，有的能名垂史册，有的只是昙花一现，但无论如何它们都是人类科技时代的产物，都有着非凡的现实意义。

请以小组为单位，选择一款你最感兴趣的计算机进行分享。

实践内容主要包括：

- (1) 确定分享的主题。
- (2) 收集、整理相关资料。
- (3) 制作 PPT 并进行分享。

拓展阅读

未来计算机

以集成电路为基础的传统计算机芯片，会因为线路密度的增加而达到物理极限。为此人们开始研发各种新型芯片计算机，其中生物计算机、光子计算机与量子计算机被认为是最具前景的未来计算机。

生物计算机是用生物芯片取代传统计算机芯片而制成的计算机。如图 1.1.11 所示，它是利用生物蛋白质的开关特性，用蛋白质分子作为元件制成生物芯片，具有体积小、功效高并拥有自我修复能力的特点。

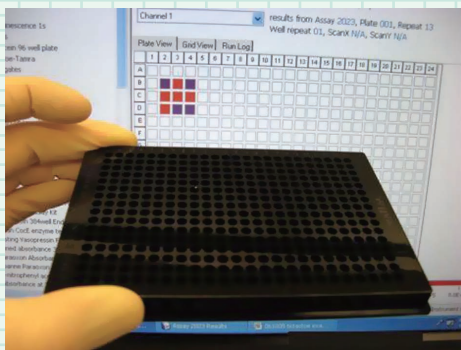


图 1.1.11 生物计算机

光子计算机是一种通过光信号来进行信息存储、处理、运算与操作的新型计算机。它由激光器、光学反射镜、透镜、滤波器等光学元件和设备构成，靠激光束进入反射镜和透镜组成的阵列进行信息处理。光子计算机的并行处理能力很强，运算速度很快，将在图像处理、目标识别及人工智能等方面发挥更大的作用。我国科技公司推出了第二代光子计算处理器——PACE，如图 1.1.12 所示。它在单个光子芯片中集成超过 10000 个光子器件，运行 1GHz 系统时钟，算力是上一代处理器的 100 万倍以上，在运行特定循环神经网络时可比当时的高端 GPU 快 100 倍。

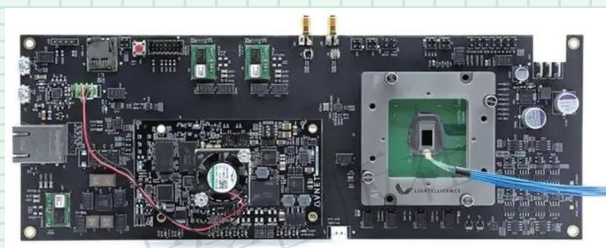


图 1.1.12 PACE 光子计算处理器

量子计算机是根据量子力学中微观物质所特有的量子相干叠加特性，采用新的信息处理单位——量子比特，并结合特定的量子算法进行运算。它在信息存储和处理能力方面远高于传统的计算机。中国科学技术大学团队于 2023 年 10 月 11 日宣布，成功构建了如图 1.1.13 所示的量子计算原型机“九章三号”，并再度刷新光量子信息技术的世界纪录。“九章三号”在求解高斯玻色取样数学问题方面的速度，比目前全球最快的超级计算机快一亿亿倍。

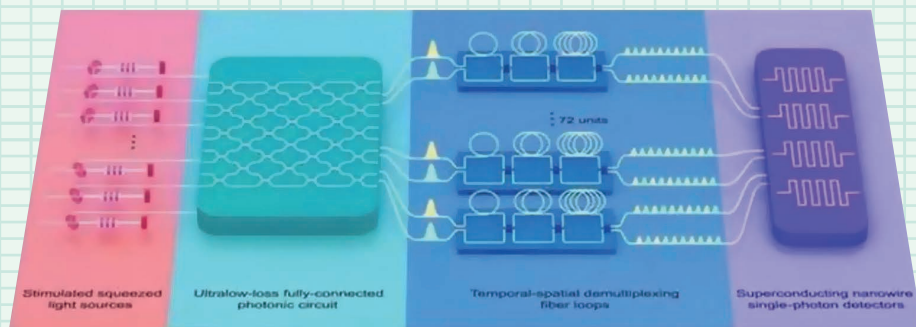


图 1.1.13 “九章三号”实验装置示意图

项目实施

本节课的知识可以帮助你解决项目中的哪些问题？小青团队经过辨识，共同确认小青的计算机是一台家用型的笔记本电脑。

你要升级的计算机属于哪种类型？请填写在下面的横线上。

清华大学出版社
清华大学出版社

第 2 节 计算机硬件

本节知识

- ◆ 硬件系统及原理
- ◆ 输入输出接口与设备
- ◆ 内存储器和外存储器
- ◆ 中央处理器

本节活动

- ◆ 硬件系统检测实验
- ◆ 图拉丁硬件检测软件的探讨
- ◆ 扩展坞开发原因的分析
- ◆ 芯片实现“自主可控”的讨论
- ◆ 平板电脑选购实践

人们在选购计算机时，总会优先比较它们的各项硬件性能指标。硬件（hardware）是计算机科学的专用名词，是计算机在现实世界中可以被触摸和感知的部分，是计算机系统所有实体设备的总称，也是计算机软件运行的物质基础。

一、硬件系统及原理

从体积异常庞大的 ENIAC 到可以握在手里的智能手机，电子计算机经历了七十多年的发展，虽然外形发生了很大的变化，但是硬件系统的基本规则并没有改变，尤其是硬件系统的基本组成与工作原理等方面。

1. 计算机硬件系统的组成

如图 1.2.1 所示，电子计算机的硬件系统由五大基本部件组成，分别是运

算器、控制器、存储器、输入设备和输出设备。其中运算器、控制器通常会被集成到一起，合称为中央处理器（central processing unit, CPU）。而存储器则有内存储器与外存储器之分。

硬件系统的五大部件彼此分工明确又配合默契。输入设备负责将外界信息转换为机器能够识别的二进制数据；输出设备负责将机器运算结果转换为人们能够感知的图像、声音等信息形式；内存储器主要负责数据交换或加载程序与数据，外存储器则用来存放计算机程序与数据，并通过内存储器与中央处理器进行间接的数据交换；运算器是数据的加工厂，负责数据的各种运算；控制器是总指挥，它根据程序中的指令协调各个部件的工作。

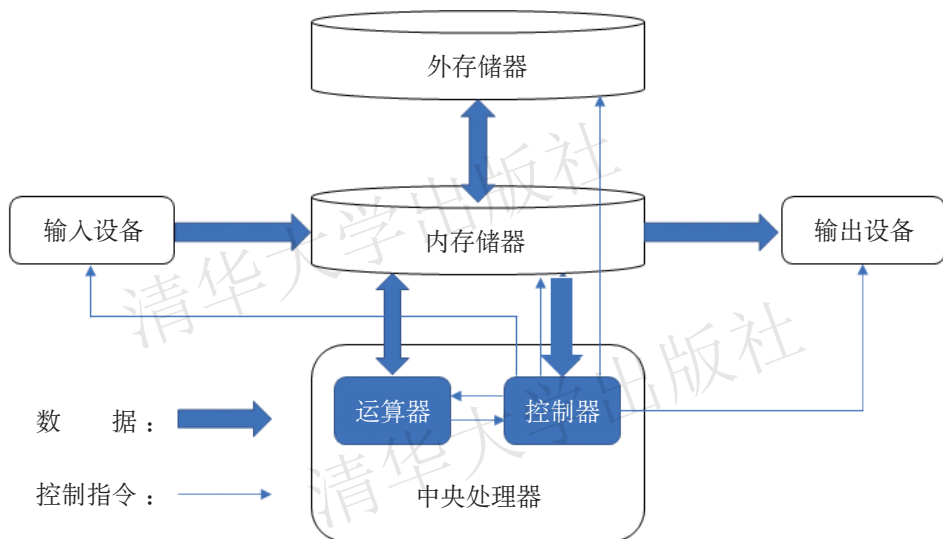


图 1.2.1 计算机硬件系统

2. 计算机工作原理

能够自主地开展工作是电子计算机的主要特征，其背后是存储程序与程序控制这两大工作原理。存储程序是指计算机将解决问题的程序加载到内存储器中，以方便中央处理器直接访问。程序控制是指控制器按照先后顺序自动读取程序中的指令来执行任务，直到程序结束。这好比是想让他人为你做事，要先将“做什么事、怎么做事”的纸条递给他，他人再根据纸条上的具体指示进行做事。

概念解释

指令与程序

指令是程序员指示计算机 CPU 开展工作的最小操作命令，通常由操作码和地址码组成。操作码用于指示 CPU 将要开展什么工作，而地址码则是指向内存存储器的某个空间位置，这里存放着被操作的程序或数据。程序是由程序员编写的，是为了实现某个目标而设计的一组有顺序的指令集。加载在内存存储器中的程序与数据，都是通过地址码而被 CPU 访问到，如图 1.2.2 所示。

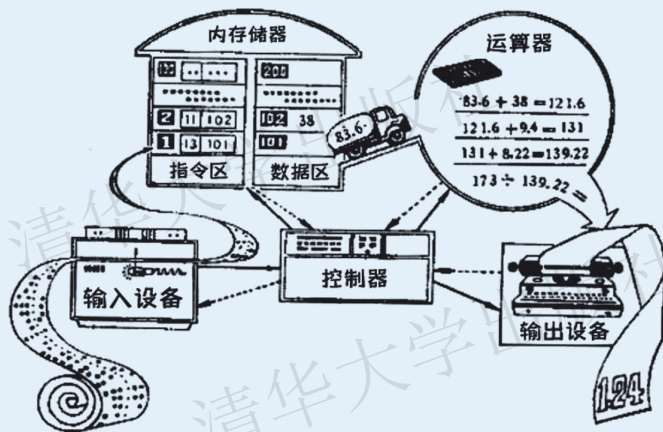


图 1.2.2 内存存储器中的指令与数据区



实验活动

硬件系统检测实验

封装在机箱里的核心硬件通常是衡量计算机性能的重要指标，但一般用户不易了解其确切信息。利用某些专用的工具软件，可以在不开箱的情况下检测它们的配置信息，从而为用户提供帮助。请完成硬件系统检测实验活动，并填写表 1.2.1 及结论。

实验内容：利用专用工具软件检测教师机与学生机的硬件配置差异。

实验准备：学生机、教师机及“图拉丁硬件检测”工具（一款免费的国产软件）。

表 1.2.1 硬件检测记录表

硬件名称	教师机检测信息	学生机检测信息
中央处理器		
内存		
硬盘		

结论： _____

问题讨论

如图 1.2.3 所示，在硬件系统检测实验中用到的图拉丁硬件检测软件有哪些特点，这对你今后选择工具软件有什么启发？



图 1.2.3 图拉丁硬件检测软件

二、输入输出设备

输入输出设备简称为 I/O 设备，是计算机硬件系统中输入设备与输出设备的合称，也是计算机实现与用户进行人机交互的必要装备。为了满足不同工作环境下的交互需求，人们开发出了种类繁多的输入输出设备。

1. 常见的输入设备

如图 1.2.4 所示，键盘、鼠标、触摸屏、摄像头、麦克风、扫描仪、手写板及传感器等，都是常见的电子计算机输入设备。



图 1.2.4 常见的输入设备

其中最常见的输入设备是键盘与鼠标，键盘主要用于输入字符与控制命令，鼠标则用于屏幕坐标的定位。触摸屏是一款可以接收触头输入信号的装置，广泛应用于手持设备中。触摸屏是盖在显示屏上的一层薄透明层，内有细小、透明的金属丝网格，当手指触摸网格时，就会影响流过该网格的电流，从而产生输入信号。

2. 常见的输出设备

常见的输出设备有显示器、打印机、3D 打印机及音箱等，如图 1.2.5 所示。



图 1.2.5 常见的输出设备

其中显示器是重要的输出设备，当前主流显示器可以分为 LCD 液晶显示器与 OLED 屏两种。LCD 液晶显示器适用于台式计算机，OLED 屏适用于智能手机与平板电脑。打印机可将计算机处理的结果打印到纸张及其他相关介质上，较常见的有喷墨打印机与激光式打印机。

3. 输入输出设备的物理接口

无论是何种输入输出设备，都要通过特定的物理接口与主机进行连接。大部分的物理接口，都会采用统一的标准，以便不同厂家生产的设备都能接入。以台式计算机为例，常见的输入输出设备的物理接口及连接线如图 1.2.6 所示。



图 1.2.6 输入输出设备的常用接口及连接线

蓝牙 (bluetooth) 是一种支持不同设备间进行短距离通信的无线技术标准，使用蓝牙设备必须先进行连接配对。大部分的无线键盘、无线鼠标及蓝牙音箱等输入输出设备，都通过蓝牙方式进行连接。

PS/2 接口是旧款台式计算机的专用输入装置接口，不支持带电插拔，分为

USB 的全称是通用串行总线，是一种用途广泛，支持带电插拔并且种类繁多的接口。当前常用的有 USB2.0（接口为黑色）、USB3.0（接口为蓝色）及 Type-C。USB2.0 与 USB3.0 均支持对应接口的有线键盘、有线鼠标及打印机等设备，而 Type-C 则是一款新兴的接口，目前广泛应用于笔记本电脑与手机领域。

VGA 是老款台式计算机显卡上专门用于连接显示器的接口。它采用 D 形接口, 共有 15 针, 分成 3 排, 每排 5 个孔。

音频接口位于声卡上，主要用于连接麦克风、音箱等设备。绿色孔连接音频输出，粉色孔连接麦克风。

扩展坞

扩展坞又称端口复制器，是专为笔记本电脑与手机等种类计算机而设计的一种连接设备。如图 1.2.7 所示扩展坞，可将计算机的一个 Type-C 端口扩展成为支持 HDMI、Type-C、USB3.0、USB2.0 及 SD 卡与 TF 卡的插槽。



图 1.2.7 扩展坞

问题讨论

为什么要为笔记本电脑与手机等可移动的计算机开发扩展坞，请分析原因。

三、存储器

存储器是直接影响计算机整体性能的重要组件，而存储容量与存储速度则是评价存储器性能两大重要指标。一般来说存储速度越快、存储容量越大，其价格就越贵，无论是外存储器还是内存储器都是如此。

1. 外存储器

外存储器简称外存，又称为辅助存储器，通常用来存放需要永久保存或暂时不需要的程序或数据。它具有不依赖电源、读写速度慢、容量大且价格低等特性。

计算机常用的外存储器有硬盘、U 盘等。硬盘又分为机械硬盘（HDD）、固态硬盘（SSD）和混合硬盘（HHD）等。如图 1.2.8 所示，机械硬盘依赖磁性介质存储数据，通过磁头读取信息，因此抗震防摔性较差；固态硬盘具有读写速度较快、体积小、无噪声、低功耗等特点，但价格相对较高；混合硬盘是将机械硬盘与固态硬盘制作在一起的硬盘。

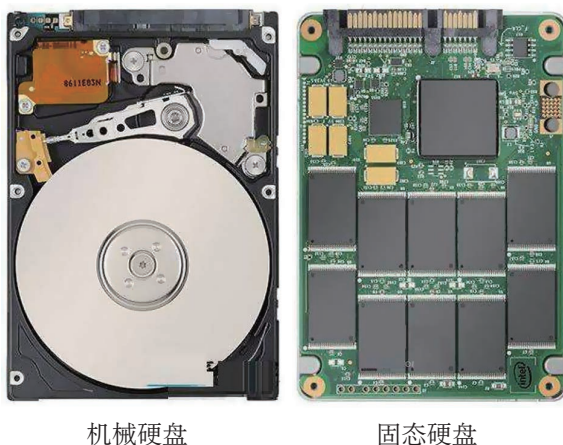


图 1.2.8 计算机硬盘

常见的智能手机的外存储器以 TF 卡为主。TF 卡即 T-Flash 卡，又叫作微型 SD 卡（microSD），是一种外形小巧的存储器卡。数码相机、多媒体播放器等便携式装置广泛使用 SD 卡，它是一种高速存储设备。TF 卡与 SD 卡如图 1.2.9 所示。

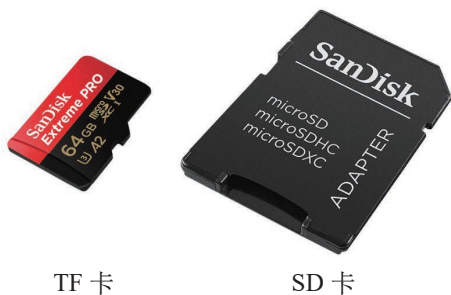


图 1.2.9 TF 卡与 SD 卡

拓展阅读

存储容量计算标准的争议

存储容量指存储器所能存储二进制信息的总量，最基本的存储容量单位是字节（Byte），常见的存储容量单位有 KB、MB、GB 与 TB。在计算机科学里，标准的换算关系如下：

$$1\text{KB}=1024\text{B} \quad 1\text{MB}=1024\text{KB} \quad 1\text{GB}=1024\text{MB} \quad 1\text{TB}=1024\text{GB}$$

生产存储器的厂家为了计算方便，普遍按照 1:1000 的关系进行换算和标注。当买到标注为 500GB 的硬盘时，实测可能只有 465GB 左右，故容易引起买卖双方的争议。

2. 内存存储器

内存存储器简称内存，又称为主存储器，是 CPU 可以随机访问的存储器，具有断电后信息立即丢失的特点，一般用于暂时存放信息。

人们平时所说的内存条，是指插在计算机主板内存插槽中的部件，它是内存存储器的一种。当前主流的内存条采用的是 DDR（double data rate，双倍速率）体系，其工作频率从 2133 ~ 4200MHz 不等，单条内存条的存储容量有 4GB、8GB、16GB 几种。计算机主板内存插槽与内存条如图 1.2.10 所示。

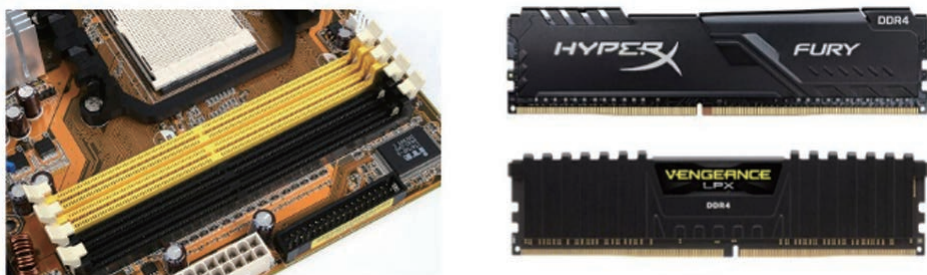


图 1.2.10 内存插槽与内存条

四、中央处理器

中央处理器（CPU）是计算机所有硬件中最核心的设备，也是评价计算机性能的重要指标之一。CPU 是一块超大规模的集成电路，包含运算器、控制器、缓存及其他外围设备。CPU 的内核数量、主频、制作工艺及缓存等都是评价其性能的重要指标。在同等条件下，内核数量越多，主频越高，CPU 的性能越好。

Intel、AMD 是目前国际市场上主要的台式计算机 CPU 品牌。飞腾、鲲鹏、海光、龙芯、兆芯与申威等是国产的台式计算机 CPU 厂商或品牌；麒麟是国产的智能手机 CPU 品牌，它们如图 1.2.11 所示。



图 1.2.11 国产 CPU 的厂商或品牌

问题讨论

CPU 只是众多芯片中的一个类别。芯片又称为集成电路，是一种微型电子器件或部件，它内含集成电路，通常制造在半导体晶圆表面上。国产芯片当前正经历从“被卡脖子”到逐步实现“自主可控”的发展阶段，你认为实现芯片技术自主可控，对提升国家信息安全有什么意义？

拓展阅读

CPU 指令集及其架构

CPU 指令集是 CPU 用来控制与计算的指令集合。主流的 CPU 指令集可分为复杂指令集（CISC）与精简指令集（RISC）两大类。采用 CISC 的称为复杂指令集架构，典型如 x86 架构；采用 RISC 的称为精简指令集架构，典型如 ARM 架构。台式计算机与笔记本的 CPU 一般采用 x86 架构，而手机等手持设备的 CPU 大多采用 ARM 架构。

RISC-V 是近年来异军突起的一个基于 RISC 原则的开源指令集架构。它与 x86、ARM 等传统架构相比，具有技术后发与不受版权限制等方面优势，是部分国产 CPU 采用的指令集架构。

LoongArch™是龙芯中科推出的具有中国自主知识产权的 CPU 指令集。使用该指令集的架构被称为龙架构，它具有如图 1.2.12 所示的主要功能。

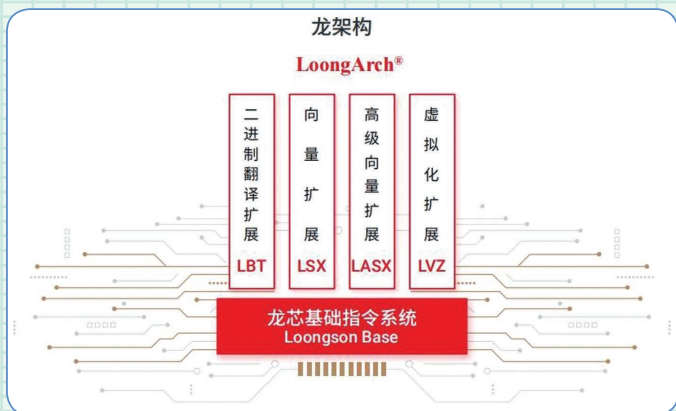


图 1.2.12 龙架构及其主要功能



实践活动

平板电脑推荐实践

平板电脑的屏幕通常要比手机大，而重量又比手提电脑要轻，因此成为大家用来开展在线学习与娱乐活动的首选。平板电脑的品牌、性能与价格，通常是影响人们选购的重要因素。请模拟为他人推荐几款价位区间不同的平板电脑。

实践内容主要包括：

- (1) 自选网购平台，以检索相关信息。
- (2) 对收集到的相关信息进行分析与评价。
- (3) 撰写不同价位区间的推荐购买建议。
- (4) 与组内同学进行交流分享。

项目实施

本节课的知识可以帮助你解决项目中的哪些问题？小组团队检测并记录了可能会影响计算机软件升级的部分硬件性能，如硬盘、内存与 CPU 等。

你的计算机硬件性能如何？请填写在下面的横线上。

第 3 节 计算机软件

本节知识

- ◆ 软件
- ◆ 操作系统
- ◆ 应用软件

本节活动

- ◆ 安装并设置 Linux 系统
- ◆ 操作系统类别划分的讨论
- ◆ 学生机应用软件类别的分析
- ◆ 安装 Linux 应用软件

计算机的功能不只受限于硬件，还取决于它的软件。因为硬件好比是人的身躯，而软件就是其思想与知识。性能强大的计算机不能只有“发达的四肢”，更要有“聪明的头脑”。因此人们为计算机安装软件的目的，就是不断地丰富它的思想与知识。

一、软件与通用计算机

仅有硬件的计算机是一台不能工作的“裸机”，只有配上软件后才能称为完整的计算机系统。软件（software）是指在计算机上运行的程序、数据和文档的总和。软件的运行必须以硬件为基础，硬件的性能发挥则须依赖软件设计，因此它们是相辅相成的发展关系。当硬件进步时，就会对软件提出迭代更新的要求；而新软件的普及，往往又会推动硬件的进一步发展。

早在 1888 年，霍列瑞斯的穿孔制表机上就出现了“软件”的雏形。自从有了“软件”这个概念之后，计算机就开始逐步发展成为一个“通用”的机器，因为只要为其安装相应的不同软件就可以为各行各业服务了。但早期计算机大多只能应用于某些特定工作领域，因为那时软件都是使用二进制数字来编写，

并且是用如图 1.3.1 所示的打孔纸带来存储，其程序代码的可读性、通用性与可移植性都很差，导致软件品种稀少而功能不像今天这般强大。

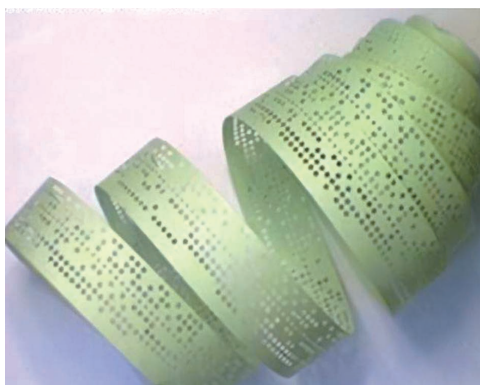


图 1.3.1 早期存储程序代码的打孔纸带

如今计算机能广泛地应用于各行各业，主要得益于软件业的发展与进步，他们不断地开发出种类繁多的软件，并能满足不同用户各种需求。软件可以分为多种类别，其中系统软件与应用软件最为常见。大家所熟知的操作系统，就是最主要的系统软件，因为它为计算机系统的运行和维护提供了基础。

二、操作系统及其分类

操作系统是负责管理操作计算机各种资源的软件，可对计算机系统中的所有软硬件资源进行统一控制、调度与管理。无论是安装或卸载软件、存储或删除文件，还是插拔 U 盘与耳麦等，都必须通过它来实现，因此是用户和计算机之间的桥梁。

操作系统发展与计算机硬件进步密不可分。当计算机发展出服务器、个人计算机、智能手机与其他不同类别的计算机之后，就需要研发与之匹配的操作系统，于是出现如图 1.3.2 所示的各种操作系统。但这种分类不是绝对的，因为今后还有可能发展出新类别的计算机。

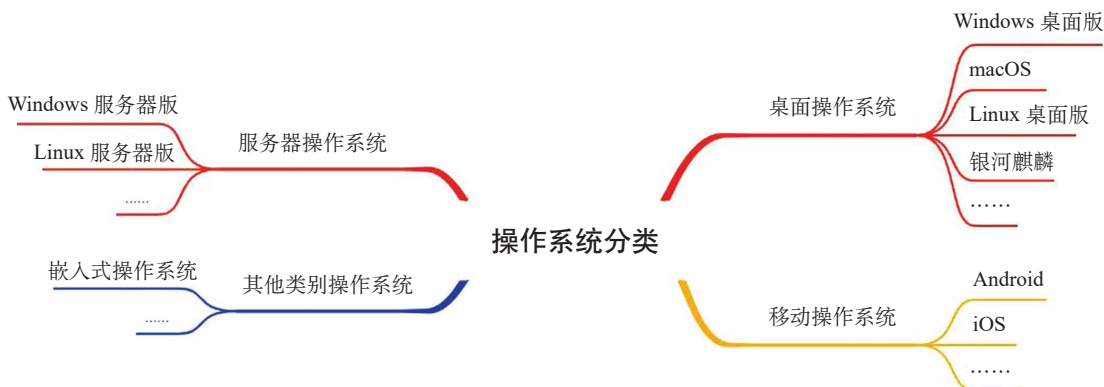


图 1.3.2 计算机操作系统的分类

1. 桌面操作系统

桌面操作系统是面向普通用户，适用于个人计算机与笔记本电脑的操作系统，常见的操作系统有 Windows、macOS 与 Linux 三大系列。其中 Windows 操作系统市场占有率较大，具有硬件支持良好、应用程序多等方面的优势。而 macOS 操作系统仅限于 Macintosh（简称 Mac）系列计算机，是一款相对封闭的操作系统，它对应用程序的安装有较为严格的限定，但运行流畅感反映普遍较好。Linux 则是一款免费、开源的操作系统，并且有非常多的衍生版本，它除了桌面操作系统之外，还有服务器及其他类型的发行版本。

拓展阅读

开源软件与共享软件

开源软件 (open-source) 是指开放源代码并且使用、修改与分发不受许可证限制的软件。由于它降低了软件开发成本，提升了使用灵活性与安全性，带来了技术创新与发展等，因此对促进社会向上发展具有非常重要的意义。

共享软件按照用途可以分为商用软件与非商用软件两大类：商用软件采用先试用后购买的模式，即软件开发商为用户提供有限期的或者部分功能受限的免费试用版本，以帮助用户测评软件是否符合自己的需求，进而决定是否购买；非商用软件大多是软件设计爱好者为展示才华而向大众提供的免费软件。



体验活动

安装并设置 Linux 操作系统

Linux 是一款可供任何个人和机构自由使用、修改和发布的多版本开源操作系统。Ubuntu 是众多 Linux 发行版的后起之秀，它出现之后就打破

了 Linux 被认为是难以安装与使用的普遍认知。Ubuntu 拥有庞大的社区力量，用户可以方便地从社区获得帮助。

请在虚拟机中加载 Ubuntu (21.04) ISO 镜像文件（见图 1.3.3），然后进行安装并运行。通过桌面空白处的右键快捷菜单中的“更改背景”命令，来为桌面更换背景图片。通过系统设置里的“日期和时间”选项，来更改系统的显示时间格式。

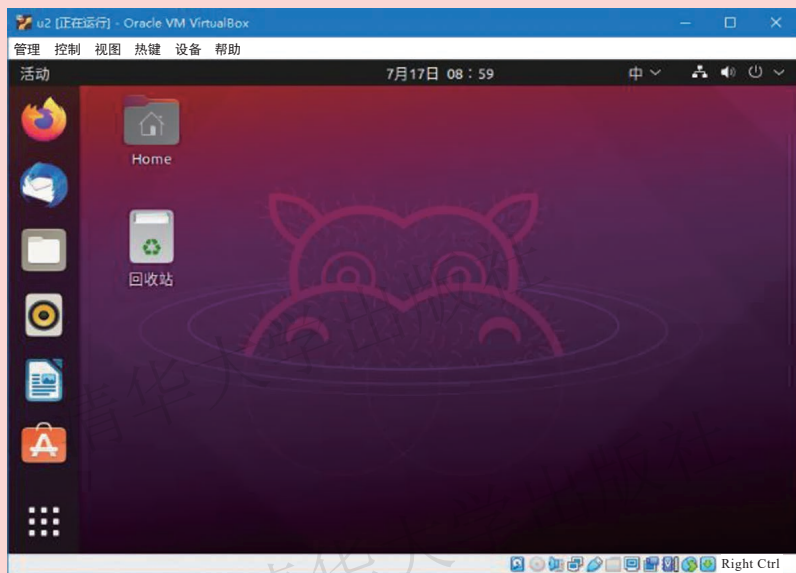


图 1.3.3 Ubuntu 操作系统 (21.04)

概念解释

虚拟机

虚拟机即计算机系统的仿真器。它是在实体计算机上通过软件模拟出的一台或者多台虚拟计算机。虚拟机使用宿主机的硬件资源，拥有真实计算机的大多数功能。较常见的虚拟机软件有 Oracle VM VirtualBox、VMware Workstation、Parallels Desktop、VirtualBox 与 Microsoft Hyper-V 等。



2. 移动操作系统

移动操作系统适用于手机、平板电脑等设备，常见的有 Android 与 iOS 两大类。Android 又称为安卓，是基于 Linux 开发的开源操作系统，它支持多数厂家生产的手机、平板电脑及其外接设备，并且广泛应用于智能电视、数码相机等领域。iOS 仅支持某些特定类别的手机、平板电脑以及经过相关认证的外接设备。

3. 服务器操作系统

服务器往往长期工作在网络环境中，在稳定性、安全性方面的需求比个人计算机高，因此需要安装专用的操作系统。常见的服务器操作系统包括 Windows Server、Netware、UNIX 和 Linux 四类。

服务器操作系统也可以安装在个人计算机上。虽然它在文件管理与网络应用方面功能较好。但在配置方面相对更加复杂。

4. 其他类别操作系统

操作系统种类繁多，除了上述三类基本操作系统外，还有一些其他特殊应用的操作系统。如在一些专用设备中（航空航天、家用电器等），把计算机嵌入其中进行实时监控和管理。运行在嵌入式系统中的操作系统，称为“嵌入式操作系统”。一般来说，这类系统中所有计算机软件都事先保存在 ROM 里，不允许用户在应用时安装软件。

拓展阅读

国产操作系统

近年来国产操作系统呈现较好的发展势头，诞生了银河麒麟 KylinOS、统信 UOS 与鸿蒙 HarmonyOS 等一批知名操作系统，进而为实现国家安全提供保障。

银河麒麟可提供服务器操作系统、桌面操作系统与嵌入式操作系统等产品，并支持 x86、ARM 与龙架构等 CPU，如图 1.3.5 所示。



图 1.3.5 银河麒麟操作系统

统信提供了家庭版、专业版与服务器版三款操作系统产品，并支持 ARM 与龙架构等 CPU，如图 1.3.6 所示。



图 1.3.6 统信操作系统

鸿蒙操作系统是适用范围广泛的新一代智能终端操作系统，它宣称只要内存容量大于 128KB 的任何智能终端设备都能使用，如图 1.3.7 所示。



图 1.3.7 鸿蒙操作系统

问题讨论

鸿蒙是一款类型独特的操作系统，它既支持台式计算机、平板与手机等常见的计算机，也支持车载电脑、智能电视、智能冰箱与其它各种智能终端。你能否将它进行正确分类？如果不能，又说明了什么？

三、应用软件及其划分

应用软件是指为解决计算机用户的特定问题而开发的软件。它通过使用操作系统提供的接口，完成人们实际的工作，如编辑图片、播放音乐、文字编辑及数据处理等。

1. 应用软件的分类

应用软件的分类非常复杂。比如可以根据是否收费，将其分为商业软件和免费软件；也可以根据功能的不同，将其细分为计算机辅助设计、文字处理、网页浏览、媒体播放等软件；还可以根据运行操作系统的不同，将其分为 Windows 应用（又可分为 32 位应用与 64 位应用）、macOS 应用与 Linux 应用等；甚至可以根据硬件平台的不同，将其分为服务器应用、计算机应用和移动应用（Application 或 App）等。

问题讨论

学生机上通常会安装好教学所需的各种应用软件。请以你的计算机为例，查看并分析它们都属于哪类的应用软件。

2. 常见的文件类型

软件以及软件运行中产生的数据，在计算机中以文件的形式存储。按照计算机文件能否在操作系统中直接运行，可以将其简单地分为可执行文件和不可执行文件两大类。一般来说，由应用软件生成或编辑的文件都属于不可执行文

件，也称为数据文件，如 Word 软件编辑产生的“.docx”文件，画图软件生成的“.bmp”文件等。

为了方便人们快速地识别不同类型的文件，操作系统通常会通过文件名及图标进行区分。文件命名通常采用“主文件名.扩展名”的格式，主文件名一般用于描述文件的性质与作用，扩展名则用来表示不同格式类型的文件。比如 Windows 系统下可执行文件的扩展名通常为“.exe”。

3. 应用软件的获取

应用软件的获取主要有两种途径：一是从实体商店直接购买；二是通过网络获取。按照渠道的不同，网络获取又可以分为三种方式：一是到软件开发商的官方网站下载；二是通过第三方软件下载，如“360 软件宝库”“腾讯软件中心”等审核相对严格的网站；三是通过手机或部分计算机自带的“应用商店”获取。

大部分的商业软件需要购买其授权许可序列号才能正常使用，但也有很多免费的软件，比如测试版商业软件、开源软件、共享软件等。一般来说，开源软件是一种相对较好的选择，因为测试版商业软件可能不够稳定，而共享软件则有植入广告、恶意代码及绑定安装等陷阱。

手机、平板电脑等手持设备上的应用软件又称为移动应用（App），它们的主要获取渠道是应用商店。应用商店是专为 App 开发商与用户提供交易的平台，App 开发商可以在这里展示并推销自己的 App，用户则通过应用商店搜索、下载与评价 App。应用商店的运营者虽然也会对里面的 App 进行资格、安全及质量等方面的审查，但仍然难以解决安全漏洞、要求权限过多、远程载入插件与广告等一系列问题。因此用户在选择 App 时，应该注意查看应用商店里的评价信息，并在安装过程中注意对应用权限进行限制。

4. 应用软件的安装

应用软件的安装与操作系统有关。

在 Windows 操作系统下的安装，一般只需找到并运行“setup.exe”或“install.exe”之类的文件，然后按照提示一步一步操作即可。

macOS 操作系统的应用软件，大部分是通过运行“.dmg”文件进行安装。若是不经 AppStore（Mac 计算机自带的“应用商店”）下载的安装文件，会

收到警告信息。安装完成以后需要把快捷方式拖到应用程序文件夹中。

在 Linux 下安装应用软件通常有两种方式：以 Ubuntu 为例，一是使用 Ubuntu Software（即 Ubuntu 系统自带的“应用商店”）安装，如图 1.3.8 所示；二是直接在终端通过 apt 命令完成，如图 1.3.9 所示。“sudo apt install pinta”表示安装图片编辑软件 pinta，而使用 apt 命令安装的前提是需要知道软件的名称。

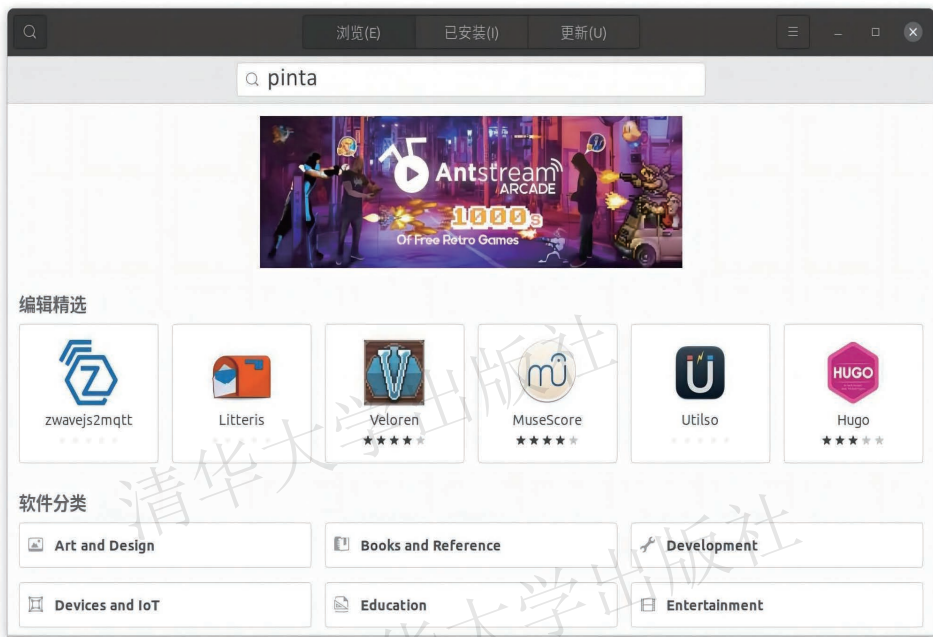


图 1.3.8 通过 Ubuntu Software 安装应用软件

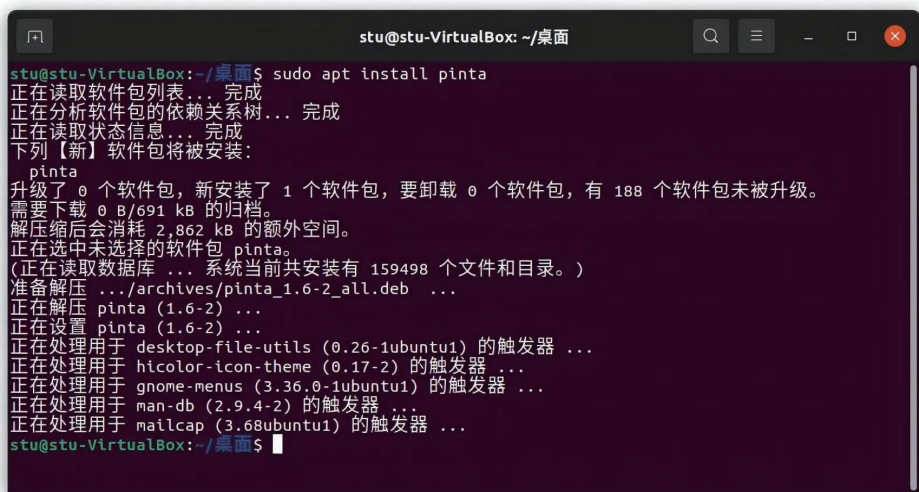


图 1.3.9 通过终端 apt 命令安装应用软件

体验活动

安装 Linux 应用软件

近年来，随着 Linux 系统的市场占有率逐步扩大，支持它的应用软件品种也日渐丰富。以 PDF 阅读软件为例，就有很多支持 Linux 系统的产品，Okular 就是其中一款，它功能强大、开源并且可以跨平台应用，如图 1.3.10 所示。

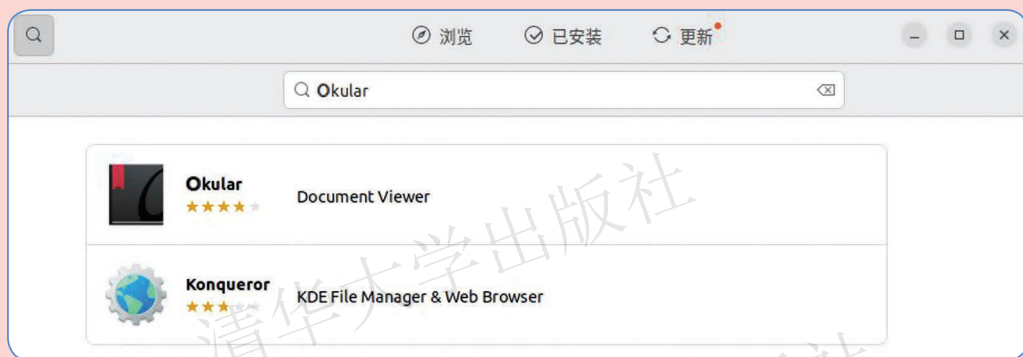


图 1.3.10 安装并体验 Okular

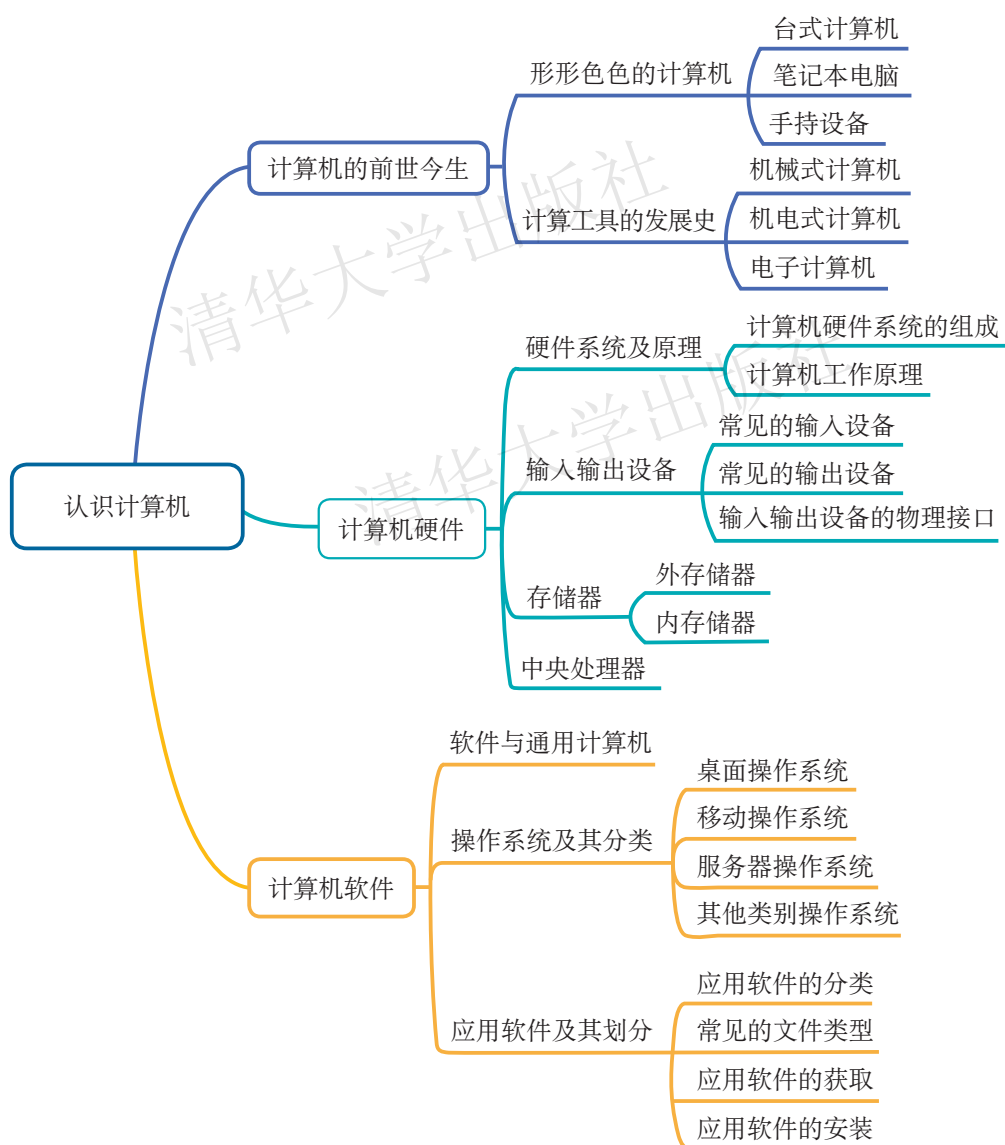
请打开 Ubuntu 自带的应用商店“Ubuntu Software”，在搜索框搜索并安装 Okular，然后体验使用 Okular 浏览 PDF 文件。

项目实施

本节课的知识可以帮助你解决项目中的哪些问题？小清团队决定将小清计算机的操作系统更换为开源的 Ubuntu，并将原办公应用软件改为 Libre Office。你的计算机软件将会做怎样的升级？请填写在下面的横线上。

单元小结

一、知识回顾

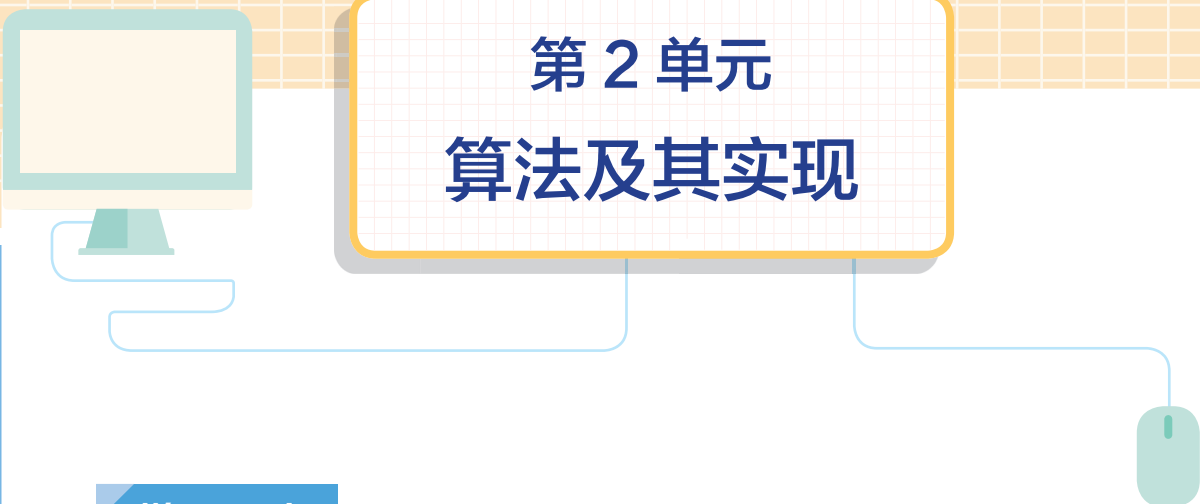


二、成果交流与评价

1. 参考本书附录“项目报告模板”撰写项目报告，并制作演示文稿。
2. 在课堂内展示自己的学习成果并分享经验，在下表中进行自评和他评。
3. 保存相关数据和文档，并上传至校园网或者其他学习空间，与他人分享学习成果。

项目成果评价表

评价维度	自评	他评
(1) 完整性 项目材料齐全，有需求分析、方案设计、分工协作、项目实施记录表及项目成果等	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般
(2) 实用性 成果内容具体，有真实的项目问题描述、有效的解决方案及相关工具等	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般
(3) 规范性 项目报告规范，符合项目报告的一般格式要求，文字表述准确	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般



第2单元

算法及其实现

学习导引

计算机是信息时代最为重要的工具，人们的工作、学习、娱乐及生活都离不开它。了解信息需要互联网和搜索引擎，打字绘图需要文字处理与绘图软件，出门旅行需要专业的网络购票、打车、导航及住宿平台，即便是在家看电视或听音乐，也需要视频播放软件的支持。软件也称为计算机程序（computer program），其本质就是一组指挥计算机进行工作的指令。使用软件解决日常生活中的问题，就是一个分析问题、设计算法并编写程序代码的过程。

本单元将带领大家初步学习计算机算法，了解算法设计及其实现的一般过程，掌握运用计算机语言编写程序的基本方法与注意事项，为进一步使用计算机程序解决实际问题打好基础。

项目情景

小清发现网络中的计算机应用软件虽然品种丰富，但却无法完美地解决类似于“给现场比赛自动计分”“给汉字自动注音”等充满个性需求的问题。于是就和几位同学一起成立了“信息科技社团”，准备自行开发这些程序。但在实施过程中，遇到了如下一些问题。

- (1) 不知道选用哪种计算机程序语言来开发？
- (2) 也不知道要用哪些算法才能实现？
- (3) 更不清楚如何快速地实现开发？

.....

假如你也是该社团的一员，请和小清一起，共同完成其中一项开发任务吧。

项目方案



经过咨询与了解，小清设计了如下方案。

知 识 学 习	实 施 步 骤	预 期 成 果
(1) 学习教材中的相关知识 (2) 自学教材之外更多关于计算机语言的知识与程序开发要求 (3) 通过网络了解与项目主题相关的专业知识	(1) 分析并确定程序的具体功能 (2) 设计实现程序功能的方法与步骤 (3) 运用计算机语言编写代码，测试并优化程序 (4) 撰写程序使用文档 (5) 撰写项目报告	(1) 程序设计方案 (2) 程序使用文档 (3) 程序代码 (4) 项目报告

你对小清的项目方案有什么不同的看法或建议？你准备如何设计项目方案？请填写在下表中。

知 识 学 习	实 施 步 骤	预 期 成 果

项目分工

方案设计完成后，小青发现仅凭一己之力很难完成这个项目，于是决定邀请有共同想法的同学一起参与，并在项目方案中增加了如下表格。



姓名	角色	分 工	任 务
小青	组长	负责提出项目需求，并主持整个项目实施活动，撰写项目报告	项目整体规划设计，项目实施统筹、协调、监督、总结，撰写项目报告文档，制作项目汇报 PPT，对外展示交流等
同学甲	成员	撰写程序设计方案	明确程序的具体功能、设计程序的算法及结构等
同学乙	成员	编写程序代码	研究并确定程序的开发环境，编写并测试程序代码
同学丙	成员	撰写程序使用文档	确定程序的使用环境、适用对象及操作流程等

你认为小青项目的成员构成、分工和任务分配是否合理？请在下表中填写你的项目分工情况。

姓名	角色	分 工	任 务

第1节 问题与算法

本节知识

- ◆ 生活中的问题与算法
- ◆ 算法的特征及设计描述

本节活动

- ◆ “指挥机器”与“引领他人”的相似性讨论
- ◆ 算法五大特征的必要性讨论
- ◆ 绘制求解绝对值的算法流程图

随着计算机技术的发展，人们解决问题的方式也在不断地发生变化。人们越来越多地利用计算机领域的概念、思想和方法解决现实生活中的问题，并逐渐形成问题分解、抽象建模、设计算法、编写程序的解决模式。

一、生活中的问题与算法

人类社会之所以进步与发展，是因为人们在日常生活中一直不断地发现、提出各种问题，并积极寻求解决问题所需的各种策略、方法与途径。在运用计算机解决问题的实践过程中，人们逐步形成了算法的概念。比如寝室墙上贴有提示学生良好生活作息的时间表，茶叶包装袋上印有提示消费者科学泡茶步骤，网购平台提供指导用户正确操作的方法等。

1. 引领问题解决的指令表达

人们在生活中为了引领他人精准地完成特定任务或解决指定问题，通常会采用各种恰当的方式来表达自己的指令，如图 2.1.1 所示。

春夏作息时间表

2024 年 3 月 1 日起实行

内容	时间	内容	时间	内容	时间
起床	6:20	上学	7:10	散步	18:35—18:55
洗漱	6:25—6:40	放学	17:50	学习辅导	19:10—21:30
晨操 晨读	6:40—6:55 下雨时晨读	活动 洗澡	17:50—18:20 按顺序洗澡	洗漱	21:30—21:40
早餐	6:55—7:10	晚餐	18:20—18:35	就寝	21:50

(a) 作息时间表

泡/茶/方/法



撬茶 用茶针从整块茶中撬取 10g 左右的茶叶



投茶 将茶叶投入已经温好的杯中



醒茶 可使茶叶充分“苏醒”，以便后期冲泡迅速出味



泡茶 将沸水沿杯壁冲入杯中，第二泡时间不宜超过 10s，此后每泡时间递增 10s 即可



出汤 将茶水倒入公道杯中



分茶 将公道杯中的茶水按量分入品茶杯中

(b) 泡茶步骤

图 2.1.1 引领他人完成任务的方式



图 2.1.1 (续)

下指令者之所以采用表格、文字与图示等简单易懂方式来表达，是因为该问题需要被引领者自行解决，并且他们大多不熟悉问题解决所需的过程、步骤与安排。因此想要避免他人在理解并执行指令时出现错误或者无法完成任务，必须让指令表达尽可能地清晰、有序、完整与可执行。

2. 指挥机器执行的算法

如今使用机器替代人来解决问题，已经是一种非常普遍的现象。那些用来指挥机器解决问题或执行任务所需的一系列步骤，便是算法 (algorithm)。想要机器能够正确执行人给定的任务，就必须从它的视角来设计并描述可执行的算法。

比如让你去指挥简易计算器来完成指定的“ $5+6=?$ ”这一计算任务，就可以将执行步骤描述成如图 2.1.2 所示的内容。

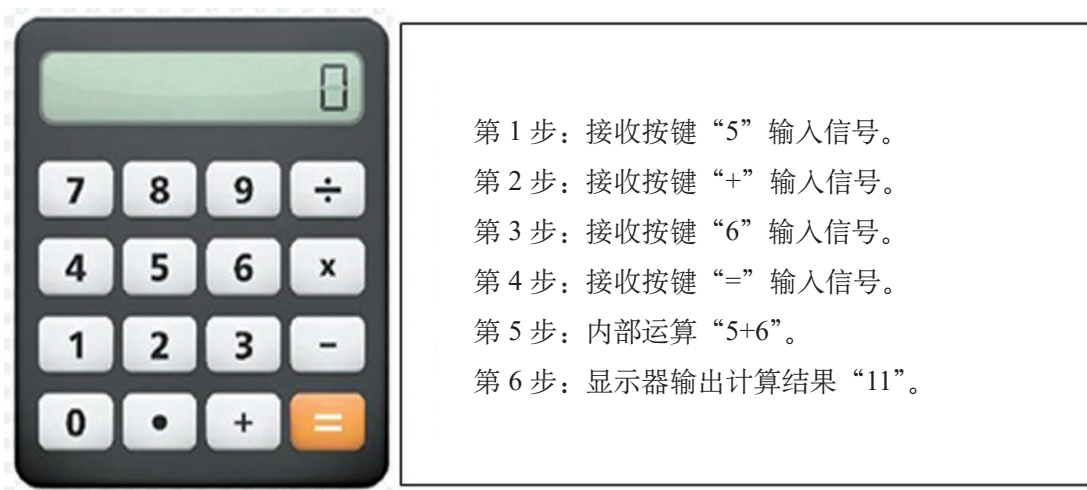


图 2.1.2 指挥简易计算器计算“5+6=？”

问题讨论

请以图 2.1.2 所示的算法为例，讨论“指挥机器”与“引领他人”有哪些相似之处？

二、算法特征与设计描述

算法由一系列步骤组成，但一系列步骤组成的却不一定是算法，因为算法必须具备一些基本要求。

1. 算法的特征

一个科学、完整的算法，首先必须要有输入（或预设初始值）与输出这两个步骤，其次要规定所有的步骤必须是明确的、可以执行的并且不能无限执行的指令，从而确保任务能够在一定时间内完成。也就是说，算法要具备输入、输出、确定性、可执行性及有穷性这五大特征，其具体内涵与简易计算器的实例解析如表 2.1.1 所示。

表 2.1.1 算法特征及其解析

特征	具体内涵	实例解析
输入	必须有预设初始值或输入数值	必须要输入“5”“+”“6”“=”的数据，否则计算器不知道要计算什么
输出	必须有一个或多个数据输出	必须输出计算结果“11”，否则计算没有意义，用户也不知道计算是否结束
确定性	要有明确的执行对象及行为	确定要从按键处依次接收到“5”“+”“6”“=”的数据，否则不知道怎么做
可执行性	每个步骤都必须可以执行	要确保图 2.1.2 中的 6 个步骤都可以执行，否则无法完成任务
有穷性	执行步骤的次数必须有限	必须在有限的步骤内完成并结束任务，如果步骤无限，将永远无法结束

问题讨论

计算机的算法，为什么必须要具备这五大特征？

2. 算法的设计

在需要计算机解决的众多问题中，有些是属于常见的特定问题。如排序、查找与插入等问题，人们很早就对它们开展研究，并提出多种可靠、高效的算法。当我们再次遇到这类问题时，就可以采用先学习再评价的方式来从中挑选，而无须再另行设计。

还有一些是属于非特定或是个性化的问题，则通常需要我们自行设计算法。简单的算法设计过程，如图 2.1.3 所示。一般是先从理解问题入手，即要根据解决问题的目标，来明确输入、输出与约束条件。以图 2.1.2 中的算法为例，向简易计算器输入的是“5”“+”“6”和“=”，其输出的是计算结果，约束条件是必须依次接收输入的“5”“+”“6”和“=”。然后再用适当方式对解决该问题的步骤进行描述。最后还要通过编写代码方式，来对算法进行实践检

验、评估与优化等。

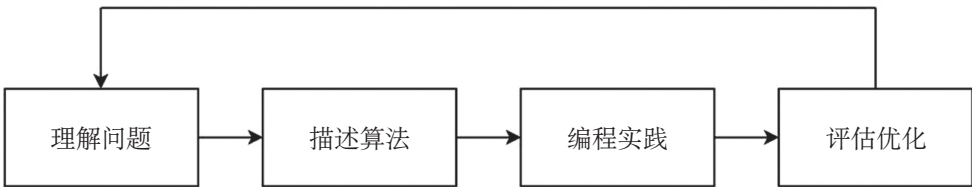


图 2.1.3 简单的算法设计过程

3. 算法的描述

跟引领他人解决日常生活问题的指令表达类似，计算机算法也有一些约定成俗的描述方式。用以保障你所描述的内容，能够被他人所理解，并促进算法自身的交流与提升。常用来描述算法的方式，有自然语言、流程图与伪代码等。

这三种描述方式，有各自的优势与不足。以计算圆面积算法为例，如图 2.1.4 所示。用自然语言描述，具有简单、通俗易懂的优点，但也容易产生歧义而给后续编写代码带来不必要的困扰，比如“计算面积”这一步，就没有给出具体的计算公式；用流程图描述则有清晰、直观与简洁的优点，虽然画图比较费时，但对专业性要求不高，因而较受人欢迎；而伪代码则是一种近似程序代码的算法表达方式，它方便程序员快速编程实现，但专业性要求相对较高，比如算法中的每条指令，都必须占用一行并用编号来表示步骤等。

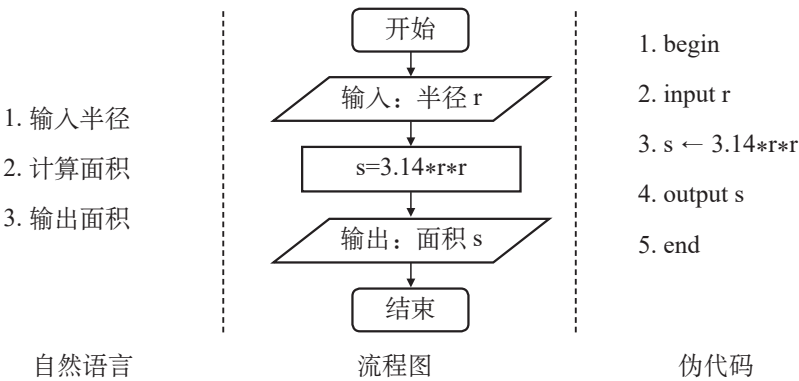


图 2.1.4 计算圆面积的三种算法描述



实践活动

用流程图描述求解绝对值的算法

绝对值是指一个数在数轴上的对应点到原点的距离，正数和 0 的绝对值是它本身（即 $|x|=x$ ），而负数的绝对值则是它的相反数（即 $|x|=-x$ ）。如果让计算机来自动求解，则必须设计相应的算法。

请用流程图描述计算机求解一个数的绝对值的算法。流程图的符号如表 2.1.2 所示。

表 2.1.2 流程图的符号

名称	符号	意义	范 例
起止框		程序的开始或结束	<div>开始</div> <div>结束</div>
处理框		对数据的处理	$r \leftarrow 5$
流程线		流程的进行方向	
输入 / 输出框		数据的输入或输出	<div>输入运算数</div> <div>输出计算结果</div>
判断框		对某个条件进行判断	$r \geq 0?$

实践内容主要包括：

- (1) 确认计算机自动求解的基本步骤。
- (2) 明确输入、输出与相关的约束条件。
- (3) 绘制流程图并与邻桌同学进行交流。

生活经验告诉我们，解决同一个问题，往往会有多种方案。比如想要从贵阳前往遵义，就有如表 2.1.3 所示的多种出行方案，并且它们的耗时与出行

成本各不相同。如果你对时间与出行成本比较在意，那么火车将是一种最优的选择。

表 2.1.3 从贵阳到遵义的几种出行方案

出行方案	耗 时	出行成本
汽车（出租车，非高速）	约 3 小时	最高
汽车（自驾，高速）	约 2 小 15 分钟	较高
火车（高铁）	约 2 小时	较低
自行车	约 12 小时	很低

当你选择运用计算机解决问题时，也同样会面临算法选择的挑战，因此需要对相关的各种算法进行评价。评价的主要指标有正确性、可读性、健壮性、高效性与低存储性等。在这些指标中，高效性因容易被人们直观感受而备受关注，它主要体现在用算法指挥计算机的执行次数方面。

以如图 2.1.5 所示的两种算法为例，在实现运用计算机解决“在 1 ~ 50 范围内，猜中 50 这个数字”这个问题过程，用“折半猜测”算法只要执行 6 次就猜中，而用“逐个猜测”算法则需执行 50 次。因此在解决该问题上，算法 2 要比算法 1 来得高效。

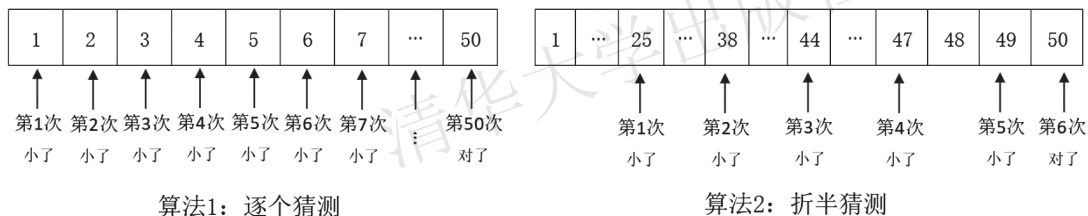


图 2.1.5 “猜数游戏”的不同算法比较

项目实施

小清团队准备使用流程图来描述要开发的软件的算法。
你会采用哪种方式描述算法，请填写在下面的横线上。

第2节 人机的互动

本节知识

- ◆ 计算机语言与集成开发环境
- ◆ 程序的输入与输出
- ◆ 程序的注释与规范

本节活动

- ◆ 语法高亮的意义分析
- ◆ “你问我答”编程体验

人机互动就是人与计算机的交互，主要作用是计算机理解并执行人的命令与要求。它的实现，不仅要依赖计算机的输入输出设备，还要有配套的程序代码来完成。程序代码就是用计算机语言编写的算法，它经过编译之后才成为大家熟知的软件。

一、计算机语言与集成开发环境

用自然语言、流程图及伪代码来描述的算法，仅适用于设计者的想法构思和人与人之间的展示交流，却不能让计算机直接理解并自动执行。因此最终还是要通过计算机能听懂的语言来描述并表达。

1. 计算机语言

就像人类之间是通过自然语言进行思想交流那样，人与计算机之间也有沟通的工具，它就是计算机语言。计算机语言经历了一场由难到易、由少到多的发展历程，并由此影响到计算机软件行业的发展。

在计算机的发展初期，人们直接使用二进制数来编写程序，这时计算机语言被统称为机器语言。二进制数表达的程序代码又长又难识记，只有极少数人才能编程，因此人们就发明出用文字符号来替换的汇编语言。汇编语言解决了

程序代码难以识记的问题，但还是因为其语法奇特而不被一般人所理解。直到类似人类自然语言的计算机高级语言出现，才让编程真正走进普通大众。因为它的语法跟人类自然语言相近，所以用它编写的程序代码容易被大家理解并接受。计算机高级语言的种类繁多，当前使用较为广泛的要属 Python、Java、C/C++ 与 JavaScript 等。

拓展阅读

Python 语言

Python 是一种使用广泛的计算机高级语言，如图 2.2.1 所示。它具有免费、开源、功能强大并支持跨平台应用等优势。其语法简单又灵活，既容易让初学者迅速掌握，又可以为专业程序员提供各种便利，因此广受大家的欢迎。

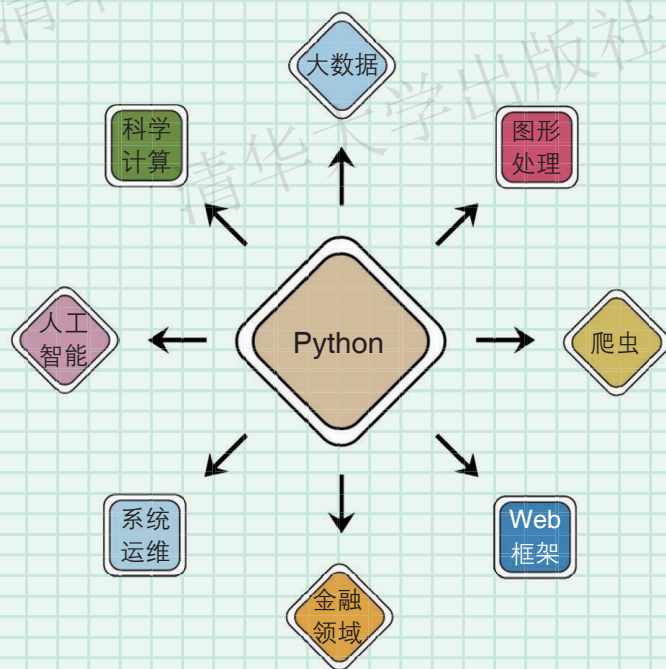
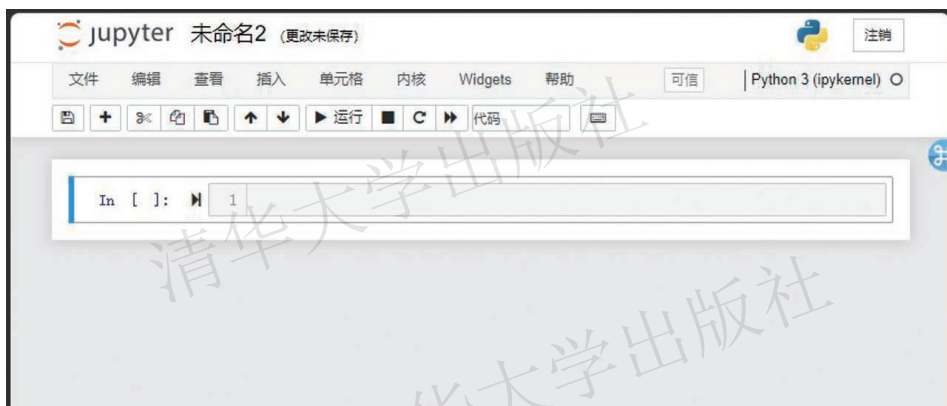


图 2.2.1 用途广泛的 Python

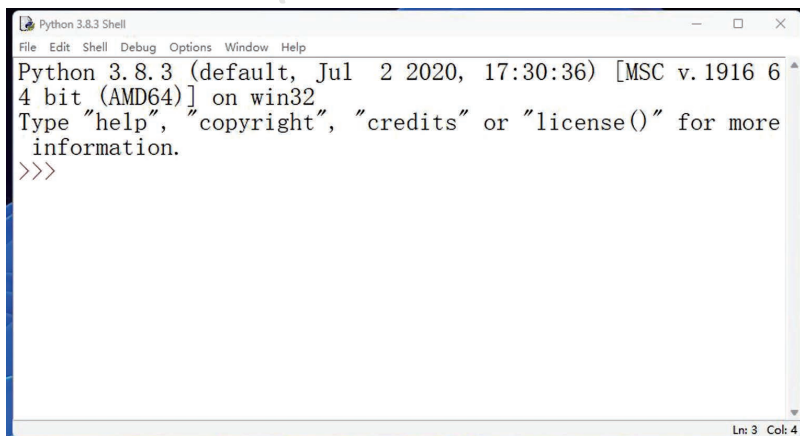
2. 集成开发环境

为了方便编程，人们还研发了专为其提供服务的应用软件——集成开发环境（integrated development environment，IDE），它集成了程序开发过程中所需要的编码、编译、调试和运行等服务功能。

每种计算机语言都会有各自配套的 IDE 软件。有些计算机语言的适用 IDE 软件较多，这时就要根据自身的需求来进行选择。如 Python 除了自带的 IDLE 以外，还会有 Jupyter Notebook、Thonny、Spyder、PyCharm 等多种。不同的 IDE，大多会有不同的功能与界面。以 Python 的 IDLE、Jupyter 与 Thonny 为例，它们的界面如图 2.2.2 所示。

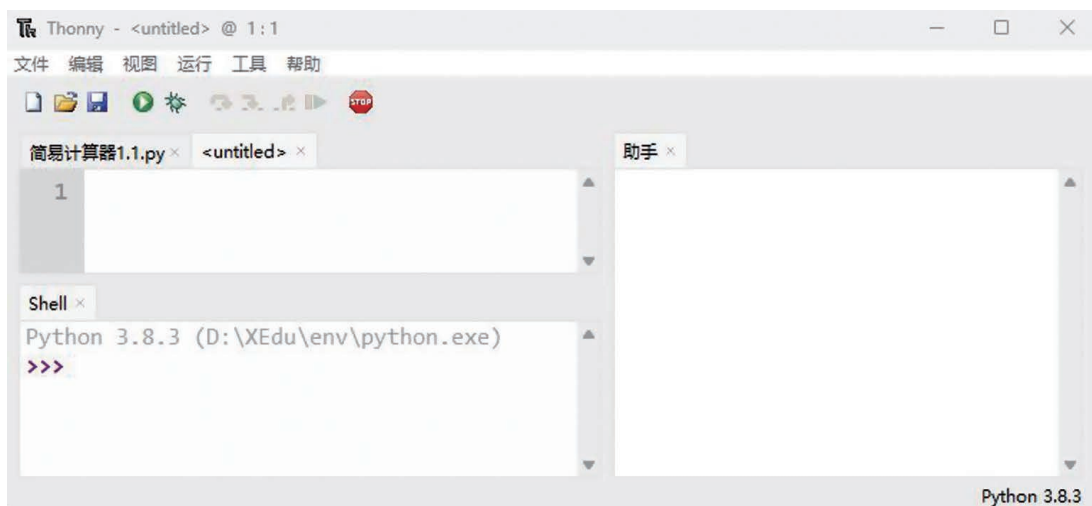


Jupyter Notebook 界面



IDLE 界面

图 2.2.2 支持 Python 的集成开发环境



Thonny 界面

图 2.2.2 (续)

二、程序的输入与输出

不同的计算机语言在实现程序的输入与输出方面可能会有所不同。以 Python 为例，Python 主要有两种输入、输出方式。一是用 Python 的命令解释器 (interpreter) 直接输入、输出。如图 2.2.3 所示，先在 Thonny 的 Shell 窗口中输入“3+5”，然后按回车键，就可以直接输出运算结果。二是通过 `input()` 与 `print()` 函数来实现，这在 Thonny 的文本编辑器（单击“运行”按钮或使用快捷键 F5）和 Shell 里都可以使用，如图 2.2.4 所示。

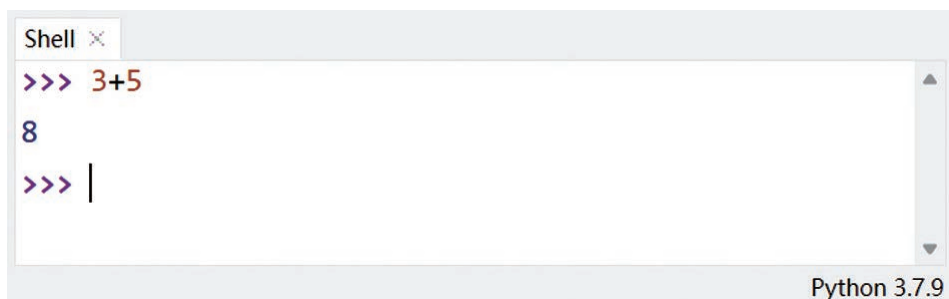



图 2.2.3 直接输入与输出



```

2.2.4 input()与print().py
1 info = input('输入的内容是: ')
2 print('输出的内容是: ', info)

Shell
Python 3.7.9 (bundled)
>>> %Run '2.2.4 input()与print().py'
输入的内容是: python
输出的内容是: python
>>>
Python 3.7.9

```

图 2.2.4 通过 input() 与 print() 实现输入与输出

问题讨论

编写程序时，编辑器会以不同的颜色显示程序的不同部分，这种功能称为语法高亮。对编程者而言，语法高亮有什么意义？

拓展阅读

input() 与 print() 常用格式

输入语句格式: `input(['输入提示内容字符'])`

功能: 读取从键盘输入的字符串，并以回车符作为结束。['输入提示内容字符'] 通常表示该参数为可选。

输出语句格式 1: `print(数据 1, 数据 2, ...)`

功能: 可以一次性输出多个内容对象，不同对象参数之间用英文半角空格分隔。如使用输出语句 `print(1, 2, 3)`，可以输出“1 2 3”。

输出语句格式 2: `print(数据 1, 数据 2, ..., sep='数据间分隔符')`

功能: 可以一次性输出多个内容对象，不同对象参数之间用 `sep` 参数指定的数据间分隔符分隔。如使用输出语句“`print(1, 2, 3, sep='+')`”，可以

输出“1+2+3”。

输出语句格式3: `print(数据, end='行末分隔符')`

功能: 输出数据, 并输出行末分隔符 (如没有 `end` 参数, 默认输出换行符)。如果使用下面几行代码, 则会输出以下结果。

代码:

```
print(1)
print(2)
print(3)
print(1, end=',')
print(2, end=',')
print(3)
```

输出结果:

```
1
2
3
1,2,3
```



体验活动

“你问我答”的编程体验

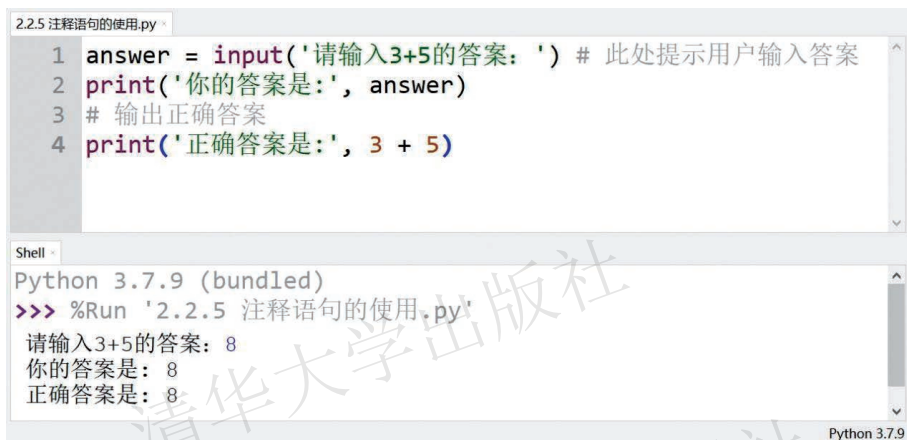
输入与输出是人与计算机开展互动交流的一种形式。实现“你问我答”的人机互动功能, 要通过编程的方式来实现。

请在 Thonny 中编写、调试并运行如下 Python 代码, 体验“你问我答”程序的实现过程。

```
answer = input('请输入 3+5 的答案: ')
print('你的答案是: ', answer)
print('正确答案是: ', 3+5)
```


三、程序的注释与规范

为了增强代码的可读性，编写者通常要在代码中插入注释语句。不同的计算机语言可能会有不同的注释方法。比如 Python 中最简单的注释方法，就是在代码中插入以“#”开头的语句，如图 2.2.5 所示。代码中的注释语句既不被执行，也不影响程序的正常运行。



```

2.2.5 注释语句的使用.py
1 answer = input('请输入3+5的答案: ') # 此处提示用户输入答案
2 print('你的答案是:', answer)
3 # 输出正确答案
4 print('正确答案是:', 3 + 5)

Shell
Python 3.7.9 (bundled)
>>> %Run '2.2.5 注释语句的使用.py'
请输入3+5的答案: 8
你的答案是: 8
正确答案是: 8
Python 3.7.9
    
```

图 2.2.5 注释语句的使用

编写程序代码时，还必须要了解计算机语言的规范性要求。以编写 Python 程序代码为例，新手特别容易犯“缩进不规范”“使用非法字符”等错误。

“缩进”是 Python 用来识别不同行代码之间主从关系的标志，没有主从关系就必须行首对齐，否则就会如图 2.2.6 所示出现调试出错。



```

2.2.6 代码缩进不规范.py
1 answer = input('输入3+5的答案: ')
2 print('你的答案是:', answer)
3 # 缩进不规范
4     print('正确答案是:', 3 + 5)

Shell
return compile(source, filename, mode, PyCF_ONLY_AST)
File "C:\Users\makespace01\Desktop\2.2.6 代码缩进不规范.py", line 4
    print('正确答案是:', 3 + 5)
    ^
IndentationError: unexpected indent
Python 3.7.9
    
```

图 2.2.6 代码缩进不规范

Python 代码主要由英文符号组成。如果是在非英文输入法状态下编写程序，就会出现“使用非法字符”的错误。如图 2.2.7 所示，只需将代码中的中文单引号修改为英文单引号，即可正常调试运行。



图 2.2.7 使用非法字符

随着学习的深入，我们还会接触更多关于程序代码编写规范的知识。

项目实施

小清团队准备选用 Python 语言与 Thonny 软件开发自己的软件。
你会选择哪一种计算机语言与 IDE 软件？请填写在下面的横线上。

第3节 信息的处理

本节知识

- ◆ 数据类型及其运算
- ◆ 变量与赋值

本节活动

- ◆ 字符串型与浮点型数据的理解
- ◆ 算术运算表达式的验证实验
- ◆ Python 变量名规范性分析
- ◆ 矩形周长计算程序的编程实践

从诞生之初为了实现自动计算，到后来能够实现文字、图片及声音等信息的自动处理，计算机在功能方面有了巨大的突破，并推动了信息社会的形成。究其原因，在于人们找到了实现信息编码及其运算处理的方法。

一、数据类型及其运算

计算机中的所有信息都是以二进制数据形式编码并存放在计算机中，以便程序进行读取与运算等操作。为了帮助程序识别不同类型的信息并实现高效的运算，多数计算机语言都会制定相应的数据类型与配套的运算规则。

1. 数据类型

计算机语言大多从计算机的角度细分生活中的信息，其定义的数据类型也都大同小异。以 Python 为例，数值信息就被其细分为整型与浮点型等，而文字符号信息则被定义成字符串型，另外还有专为表示真假信息而定义的布尔型（见表 2.3.1）等，与其他计算机语言相差不大。

表 2.3.1 Python 的部分数据类型

数据类型	实例表示
整型 (int)	数学中的整数，如-1，0，2
浮点型 (float)	数学中的小数，如 2.0，3.14
字符串型 (str)	用各种英文引号作为界定，如 '1'，"abc"，"12"
布尔型 (bool)	只定义真和假两种数值，真为 True，假为 False

问题讨论

生活中的各种票据，通常会印有使用数字来表达的信息，如图 2.3.1 所示的发票代码、发票号码与票价等。虽然同为阿拉伯数字，但人们都可以识别与区分它们的具体含义。如果将这些信息交给计算机来处理，却通常要用字符串型来存储发票代码与发票号码信息，而另用浮点型或整型来存储票价信息，请分析并讨论具体原因。



图 2.3.1 票据中的数字信息处理

2. 算术运算

计算机在实现整型或浮点型数据运算时，必须遵循数学的算术运算规则。以 Python 为例，其部分运算符及实例表示如表 2.3.2 所示。

表 2.3.2 Python 的部分算术运算

运算符	名 称	示 例	输 出	优先级别
+	加法运算符	<code>print(3 + 2)</code>	5	3
-	减法运算符	<code>print(3 - 2)</code>	1	3
*	乘法运算符	<code>print(3 * 2)</code>	6	2
/	除法运算符	<code>print(3 / 2)</code>	1.5	2
//	整除运算符	<code>print(3 // 2)</code>	1	2
%	求余运算符	<code>print(3 % 2)</code>	1	2
**	乘方运算符	<code>print(3**2)</code>	9	1

拓展阅读

Python 代码中的整除、求余、乘方与运算优先级别

整除、求余与乘方都是数学中常见的算术运算。在 Python 代码中，整除通常是指两个整数相除之后，取商的整数部分；求余通常是指两个整数相除后取余数；乘方通常是指若干个相同因数相乘的运算。而运算优先级别是指在一个 Python 表达式中，不同运算符的先后运算执行顺序。高级别的运算符会先于低级别的执行，若运算符级别相同时，则按照从左到右顺序进行依次运算。



实验活动

算术运算表达式的验证

算术运算中的优先级别是一种统一的约定，级别高的先运行，级别低的后运行。由于书写方式与数学有所不同，新手在编写较为复杂的算术表达式代码时，常因为容易犯错而需对其进行验证。请

完成算术运算表达式的验证活动，并填写表 2.3.3。

实验内容：验证较复杂的算术运算表达式。

实验准备：安装 Python 集成开发环境的计算机、复杂算术表达式。

表 2.3.3 较复杂算术运算表达式验证记录表

数学算式		Python 表达式		结论及建议
式子	答案	代码	运行	
$\frac{2+1}{1+2}$	1	<code>print(2+1/1+2)</code>	5.0	错误。正确代码为 <code>print((2+1)/(1+2))</code>

3. 字符串运算

字符串运算是处理字符串型数据的运算。简单的字符串运算有连接、重复及判断等，相关运算符及其实例表示如表 2.3.4 所示。

表 2.3.4 Python 的部分字符串运算

运算符	作用	实例表示	输出
+	连接	<code>print('中'+'国')</code> # 实现两个字符串连接	中国
*	重复	<code>print('H' * 2)</code> # 重复输出字符串	HH
in	判断	<code>print('t' in 'steven')</code> # 判断 't' 是否在 'steven' 中，如果是则为 True，否则为 False	True

字符串还有转义、格式化和切片等“运算”。

在 Python 中，可以用 “\” “\” “\” 分别输出单引号、双引号和右斜杠，其中 “\” 表示转义字符。如使用代码 “`print('\ ' '\')`”，可以输出 “' " \””。常用的还有 “\n” 表示换行，“\r” 表示光标移到行首。

Python 提供了多种字符串格式化的方法，`str.format()` 是其中之一。如语句

“`r = '{} + {} = {}'.format(3, 4, 7)`”中，“`.format`”前是字符串模板，紧跟的括号中是用逗号分隔开的参数列表。字符串模板中的每对“{}”是一个占位符，由参数列表中的参数依序替换（见图 2.3.2），对字符串进行格式化后，`r` 的值为“`3 + 4 = 7`”。也可以在大括号内标出参数序号，占位符将被相应序号的参数替换，如使用语句“`r = '{0} + {1} = {2}'.format(3, 4, 7)`”后的结果也是 `r` 的值为“`3 + 4 = 7`”。

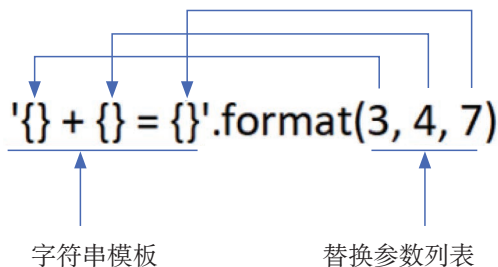


图 2.3.2 str.format() 字符串格式化示意图

字符串格式化的结果是一个字符串，常常与输出语句一起使用（见图 2.3.3）。

```
In [1]: 1 r = '{} + {} = {}'.format(3, 4, 7)
        2 print(r)
3 + 4 = 7

In [2]: 1 print('{} + {} = {}'.format(3, 4, 7))
3 + 4 = 7
```

图 2.3.3 输出语句中使用格式化字符串的两种方法

字符串模板中的占位符，还可以设置成各种格式，表 2.3.5 列出了部分数值格式。

表 2.3.5 字符串格式化中的部分数值格式

数 值	格 式	输 出	描 述
1	{:.2f}	1.00	保留小数点后两位
-1	{:+.2f}	-1.00	带符号保留小数点后两位
1	{:0>2d}	01	数字补零（填充左边，宽度为 2）
1	{:x<4d}	1xxx	数字补 x（填充右边，宽度为 4）

字符串的切片就是获取字符串的一部分的操作，语句常用格式举例如表 2.3.6 所示。

表 2.3.6 字符串切片格式举例

字符串	举 例	返回值
s='01234567'	s[0]	'0'
	s[3]	'3'
	s[::]	'01234567'
	s[0:8:1]	'01234567'
	s[1:7:1]	'123456'
	s[1:7:2]	'135'
	s[1::]	'1234567'
	s[:7:]	'0123456'

拓展阅读

数据类型与运算符的匹配要求

通常情况下，Python 的数据类型要与其运算符相匹配，否则可能会出现错误，如图 2.3.4 所示。

```

2.3.2 数据类型与运算符不匹配.py
1 print(1 + '2')    #1为整型，'2'为字符串

Shell
Python 3.7.9 (bundled)
>>> %Run '2.3.2数据类型与运算符不匹配.py'
Traceback (most recent call last):
  File "C:\Users\makespace01\Desktop\2.3.2数据类型与运算符不匹配.py", line 1, in <module>
    print(1 + '2')    #1为整型，'2'为字符串
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>>
Python 3.7.9

```

图 2.3.4 数据类型与运算符不匹配

二、内存数据的访问与修改

凡是程序能够直接进行运算的各种类型数据，均存放在计算机的内存中。这些数据可能会分布在不同的区域，但为了让程序能够既准确又方便地实现对它们的访问与修改，计算机语言就引入了变量这个概念。

1. 变量

变量可以简单地理解成是指向计算机内存某个区域的名称，这样可以方便程序员用它来指代该区域所存储的数据。由于程序在运行过程中会经常访问内存不同区域的数据，因此一个程序代码通常会有很多个变量。但为了区分这些变量的各自用途并增强代码的可读性，计算机语言都会对变量进行命名并制定相应的命名规则。不同计算机语言，其变量命名规则可能会略有差异。以 Python 为例，其变量名只能由字母（区分大小写）、数字、下划线或汉字（仅限于新版）组成，并且必须以字母或下划线开头，不能以数字开头。变量名不能与 Python 的关键字同名。

拓展阅读

Python 的关键字

关键字又称为保留字，它是计算机语言中具有特殊含义并被编程语言保留的单词，不能用作变量名或标识符。查看 Python 已有的关键字信息，可以通过 `help('keywords')` 命令进入帮助系统进行查看，如图 2.3.5 所示。

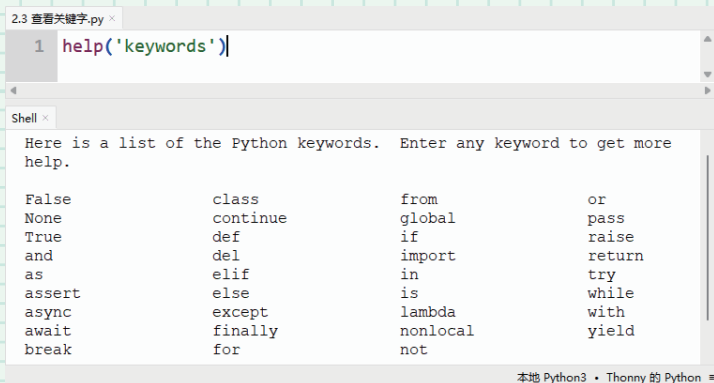


图 2.3.5 查看关键字

问题讨论

表 2.3.7 中的 Python 变量名是否规范？

表 2.3.7 变量命名规范性分析

变量名	规范性	不规范原因
False	<input type="checkbox"/> 是 <input type="checkbox"/> 否	False 为关键字
l_hao	<input type="checkbox"/> 是 <input type="checkbox"/> 否	
my name	<input type="checkbox"/> 是 <input type="checkbox"/> 否	
user_1	<input type="checkbox"/> 是 <input type="checkbox"/> 否	

2. 赋值语句

程序要修改变量中的数据，主要通过赋值语句来实现。赋值语句是计算机语言中最基本的语句，其作用是通过给变量赋值的方式，实现变量的定义或数据的更改。

Python 赋值语句的基本语法格式为：变量名 = 表达式。这里“=”就是最常见的赋值运算符，它可以把任意类型的数据赋值给变量，同一个变量可以被反复赋值。Python 中常见的赋值语句形式如表 2.3.8 所示。

表 2.3.8 Python 赋值语句的常见形式

代码实例	解 释
a = 10 a = a + 10	① 定义一个初始值为 10 的变量 a 或将变量 a 的值更改为整数类型 10 ② 先让变量 a 的值与 10 进行加法运算，再将该结果赋值给变量 a，使其值变为 20
a, b = 10, 20	等效于 a = 10 b = 20
a = b = c = 10	等效于 a = 10 b = 10 c = 10

Python 还提供了其他一些特殊的赋值运算符，如表 2.3.9 所示。

表 2.3.9 Python 部分赋值运算符

运算符	名 称	代码实例及解释
<code>+=</code>	加法赋值运算符	<code>c += a</code> 等效于 <code>c = c + a</code>
<code>-=</code>	减法赋值运算符	<code>c -= a</code> 等效于 <code>c = c - a</code>
<code>*=</code>	乘法赋值运算符	<code>c *= a</code> 等效于 <code>c = c * a</code>
<code>/=</code>	除法赋值运算符	<code>c /= a</code> 等效于 <code>c = c / a</code>



实践活动

矩形周长计算程序的编程实践

计算矩形周长，就是求矩形的四条边长度之和，常用的有如图 2.3.4 所示两种算法。请自选一种算法，并用 Python 语言编程实现。

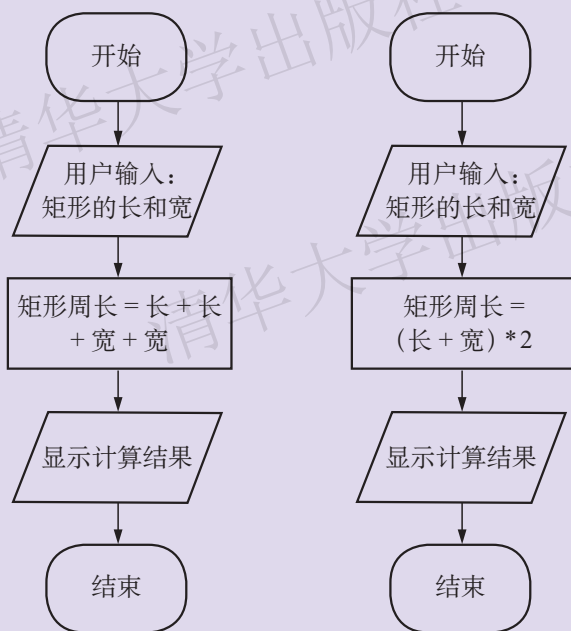


图 2.3.4 矩形周长计算流程图

实践内容主要包括：

- (1) 按照自选算法的解决问题步骤来编写、调试并运行相关 Python 代码。
- (2) 观察输出结果并与邻桌同学进行展示与交流。

项目实施

小清团队为要开发的软件的程序代码确定了数据存储类型、变量名及加减法运算表达式样式等，如评委给的分数用浮点型存储，变量名统一用大写字母开头并用英文标注它的用途等。

你为自己开发软件的程序代码确定了哪些具体内容？请填写在下面的横线上。

清华大学出版社
清华大学出版社

第 4 节 决策的实现

本节知识

- ◆ 关系运算表达式
- ◆ 逻辑运算表达式
- ◆ 单分支结构程序及其实现
- ◆ 双分支结构程序及其实现

本节活动

- ◆ 生活问题的关系运算表达
- ◆ 生活问题的逻辑运算表达
- ◆ 公交卡余额不足提示的程序实现

2021 年 5 月 15 日,“天问一号”成功登上火星。在约为 9min 的“至暗时刻”,着陆巡视器完全依靠自身计算机程序选择了最终的着陆点。这种实现自主判断并做出选择的技术背后,就有条件表达式与选择语句的“身影”。

一、条件表达式

条件表达式就是用来描述问题解决条件的程序代码样式,它是帮助计算机实现自主判断的基础。其类型有关系运算表达式与逻辑运算表达式两种,并且运算结果永远只是一个布尔值,即不是 True 就是 False。

1. 关系运算表达式

关系运算表达式较为简单,用关系运算符即可进行表达。以 Python 为例,其表达规则及输出如表 2.4.1 所示。

表 2.4.1 Python 的关系运算表达式

运算符	表达式	描 述	实例输出	备 注
>	$x > y$	x 大于 y	$1 > 2$ 结果为 False	仅限于相同类型对象的比较
<	$x < y$	x 小于 y	$1 < 2$ 结果为 True	
>=	$x \geq y$	x 大于或等于 y	$1 \geq 2$ 结果为 False	
<=	$x \leq y$	x 小于或等于 y	$1 \leq 2$ 结果为 True	
==	$x == y$	x 等于 y	$1 == 2$ 结果为 False	对象的类型不受限
!=	$x != y$	x 不等于 y	$1 != 2$ 结果为 True	



实践活动

生活问题的关系运算表达

2023 年 1 月 1 日起，儿童铁路购票采用新的规则，具体如图 2.4.1 所示。这种新规条件可以用关系运算表达式来表示，以帮助程序代码实现“当用户输入儿童年龄时，就可以用它来自动判断是否符合相应的优惠条件”功能。

实践内容主要包括：

- (1) 运用 Python 的关系运算表达式来表达儿童购票的条件。
- (2) 将代码向同桌进行展示交流。

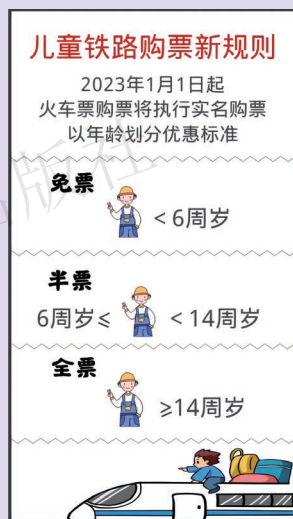


图 2.4.1 铁路儿童票新规则

2. 逻辑运算表达式

逻辑运算表达式用逻辑运算符来表示。逻辑运算符又称为布尔运算符，是专门用来处理布尔型数据的运算符，较为常用的是 and（与）、or（或）、not（非）这三种运算符。以 Python 为例，其表达规则及输出如表 2.4.2 所示。

表 2.4.2 Python 的逻辑运算表达式

运算符	表达式	实例输出
and	x and y	如果 x=True, y=True, 那么实例运算结果为 True
or	x or y	如果 x=True, y=False, 那么实例运算结果为 True
not	not x	如果 x=True, 那么实例运算结果为 False



实践活动

生活中的逻辑运算表达

国旗护卫是一项神圣而又光荣的任务。学校对参加国旗护卫队队员的要求如下，一是身高方面，男生高于 1.7m，女生高于 1.65m；二是政治面貌方面，必须为团员。这种筛选条件可以用逻辑运算表达式来表示，以帮助程序代码实现“当用户输入男女生条件时，就可以用它来自动判断是否符合条件”的功能。

实践内容主要包括：

- (1) 运用 Python 的逻辑运算表达式来表达国旗护卫队队员的筛选条件。
- (2) 将代码向同桌进行展示交流。

二、选择语句与分支结构程序

选择语句是用来描述条件判断并控制程序执行方向的计算机语句。由于它所编写的程序流程是按照条件选择性地执行其中某个分支，故称为分支结构程序。常见的有单分支结构程序和双分支结构程序。

1. 单分支结构程序及其实现

只要条件为真，就执行唯一的分支流程，否则退出，这种流程结构就是单分支结构程序。生活中有许多需要用单分支结构程序来实现自主决策的问

题，例如图 2.4.2 中的学校运动会跳高纪录的自主刷新问题。

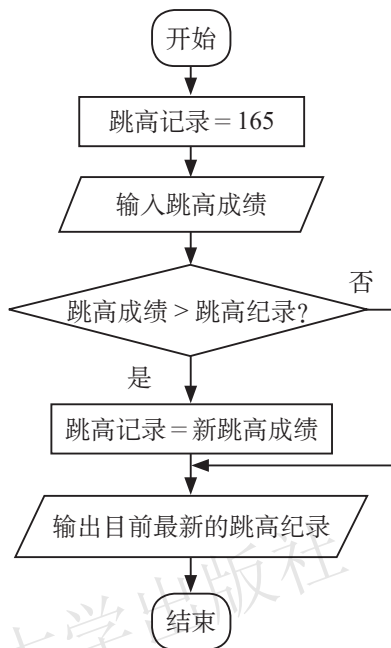


图 2.4.2 单分支结构的流程控制

用 Python 的 if 语句编写该程序，其实例代码与使用格式如图 2.4.3 所示。

<pre> record=165 height=int(input('请输入跳高成绩 (cm): ')) #int()可以将输入的数据转为整数型 if height>record: record=height print('目前的跳高纪录为',record,'cm') </pre>	<p>if 条件表达式： 语句块</p>
------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------

图 2.4.3 Python 的单分支结构程序实例代码及 if 语句使用格式

拓展阅读

Python 中 if 语句的规范性要求

if 是一个关键字，用于条件语句的编写，是分支结构程序的标志，后面紧跟着条件表达式及冒号。如果条件表达式的布尔值为 True，那么就执行下方的语句块，否则就直接退出。冒号则用于表示：下方所有相同缩进的代码行，都是归属于该分支结构程序的语句块。

2. 双分支结构程序及其实现

与单分支结构程序不同，双分支结构程序拥有两条分支流程选项。如果条件为真，就执行与之匹配的分支流程，否则执行另一条分支流程。如图 2.4.4 所示，电子密码锁的验证过程就是通过该结构程序实现的。

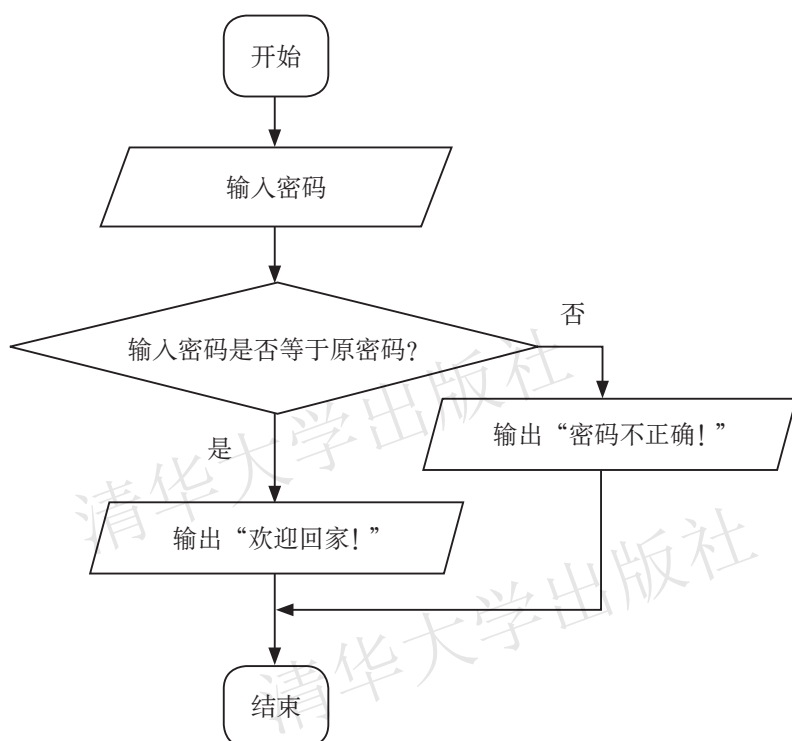


图 2.4.4 双分支结构的流程控制

用 Python 的 if-else 语句编写该程序，其实例代码与使用格式如图 2.4.5 所示。

<pre> password='hi' #初始密码 key=input('请输入密码: ') if key==password: print('欢迎回家!') else: print('密码不正确!') </pre>	<p>if 条件表达式：</p> <p> 语句块 1</p> <p>else:</p> <p> 语句块 2</p>
----------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------

图 2.4.5 Python 的双分支结构程序实例代码及 if-else 语句的使用格式



实践活动

公交卡消费提示程序的编写实践

公交车是人们日常出行的主要交通工具之一。为方便乘客及时了解公交卡上的余额信息，需要刷卡机自动提示如下文字信息：当余额小于10元时，系统会提示“请充值！”；余额大于或等于10元时，系统会提示“谢谢！”。请基于如图2.4.6所示的流程，并运用Python来解决该问题。

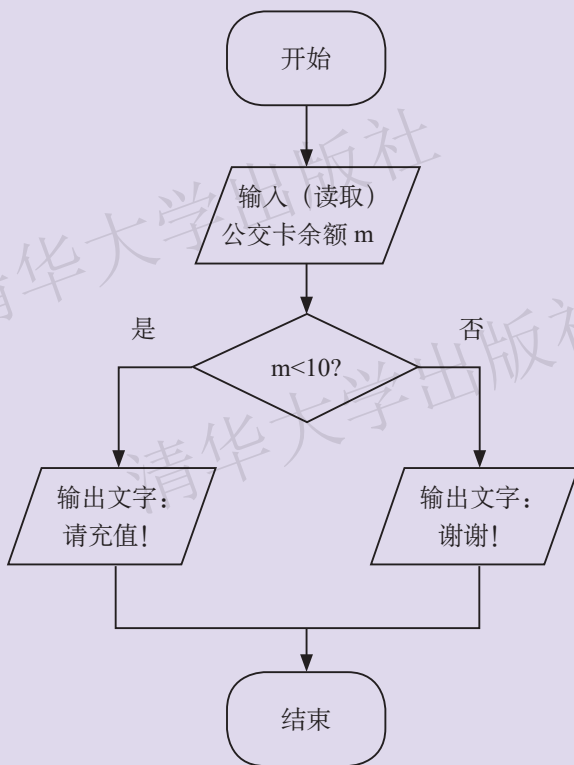


图 2.4.6 公交卡消费余额提示流程图

实践内容主要包括：

- (1) 要运用到 if-else 语句，并模拟实现公交卡消费的提示功能。
- (2) 编写、调试、运行相关代码并观察输出结果。
- (3) 将相关代码与输出结果向同桌进行展示交流。

项目实施

小清团队准备使用分支结构来解决程序的决策问题，如判断评委输入的分数是否在正确的区间内。

你会用分支结构程序解决哪些具体问题？请填写在下面的横线上。

清华大学出版社
清华大学出版社

第 5 节 循环的艺术

本节知识

- ◆ 循环结构程序
- ◆ 条件循环程序及其实现
- ◆ 遍历循环程序及其实现
- ◆ 序列

本节活动

- ◆ 循环体代码的分析
- ◆ 用户登录程序的实现
- ◆ 对车牌字母“O”与“I”的自动处理
- ◆ 查找最大值和最小值编程实践

计算机从事着人们生活中很多枯燥且无趣的工作，它让街上的红绿灯重复交替地亮起，学校的铃声总在固定的时间响起，门前的电子密码锁永不厌烦地检验用户的每次口令。计算机之所以能承担这些周而复始的工作，关键在于循环结构程序的实现。

一、循环结构程序

按照流程控制方向的不同，程序可以分为顺序结构、分支结构与循环结构三类。其中顺序结构只能按照固定顺序，自上而下一行不落地执行程序。循环结构则相对巧妙，它专为程序中需要重复执行的某个功能而设计。我们通常只需使用短短的几行代码，便可代替顺序结构程序不断重复的语句。常见的循环结构程序有条件循环与遍历循环两类。

二、条件循环程序及其实现

程序需要在一定条件下才能重复地执行某些功能，当条件不允许时会立即终止，这就是条件循环。在日常生活中，有很多可以归属为条件循环的问题。如图 2.5.1 所示，在超市里使用消费券结算便是条件循环问题。

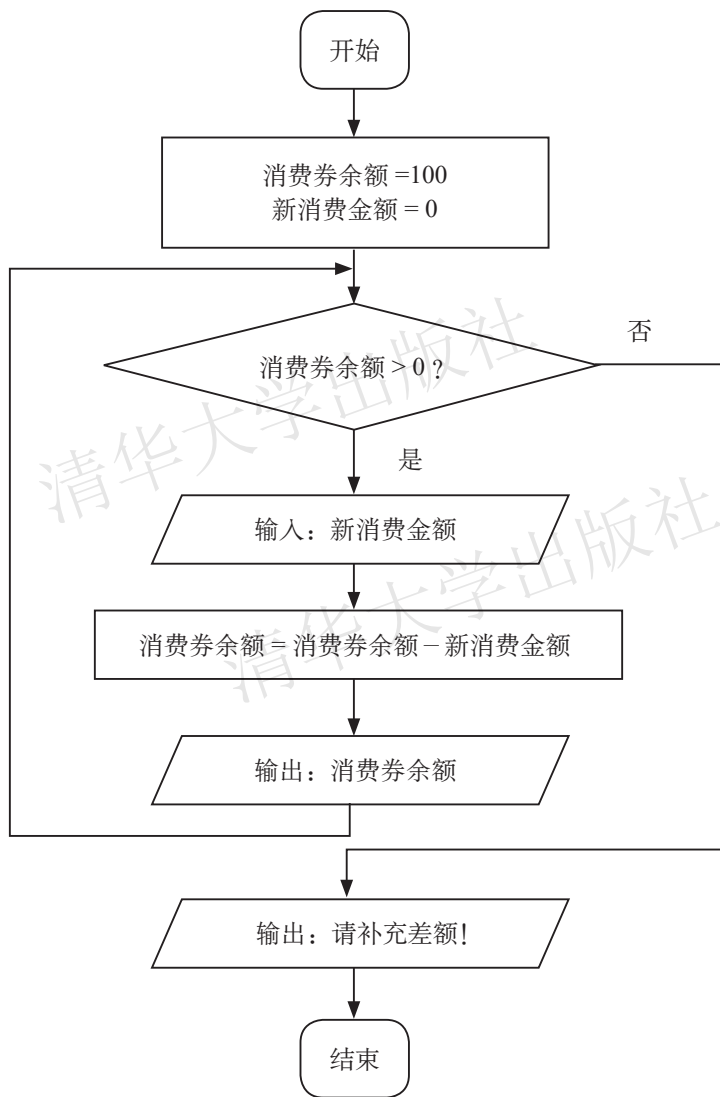


图 2.5.1 条件循环结构的流程控制

如果使用 Python 来实现这个过程，就会用到 while 语句。该过程实例及 while 语句的使用格式说明如图 2.5.2 所示。

<pre>Voucher=100 Consumption=0 while Voucher>0: Consumption=int(input('请输入新消费金额:')) Voucher=Voucher-Consumption print('当前消费券余额:',Voucher) print('请补充差额!')</pre>	while 条件表达式: 循环体
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------

图 2.5.2 while 语句的使用

问题讨论

while 语句中的循环体通常由多行代码组成。它除了要有描述执行指定功能的代码，通常还要有改变循环条件表达式布尔值的代码，如上述实例中的“Voucher=Voucher-Consumption”。如果在循环体中缺少了这种代码，会出现怎样的情形呢？



实践活动

用户登录程序的实现

出于安全考虑，大部分信息系统都会对用户的连续登录次数进行限制。如用户连续出现3次出错，则会提出“请稍后再试！”的警告，并不再提供登录界面；否则会出现“请再试！”的提示，并有登录界面。请基于如图2.5.3所示的流程，运用Python来解决该问题。

实践内容主要包括：

- (1) 运用条件循环和分支语句来模拟实现。
- (2) 编写、调试、运行相关代码并观察输出结果。
- (3) 将相关代码与输出结果向同桌进行展示交流。

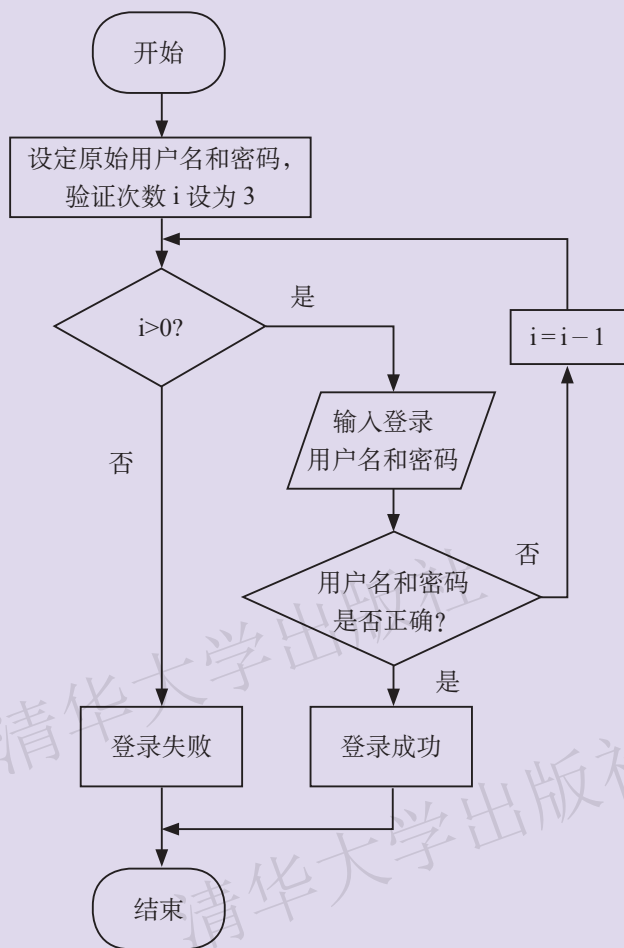


图 2.5.3 用户登录次数限制流程图

三、遍历循环程序及其实现

遍历循环是指程序对序列中的指定对象执行相同的操作，直到遍历为止。

1. 遍历循环程序结构

在计算机中，任何有序的数据集都可以看作是一个序列。比如字符串“中华人民共和国”由 7 个汉字组成，因为每个汉字都有自己的位置顺序，所以就可以看作是一个序列。遍历是指所有指定对象都要经历 1 次并且只执行 1 次

的操作过程。这类似于体育课，老师为了方便统计人数而要求队伍中的每个人报数，并且每个人只能报1次。

以依次逐行输出“中华人民共和国”字符串中的每个字为例，该遍历循环的流程图如图 2.5.4 所示。

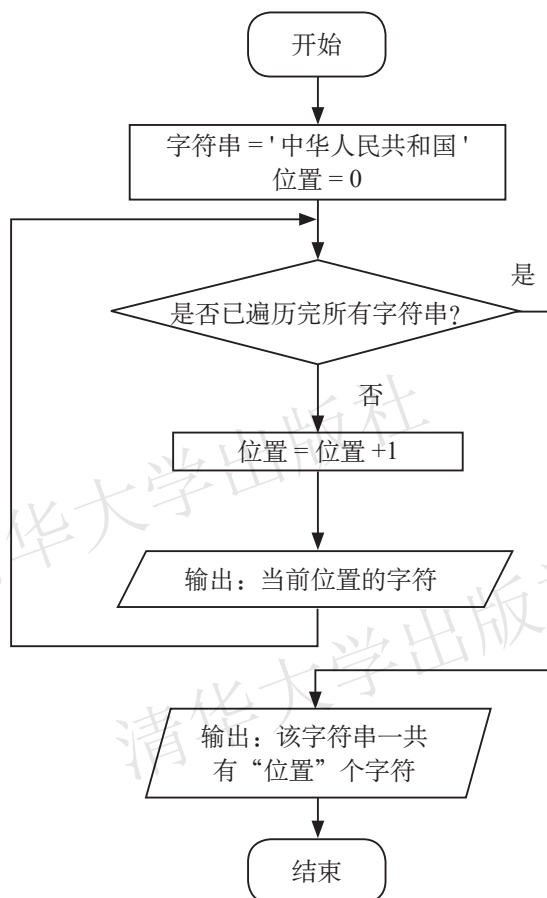


图 2.5.4 遍历循环结构的流程控制

在 Python 中实现遍历循环，要用到 for 语句。上述实例代码及 for 语句的使用格式说明如图 2.5.5 所示。

<pre> characterString='中华人民共和国' postion=0 for letter in characterString: postion+=1 print('第',postion,'个字符是:',letter) print(characterString,'字符串中共有',postion,'个字符') </pre>	<p>for 循环变量 in 序列:</p> <p>循环体</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------

图 2.5.5 for 语句的使用



实践活动

停车缴费系统的车牌字母“O”与“I”自动处理程序

停车场的自助缴费系统（见图 2.5.6）不仅节省了人员管理成本，更是大大提高了车辆通行效率。车主只需提前输入车牌号并支付停车费后，即可随时离场。为避免字母 O、I 与数字 0、1 混淆，管理部门规定车牌后 5 位编码不采用字母进行编号。但在现实中，车主却经常将字母 O 或 I 当作数字输入缴费系统，为此停车场缴费系统需要自动识别字母 O 或 I，并将其转化为“0”或“1”。请用 Python 来模拟实现这个过程。



图 2.5.6 停车场自动缴费系统

实践内容主要包括：

- (1) 用流程图设计算法。
- (2) 运用到遍历循环和分支语句。
- (3) 编写、调试、运行相关代码并观察输出结果。
- (4) 将相关代码与输出结果向同桌进行展示交流。

2. 遍历循环程序实现

除了字符串序列，最为常见的还有整数序列。比如可以借用 Python 的 `range()` 内置函数生成 0 ~ 4 的整数序列，代码及运行结果如图 2.5.7 所示。

```

2.5.5 range()函数的使用.py
1 for i in range(5):
2     print('range(5):',i)

Shell
Python 3.7.9 (bundled)
>>> %Run '2.5.5 range()函数的使用.py'
range(5): 0
range(5): 1
range(5): 2
range(5): 3
range(5): 4
Python 3.7.9
  
```

图 2.5.7 `range()` 函数的使用

列表 (list) 是 Python 中非常重要的内置数据类型。列表由一系列元素组成，所有的元素被包含在一对方括号中。列表被创建后，每个元素都有一个索引号 (索引号从左到右 0, 1, 2……编排或者从右到左 -1, -2, -3……编排)，通过索引号可以访问相应的元素值。比如我们可以把常用数学运算符存入列表 `ops` 中，并访问索引号为 0, 1, 2 和 -1, -2 的元素，如图 2.5.8 所示。

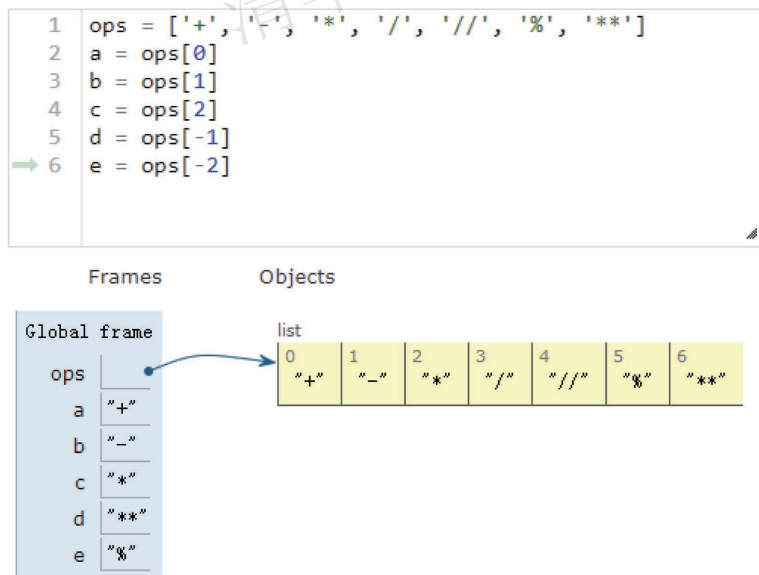


图 2.5.8 访问列表示例

通过对列表元素进行逐个访问可以完成列表遍历。一般可以用以下两种方法完成列表元素遍历：第一种方法用“for 变量 in 列表变量:”语句把列表中的元素按顺序赋值给变量，从而完成遍历；第二种方法借用“range()”函数产生的索引号来访问列表元素，完成遍历。示例代码及运行结果如图 2.5.9 所示。

```

1 # 创建名为ops的常用数学运算符列表
2 ops = ['+', '-', '*', '/', '//', '%', '**']
3 # 遍历列表的第一种方法
4 for op in ops:
5     print(op)
6 # 输出空行
7 print()
8 # 遍历列表的第二种方法
9 ops_len = len(ops)
10 for i in range(ops_len):
11     print(i, '->', ops[i])
12

```

Shell x

```

>>> %Run -c $EDITOR_CONTENT
+
-
*
/
//
%
**

0 -> +
1 -> -
2 -> *
3 -> /
4 -> //
5 -> %
6 -> **

>>>

```

图 2.5.9 遍历列表的两种方法示例

拓展阅读

常见的序列类型

Python 中常见的序列类型有字符串、列表、元组、字典和集合等（见表 2.5.1）。序列类型一般支持 len() 函数、in 操作符和遍历操作。字符串、列表和元组都可以根据索引号获取所在位置的元素值。字典可以根据

据“键”来获取该“键”所对应的“值”。不同序列类型各有特点，在程序设计过程中根据其优势和实际需要选用适当的序列类型。

表 2.5.1 常见的序列类型

类型	作用	说 明	示 例
str	字符串	由字符组成的不可变序列	'张三'
list	列表	可由各种类型组成可变序列	['张三','李四','王五']
tuple	元组	可由各种类型组成的不变序列	('张三','李四','王五')
dict	字典	由“键 - 值”对组成的可变序列，“键”具有唯一性	{'张三':13,'李四':12}
set	集合	值不可重复的可变序列	{2,3,5,7}

0 实践活动

查找最大值和最小值

为了保证比赛中评委计分的公平公正，通常是去掉一个最高分和一个最低分，再求平均分作为选手的最终得分，为此就要从所有的分数中找出最大值与最小值。现场比赛评分如图 2.5.10 所示。在校园现场赛的评分计算中，如何获取由若干整数组成的列表中的最大值和最小值？请用 Python 来模拟实现这一功能。

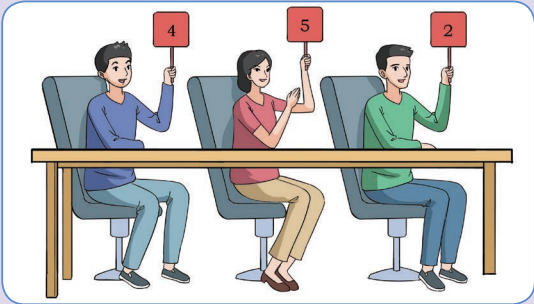


图 2.5.10 现场比赛评分

实践内容主要包括：

- (1) 用流程图设计算法。
- (2) 编写、调试、运行相关代码并观察输出结果。
- (3) 将相关代码与输出结果向同桌进行展示交流。

项目实施

小青团队使用 while 循环语句来解决程序的循环问题，如实现连续输入评委的打分。

你还会使用 while 或 for 循环语句解决哪些具体问题？请填写在下面的横线上。

第 6 节 指令的复用

本节知识

- ◆ 内置函数与自定义函数
- ◆ 模块与内置模块

本节活动

- ◆ 检验输入函数接收的数据类型
- ◆ 温标换算函数的编写实践
- ◆ 编写猜数字游戏程序

随着人们对计算机解决问题的研究不断深入，程序功能设计越来越复杂，所编写的程序代码也越来越多。于是一个非常有趣的问题出现了——人们能否像装配汽车那样快速地组装程序代码，而不必编写所有的程序代码？

一、可反复利用的代码段

我们在编写比较复杂的程序时，经常会发现算法中的有些步骤曾经有人编写过，并且可以被直接调用而无须从零开始。大部分的计算机语言都会提供这种可以被直接调用的代码段，其中最为常见的就是函数。

1. 函数

函数原本是数学中的概念，在计算机语言中指已经被编写完成的、具有一定功能的、可以重复利用的代码块。它可以通过简单的方式被反复调用，从而使代码结构变得更为简洁、清晰。函数主要有两大类，一类是计算机语言提供的内置函数，另一类则是我们自己开发的自定义函数。

2. 内置函数

内置函数是由计算机语言直接提供，用户无须知道内部情况就可以调用的函数，如前面已经接触过的 `input()` 与 `print()` 函数。常用的部分 Python 内置函数如表 2.6.1 所示。

表 2.6.1 Python 的部分内置函数

内置函数	示 例	返回值
<code>int(x)</code>	<code>int('20')</code> # 将字符串 x 转换为整数	20
<code>float(x)</code>	<code>float('20')</code> # 将字符串 x 转换为浮点数	20.0
<code>str(x)</code>	<code>str(20)</code> # 将数值 x 转换为字符串	'20'
<code>len(x)</code>	<code>len('Python')</code> # 返回字符串 x 的长度	6
<code>max(x)</code>	<code>max(2,5,11)</code> # 返回给定参数的最大值	11
<code>bool(x)</code>	<code>bool('True')</code> # 将 x 转换为布尔值	True
<code>chr(x)</code>	<code>chr(65)</code> # 返回编码 x 对应的字符	'A'
<code>ord(x)</code>	<code>ord('0')</code> # 返回字符 x 对应的编码值	48
<code>bin(x)</code>	<code>bin(15)</code> # 返回十进制数 x 的二进制字符串	'0b1111'
<code>oct(x)</code>	<code>oct(15)</code> # 返回十进制数 x 的八进制字符串	'0o17'
<code>hex(x)</code>	<code>hex(15)</code> # 返回十进制数 x 的十六进制字符串	'0xf'
<code>abs(x)</code>	<code>abs(-1)</code> # 返回 x 的绝对值	1
<code>divmod(x, y)</code>	<code>divmod(5, 3)</code> # 返回 x 被 y 整除的商和余数	(1, 2)
<code>pow(x, y)</code>	<code>pow(2, 3)</code> # 返回 x 的 y 次方的值	8

从表 2.6.1 可以看出内置函数的调用方式，其基本语法格式如下：

[变量名 =] 内置函数名 ([参数值列表])

拓 展 阅 读

查看 Python 的内置函数

通过 `print(dir(__builtins__))` 命令，可以查看 Python 的内置函数，如图 2.6.1 所示。





实验活动

检验输入函数接收的数据类型

Python 程序从键盘接收数据大多通过 `input()` 函数来实现，并通过赋值语句将其传递给变量。如果希望通过键盘传输整型数据，那么变量直接接收到的数据是否为整数呢？请运行如图 2.6.3 所示的代码，并将相关数据与实验结论填入表 2.6.2 中。

```

Thonny - C:\Users\BlueS\Downloads\2.6.3.py @ 4:39
文件 编辑 视图 运行 工具 帮助

2.6.3.py
1 data1 = input('请输入数据:')
2 print(type(data1)) # 输出变量对象data1的数据类型
3 data2 = int(data1)
4 print(type(data2)) # 输出变量对象data2的数据类型

Shell
Python 3.8.3 (D:\XEdu\env\python.exe)
>>> %Run 2.6.3.py
请输入数据:123
<class 'str'>
<class 'int'>
>>>
Python 3.8.3
    
```

图 2.6.3 程序代码

实验内容：利用 `type()` 函数检测变量接收输入函数传输的数据类型。

实验准备：安装 Python 集成开发环境的计算机、检验输入函数接收数据类型的 Python 代码。

表 2.6.2 变量数据类型检验记录表

实验次数	输入数据	输出		实验结论
		变量 a 的类型	变量 b 的类型	
1				
2				
3				
4				
5				

3. 自定义函数

自定义函数是用户自行开发的一段具有特定功能的、可反复调用的语句块。不同计算机语言的自定义函数的语法略有不同。以 Python 为例，其基本语法格式如下：

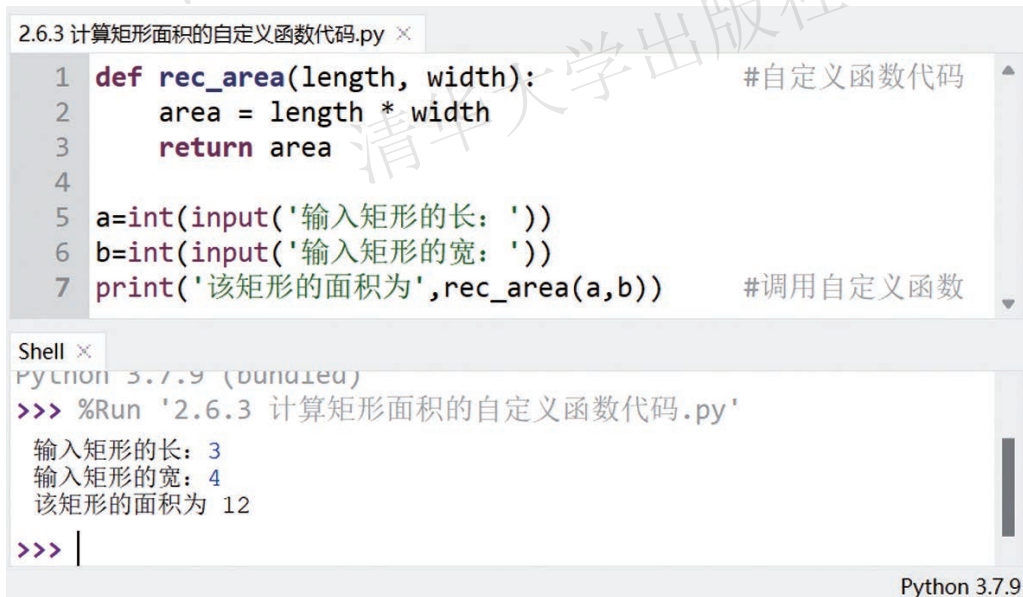
```
def 自定义函数名 ([ 参数列表 ]):  
    自定义函数语句块  
    [return [ 返回值 ]]
```

def 是自定义函数的标志，后面紧跟自定义函数名，其命名规则与变量命名规则相同。参数列表可省略或定义有默认值的参数。return 语句可以返回自定义函数的输出值，如果省略则返回 None。

自定义函数的调用与内置函数相似，其调用基本语法格式如下：

```
[ 变量名 = ] 自定义函数名 ([ 参数值列表 ])
```

以计算任何矩形的面积为例，自定义函数代码设计及其调用如图 2.6.4 所示。



```
2.6.3 计算矩形面积的自定义函数代码.py ×  
1 def rec_area(length, width): #自定义函数代码  
2     area = length * width  
3     return area  
4  
5 a=int(input('输入矩形的长: '))  
6 b=int(input('输入矩形的宽: '))  
7 print('该矩形的面积为',rec_area(a,b)) #调用自定义函数  
  
Shell ×  
Python 3.7.9 (bundled)  
>>> %Run '2.6.3 计算矩形面积的自定义函数代码.py'  
输入矩形的长: 3  
输入矩形的宽: 4  
该矩形的面积为 12  
>>> |  
Python 3.7.9
```

图 2.6.4 计算矩形面积的自定义函数代码



实践活动

温标换算函数的编写实践

如图 2.6.5 所示，摄氏度与华氏度都是世界上使用较为广泛的温标，不同地区的人经常要将其数值相互换算才能正常使用。

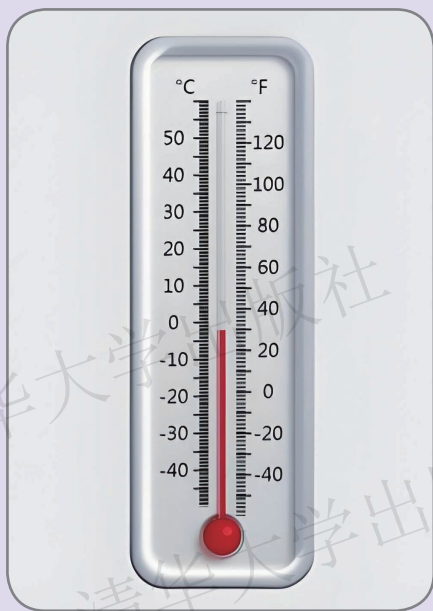


图 2.6.5 摄氏度与华氏度

摄氏温度转换成华氏温度的换算公式为

$$\text{华氏温度} = \text{摄氏温度} \times 1.8 + 32$$

华氏温度转换成摄氏温度的换算公式为

$$\text{摄氏温度} = (\text{华氏温度} - 32) \div 1.8$$

请用 Python 编写能够实现摄氏温度与华氏温度互相换算的函数，以方便人们直接调用。

实践内容主要包括：

- (1) 用流程图设计相关算法。
- (2) 编写、调试、运行相关代码并观察输出结果。
- (3) 将相关代码与输出结果向同桌进行展示交流。

二、可导入使用的代码文件

为了方便用户调用更多的代码片段，有些计算机语言还提供了一种称为模块的可反复利用的代码文件。

1. 模块

模块是复杂程序的重要组成部分，它通常是一个包含函数、变量等在内的独立程序代码文件。我们引用某一模块时，可以直接调用该模块中的函数与变量，使算法的代码实现变得更加容易，程序的结构也因此更加简洁。以 Python 为例，它为用户提供了内置模块与扩展模块两种类型。

2. 内置模块及其使用

内置模块是由计算机语言提供而无须用户自行定义的模块，使用时不仅要了解该计算机语言内置模块的功能，还要知道它的使用方法。

例如 Python 的 random 模块为用户提供了多种专用于解决随机性问题的函数（因此被称为随机数模块），极大地方便了用户算法的代码实现，如表 2.6.3 所示。

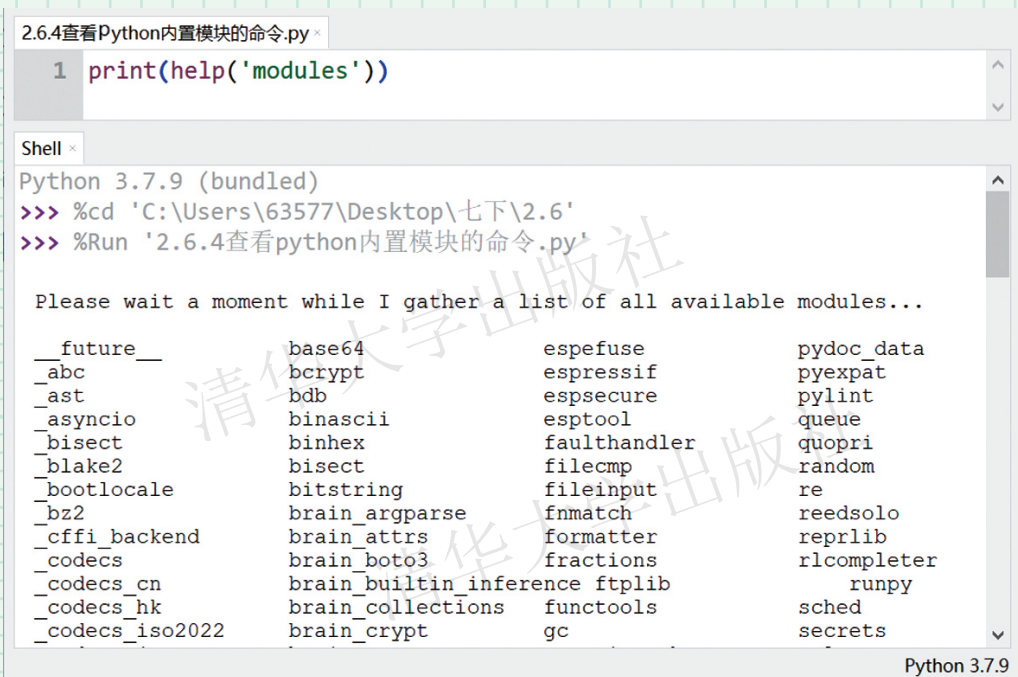
表 2.6.3 Python 的 random 模块及其部分函数

模块	常用函数	实例表示	运行结果
random	randint()	# 随机生成范围内的某个整数（包括头尾） import random print(random.randint(1, 3))	或 1 或 2 或 3
	random()	# 生成 0,1 之间的浮点数 import random print(random.random())	0 ~ 1 之间的任意浮点数
	choice()	# 随机在序列内选择一个元素 import random print(random.choice ('ABC'))	或 A 或 B 或 C

拓展阅读

Python 内置模块的查看

通过 `print(help('modules'))` 命令，可以查看 Python 的内置模块，如图 2.6.6 所示。



The screenshot shows a Python IDE with a file named '2.6.4查看Python内置模块的命令.py'. The code in the editor is:

```
1 print(help('modules'))
```

The output in the Shell window is as follows:

```
Python 3.7.9 (bundled)
>>> %cd 'C:\Users\63577\Desktop\七下\2.6'
>>> %Run '2.6.4查看python内置模块的命令.py'

Please wait a moment while I gather a list of all available modules...

__future__      base64           espfuse          pydoc_data
__abc__          bcrypt           espressif        pyexpat
__ast__          bdb              espsecure        pylint
__asyncio__      binascii         esptool          queue
__bisect__       binhex           faulthandler     quopri
__blake2__       bisect           filecmp          random
__bootlocale__   bitstring        fileinput        re
__bz2__          brain_argparse   fnmatch          reedsolo
__cffi_backend__ brain_attrs       formatter         reprlib
__codecs__       brain_boto3      fractions        rlcompleter
__codecs_cn__    brain_builtin_inference ftplib           runpy
__codecs_hk__    brain_collections functools        sched
__codecs_iso2022 brain_crypt       gc               secrets
```

Python 3.7.9

图 2.6.6 查看 Python 内置模块的命令

在 Python 程序中可以通过 `import` 命令调用模块，其格式如下：

```
import 模块名
```

调用模块中的具体函数，格式如下：

```
模块名.函数名
```



实践活动

猜数字游戏程序的编写实践

猜数字是一个简单有趣的游戏，如图 2.6.7 所示。游戏程序一般要随机生成一个用户认知范围内的合适整数，若用户的猜测数字与正确答案不同则提示大小，直到用户猜出正确数字或取消猜测，最后输出用户的猜测次数。请用 Python 编写一个猜数字的游戏程序。



图 2.6.7 猜数字游戏

实践内容主要包括：

- (1) 用流程图设计相关算法。
- (2) 编写、调试、运行相关代码并观察输出结果。
- (3) 将相关代码与输出结果向同桌进行展示交流。

3. 扩展模块及应用举例

利用模块来进行程序设计，就像搭“积木”。除了丰富内置模块外，还可以使用扩展模块。Python 可以很方便地安装与使用扩展模块。

PyPI (Python 包索引) 是一个丰富的 Python 语言程序包仓库, 在这里往往可以找到很多想要的软件包。比如, “pypinyin” 模块就具有根据汉字获得拼音的功能。在首次使用前需要安装它, 即在终端执行 “pip install pypinyin” 命令。成功安装后, 通过 “import pypinyin” 导入模块后就可以使用它的功能函数了, 如图 2.6.8 所示。

```

1 from pypinyin import pinyin
2 a = input('请输入要注音的汉字:')
3 print(pinyin(a))

```

Shell ×

Python 3.8.3 (D:\XEdu\env\python.exe)

>>> %Run myPinyin.py

请输入要注音的汉字: 贵州省

[['guì'], ['zhōu'], ['shěng']]

图 2.6.8 pypinyin 模块应用举例

Python 的扩展模块很丰富, 根据具体的需要去查找, 但也有一些扩展模块是我们经常会用到的。常见的扩展模块, 如表 2.6.4 所示。

表 2.6.4 常见的扩展模块

模块名称	模块简介
PIL	图像处理工具包 PIL (Python Image Library), 其中提供了基本的图像处理功能, 如: 改变图像大小, 旋转图像, 图像格式转换, 色彩空间转换, 图像增强, 直方图处理, 插值和滤波等功能
Matplotlib	Matplotlib 是一个 Python 2D 绘图库, 只需几行代码即可生成绘图、直方图、功率谱、条形图、错误图、散点图等
Numpy	Numpy 是一个 Python 的扩展库, 代表 “Numeric Python”, 主要用于数学处理和科学计算。它是一个由多维数组对象和用于处理数组的例程集合组成的库
Pandas	Pandas 包是一个快速、强大、灵活和易于使用的开源数据分析和处理的工具
TensorFlow	TensorFlow 是 AlphaGo 的 “最强大脑”。TensorFlow 既是一个实现机器学习算法的接口, 也是一个执行机器学习算法的框架。TensorFlow 还可以用来实现包括线性回归、逻辑回归、随机森林等算法。TensorFlow 建立的大规模深度学习模型的应用场景也非常广泛, 包括语音识别、自然语言处理、计算机视觉、机器人控制、信息抽取、药物研发等, 使用 TensorFlow 开发的模型在这些领域也获得了最前沿的成果

续表

模块名称	模块简介
Keras	Keras 是一个开源神经网络库，它是作为一个接口，提供高层次的抽象，让神经网络的配置变得简单。它能够以 TensorFlow, CNTK, 或者 Theano 作为后端运行。keras 的开发重点是支持快速的实验。能够以最小的时间把想法转换为实验结果，是做好研究的关键
scikit-learn	scikit-learn 是一个非常强大的机器学习 Python 模块，主要用于构建模型，对于统计建模技术非常有效，其特性包括监督学习算法、无监督学习算法和交叉验证
Opencv-python	OpenCV 是一个 C++ 库，用于实时处理计算机视觉，涵盖了很多计算机视觉领域的模块

项目实施

小青团队准备大量使用函数与模块来减少代码的编写，如运用 float 函数确保将输入的字符串转换为浮点数等。

你会使用函数与模块来解决哪些具体问题？请填写在下面的横线上。

第7节 算法的实践

本节知识

- ◆ 明确问题，抽象建模
- ◆ 表达流程，描述算法
- ◆ 设计程序，实现算法

本节活动

- ◆ 升级“简易计算器”

你可能已经认识到，计算机之所以能够通过程序代码来自动、高效地帮助人们解决许多实际问题，其中一个重要原因就是有可靠的算法。但可靠的算法的形成往往不是一蹴而就，而是要经历一个充满智慧的实践过程。算法实践过程，大致可分为“明确问题，抽象建模”“表达流程，描述算法”

与“编写程序，实现算法”这三个环节。就以“简易计算器”项目的实现为例，其完整的算法实践过程如下。

一、明确问题，抽象建模

这个环节又可以根据实际需求，拆分为“明确问题”“问题分解”“抽象”与“建模”等步骤。

明确问题就是进一步确定问题求解的具体目标，也就是所谓的问题需求分析。以“简易计算器”为例，就是列出它的具体功能，可以用简洁的语言进行描述，如“实现两个正整数的加、减、乘、整除与求余运算的功能”。

问题分解就是将复杂、具体的功能需求，分割成多个简单、容易实现的小问题。上述的功能需求，就要分割为“2个正整数相加”“2个正整数相减”“2个正整数相乘”“2个正整数整除”“2个正整数求余”这五个小问题。

抽象是从众多的事物中抽取出共同的、本质性的特征，而舍弃其非本质的特征的过程。上述五个小问题的共同本质特征是：都对两个正整数进行运算，并进行求解。

建模就是在抽象基础之上，建立解决问题的模型。通常是运用数学符号与公式等工具，将复杂的事物关系简化。如果用 a 、 b 表示两个正整数， op 表示运算符， c 表示结果，那么上述五个小问题就可以简化为如图 2.7.1 所示的模型。

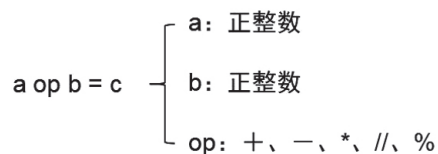


图 2.7.1 “简易计算器”的建模

二、表达流程，描述算法

这个环节包括画流程图和程序流程指引注释。图 2.7.1 所示的模型可以清晰地表明输入、输出与约束条件之间的关系。即 a 与 b 是输入，它们的约束条件是正整数； op 的约束条件是在“+、-、*、//、%”中取值， c 是输出。据此我们可以用流程图的方式，描述它的算法，如图 2.7.2 所示。还可以根据流程图编写算法流程的部分程序注释，来为设计程序做导引，如图 2.7.3 所示。

三、编写程序，实现算法

程序是算法的体现，是对算法的实现和验证。程序设计需要依赖具体的编程语言，编写代码并进行调试，同时不断优化和迭代。

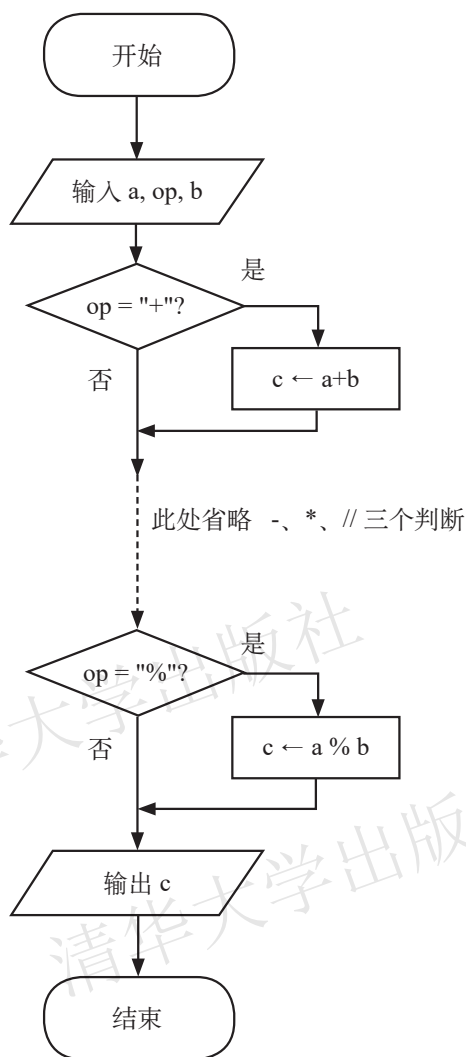


图 2.7.2 简易计算器流程图

```

1  # 输入
2
3  # 分支结构
4  # 加法
5  # 减法
6  # 乘法
7  # 整除
8  # 求余
9
10 # 输出结果
11
    
```

图 2.7.3 “简易计算器”程序部分注释

1. 编写程序

设计程序的过程，也是对流程图“转译”的过程。设计程序的成果是源程序。源程序中除了程序代码外，注释也必不可少，包括算法流程注释、变量说明、代码功能说明等，程序开头部分也应该有源码版权归属、程序主要功能等。程序注释除了用“#”开始的单行注释语句或句尾注释外，在程序头部和函数首部还可以采用文档注释，即用两个半角三引号间包含单行或多行注释文档。

以“简易计算器”为例，程序设计源码如下。

```
'''
名称: 简易计算器
版本: 1.0
功能: 进行两个正整数的加、减、乘、整除和求余的运算。
开发者: 小清、同学甲、同学乙、同学丙开发
'''
# 输入
a = int(input('请输入正整数 a:')) # 第一个数 a
op = input('输入运算符 (+ - * // %):') # 操作符 op
b = int(input('请输入正整数 b:')) # 第二个数 b
# 分支结构
# 加
if op == '+':
    c = a + b
# 减
if op == '-':
    c = a - b
# 乘
if op == '*':
    c = a * b
# 整除
if op == '//':
    c = a // b
# 求余
if op == '%':
    c = a % b
# 输出
print(c)
```

2. 运行调试

程序设计完成后,要进行运行调试。首先,根据信息提示排除各种错误,使程序能正常运行。然后,进行功能测试,使程序达到预期的目标。最后,把程序交付用户试用,接受用户反馈意见,适时进行改进或优化。

以“简易计算器”为例,图 2.7.4 是一次得到正确结果的调试,图 2.7.5 是一次除数为零的出错调试。



```

简易计算器1.0.py ×
17     c = a - b
18 # 乘
19 if op == '*':
20     c = a * b
21 # 整除
22 if op == '//':

Shell ×
>>> %Run '简易计算器1.0.py'
请输入正整数a:5
输入运算符(+ - * // %):*
请输入正整数b:8
40
>>>
    
```

图 2.7.4 简易计算器的一次运行情况



```

Shell ×
Python 3.8.3 (D:\XEdu\env\python.exe)
>>> %Run '简易计算器.py'
请输入正整数a:8
输入运算符(+ - * // %)://
请输入正整数b:0
Traceback (most recent call last):
  File "D:\其他工作\清华版教材\七上202401\简易计算器.py", line 23, in <module>
    c = a // b
ZeroDivisionError: integer division or modulo
by zero
>>>
    
```

图 2.7.5 简易计算器的一次除数为零而出错的调试

3. 迭代升级

程序在调试和使用过程中会发现一些缺陷需要改进，也有一些地方的用户体验需要优化，还有一些功能需要更新，从而不断地进行迭代与升级。



实践活动

升级“简易计算器”

“简易计算器”分享给同学们试用之后，收到了很多问题和建议。经过整理与认真分析，决定对表 2.7.1 中的三个问题和建议进行改进。

表 2.7.1 “简易计算器”程序存在的问题和建议

问题和建议	改进方法
当被除数为零时，整除与求余会报错。该问题需要修正	
误输入运算符“/”等符号，程序报错。该问题需要修正	
程序输出结果过于简单，建议输出完整的算式，如“ $1 * 8 = 8$ ”	

实践内容主要包括：

- (1) 通过小组讨论，确定表 2.7.1 中三个问题的改进方法。
- (2) 根据确定的改进方法，分工修改程序。
- (3) 对分工完成的程序进行整合并运行调试。
- (4) 汇报与交流实践成果，分享升级后的程序。

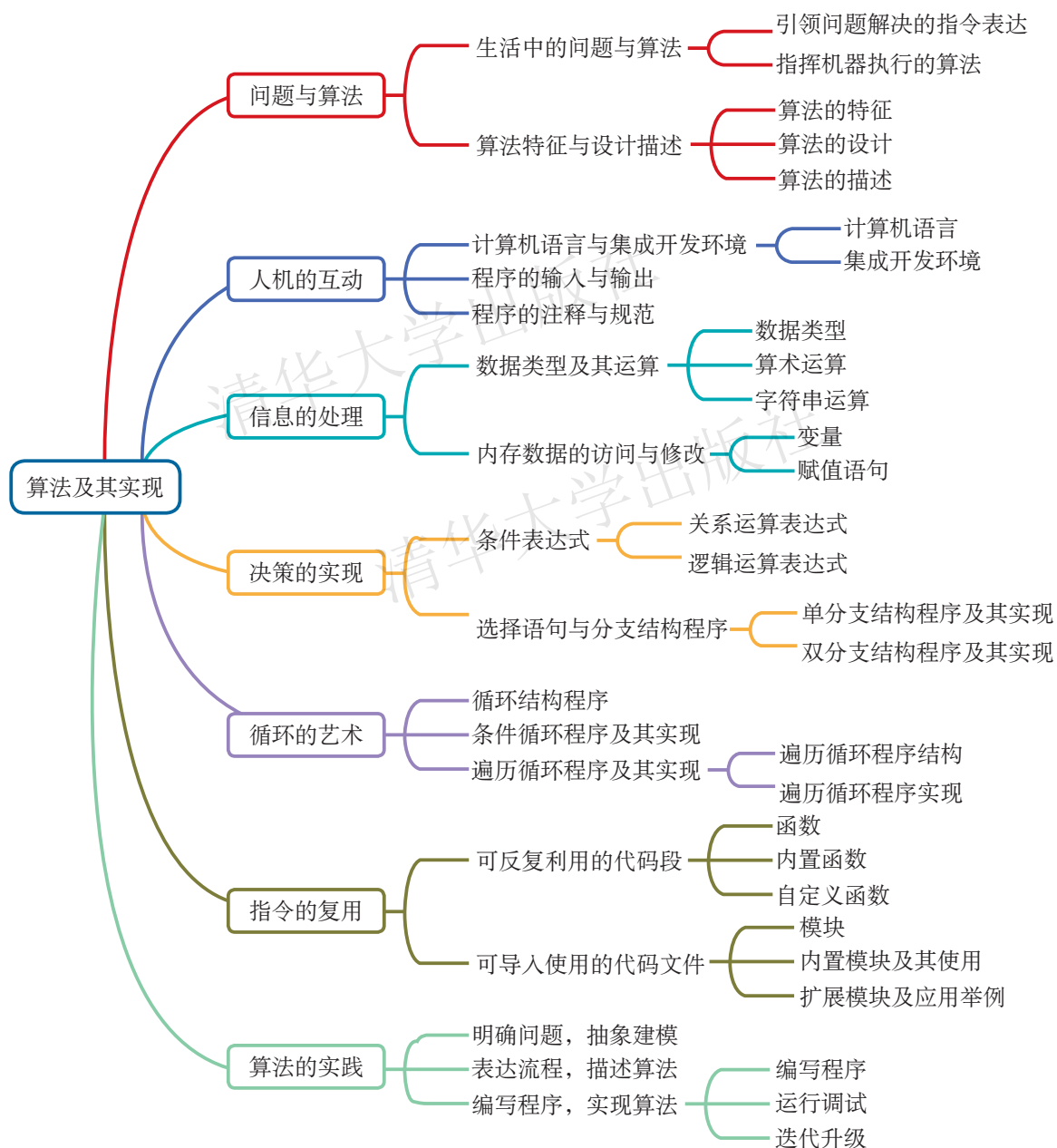
项目实施

本节课的内容，对你小组的项目问题解决有帮助吗？小清团队通过学习“算法实践过程”的相关内容，知道了算法实践的整体过程，并迁移运用到小组的项目中。你还有哪些更好的设想？请填写在下面的横线上。

清华大学出版社
清华大学出版社

单元小结

一、知识回顾



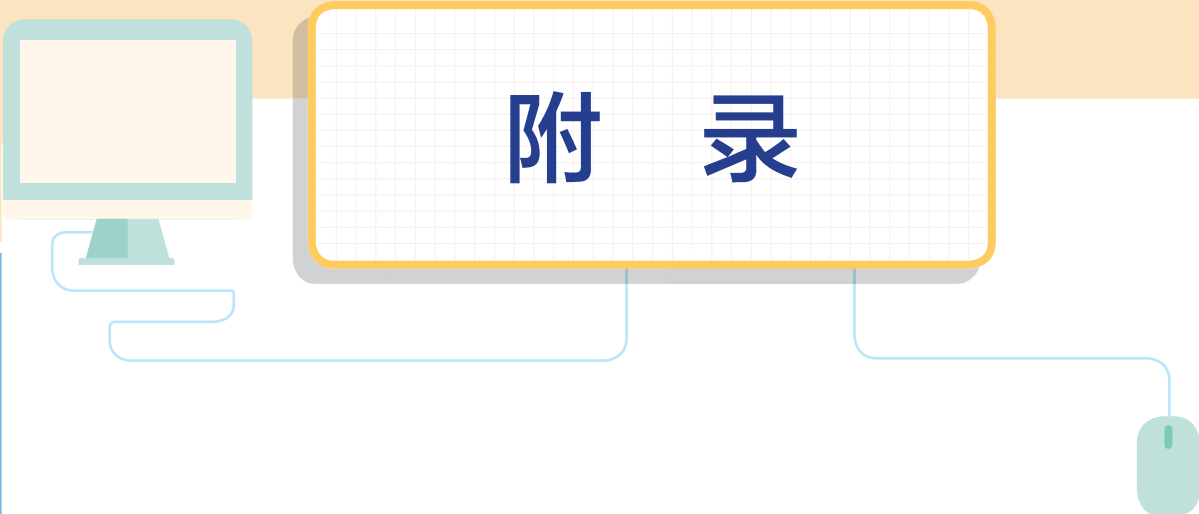
二、成果交流与评价

1. 参考本书附录“项目报告模板”撰写项目报告，并制作演示文稿。
2. 在课堂内展示自己的学习成果并分享经验，在下表中进行自评和他评。

项目成果评价表

评 价 维 度	自 评	他 评
(1) 完整性 项目材料齐全，有需求分析、方案设计、分工协作、项目实施记录表及项目成果等	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般
(2) 实用性 成果内容具体，有真实的项目问题描述、有效的解决方案及相关工具等	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般
(3) 规范性 项目报告规范，符合项目报告的一般格式要求，文字表述准确	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般	<input type="checkbox"/> 优秀 <input type="checkbox"/> 良好 <input type="checkbox"/> 一般

3. 保存相关数据和文档，并上传至校园网或者其他学习空间，与他人分享学习成果。



附录

项目报告模板

(此处填项目主题名称)

组员简介：姓名、组内职务与分工

项目问题：问题背景及需求

方案设计：知识学习、成果规划及成果呈现等

实践过程：实施过程的记录整理

成果汇总：活动成果整理

经验总结：实施活动的心得体会

撰写人：(签名)

日期： 年 月 日