第 5 章　计算思维与算法

计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学领域的一系列思维活动。计算思维不单单是计算机学科所关心的课题，对其他学科也有深远的影响，计算思维的最重要内容就是算法。框图是描述算法的最好方式，对于一个实际问题，用框图描述出来后，再用一种编程语言实现，就能解决实际问题。

本章重点

(1) 了解计算思维。
(2) 熟悉解决问题的一般方法。
(3) 掌握用框图表示解决问题的算法。
(4) 了解 Python 语言程序。
(5) 了解 Raptor 编程。
(6) 了解 C 语言。

5.1　计算思维概述

科学思维是一切科学研究和技术发展的创新灵魂，科学思维包括推理思维、实证思维和计算思维。计算思维可改变大学计算机教育沿袭几十年的教学模式，是大学计算机教育振兴的途径，是计算科学与工程领域创造的革命性的研究成果。

5.1.1　计算机教育

当前，许多人将计算机科学等同于计算机编程；认为计算机只是一个工具，计算机科学专业学生的就业范围很窄；认为计算机科学的基础研究已经完成，剩下的只是工程问题。其实，在计算机领域仍充满挑战，并有许多的科学问题亟待解决。这些问题和解答仅仅受限于人们的好奇心和创造力，一个人可以主修计算机科学并从事任何行业，就像一个人可以主修英语或者数学，接着从事各种各样的职业一样。

计算机教学应当使大学新生接触计算的方法和模型，应当设法激发学生对计算机领域科学探索的兴趣。所以，人们应当传播计算机科学的快乐和力量，致力于使计算思维成为常识，提倡计算思维，宣扬计算思维在教育和科研中的作用，并把这种思维普适化、大众化，使其真正融入人类的一切活动中。

5.1.2 计算思维的定义

计算思维(Computational Thinking)的定义是2006年3月由美国卡内基·梅隆大学计算机科学系系主任周以真教授在美国计算机权威期刊*Communications of the ACM*上给出的。周教授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涵盖计算机科学领域的一系列思维活动。

计算思维集合了数学思维(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)。计算思维建立在计算过程的能力和限制之上,由人设计模型让机器执行。计算方法和模型使人们敢于去处理那些原本无法由个人独立完成的问题求解和系统设计。

具体而言,计算思维是通过约简、嵌入、转化和仿真等方法,把一个困难的问题阐释成如何求解它的思维方法;计算思维可把一个复杂的大而难的问题分解成很多部分,同时去处理,这就是并行处理思想;计算思维是一种递归思维,它把一个难以解决的问题分成两个部分去处理,如不能求解,再把每部分分成两个部分处理之,这就是分而治之的思想;计算思维能把代码译成数据,又能把数据译成代码,是一种多维分析推广的类型检查方法;计算思维是一种采用抽象和分解的方法来控制庞杂的任务或者进行巨型复杂系统的设计,是基于关注点分离的方法;计算思维是一种选择合适的方式陈述一个问题,或对一个问题的相关方面建模使其易于处理的思维方法;计算思维是利用海量数据来加快计算,在时间和空间之间、在处理能力和存储容量之间进行折中的方法等。

5.1.3 计算思维的内容

计算思维的本质是抽象(Abstraction)和自动化(Automation)。计算思维中的抽象完全超越物理的时空观,并完全用符号来表示,其中,数字抽象只是一类特例。计算思维解决的最基本的问题是什么是可计算的,即弄清楚哪些是人类比计算机做得好的,哪些是计算机比人类做得好的。计算思维是每个人的基本技能,而不是只有计算机科学家才需要具备的。在培养孩子的能力时除了要掌握阅读、写作和算术(3R,Reading Writing and Arithmetic)能力,还要学会计算思维。正如印刷出版促进了3R的普及,计算和计算机也以类似的正反馈促进了计算思维的传播。计算思维有助于培养学生的理性思维,以及用模型化、程序化的思想解决实际问题的能力,从而提高学生的思维能力。

5.1.4 计算思维的特点

计算思维有如下特点。

(1) 计算思维是概念化而不是程序化。像计算机科学家那样去思维意味着不仅能为计算机编程,而且还能够在抽象的多个层次上思维。

(2) 计算思维是根本的而不是刻板的技能。根本技能是每一个人为了在现代社会中

发挥职能所必须掌握的。刻板技能意味着机械重复。具有讽刺意味的是，当计算机像人类一样思考之后，思维可就真的变成机械的了。

(3) 计算思维是人的而不是计算机的思维方式。计算思维是人类求解问题的一条途径，但绝非要使人类像计算机那样思考。计算机枯燥且沉闷，人类聪颖且富有想象力，是人类赋予计算机激情，配置了计算设备，用自己的智慧去解决那些在计算时代之前不敢尝试的问题，实现“只有想不到，没有做不到”的境界。

(4) 计算思维是数学和工程思维的互补与融合。计算机科学在本质上源自数学思维，因为像所有的科学一样，其形式化基础建于数学之上。计算机科学又从本质上源自工程思维，因为人们建造的是能够与实际世界互动的系统，基本计算设备的限制迫使计算机科学家必须计算性地思考，不能只是数学性地思考。构建虚拟世界的自由使人们能够设计超越物理世界的各种系统。

(5) 计算思维是思想而不是人造物。不只是人们生产的软件、硬件等人造物以物理形式到处呈现并时时刻刻触及人们的生活，更重要的是人们拥有求解问题、管理日常生活、与他人交流和互动的计算概念。当计算思维真正融入人类活动以致不再表现为一种显式哲学的时候，它就将成为一种现实。

5.1.5 计算思维对非计算机学科的影响

计算思维不只是计算机学科所关心的课题，而且对其他学科也有着深远影响。例如，计算生物学正在改变着生物学家的思考方式；纳米计算正在改变着化学家的思考方式；量子计算正在改变着物理学家的思考方式；博弈计算理论正在改变着经济学家的思考方式等。随着计算机科学的普及，计算机科学专业的术语都已经口语化了，如“非确定随机算法”“垃圾收集”这样的术语已经司空见惯。这说明计算机科学的知识、计算机科学的发展、计算思维已经不知不觉地深入到其他学科，而且人们都在使用，实际上已经被人们所接受。

5.2 解决问题的一般方法

计算思维的核心是用计算机解决实际问题。需要用计算机解决的具体问题分为两类：一类是数值计算问题；另一类是非数值计算问题。对于数值计算问题，通常要从具体问题抽象出一个适当的数学模型，然后设计一个解此数学模型的算法，最后编出程序、进行测试和修改直至得到结果。例如，预报城市交通流问题的数学模型是线性方程组。对于非数值计算问题，通常要用到复杂的数据结构，有时也会用到数学模型。不论是哪一类问题，都需要用到分支、循环等知识。

用计算机解决问题的一般方法如下。

(1) 用框图或自然语言描绘出解决问题的步骤，本书用框图描绘。描绘出的解决问题的步骤也称为算法。

(2) 用程序设计语言来实现解决问题的步骤，即用程序设计语言把框图表示的算法翻译成机器能够理解的并可以执行的程序。

由于计算机不能直接执行用高级程序设计语言编写的程序，这里的执行是指由翻译或编译程序进行解释执行或编译成机器代码再执行。常用的语言包括 Python 语言、Raptor、C 语言等，对应的常用的编程环境有 IDLE、Raptor、VC++ 等。

5.3 用框图表示解决问题的算法

算法表示和设计是培养和训练计算思维的重要途径。表示算法和设计算法的方法很多，框图是一种好方法。框图又称为流程图，是一种直观表达算法和设计算法的工具。学习算法应该首先学会使用框图。由于框图直观且易于修改，有利于人们表达出解决问题的思想和方法。对于十分复杂难解的问题，框图可以画得粗放且抽象些，首先表达出解决问题的轮廓，然后细化；对于较为简单的问题，框图可以画得粗放也可以画得细致。

常用的框图符号如图 5-1 所示。

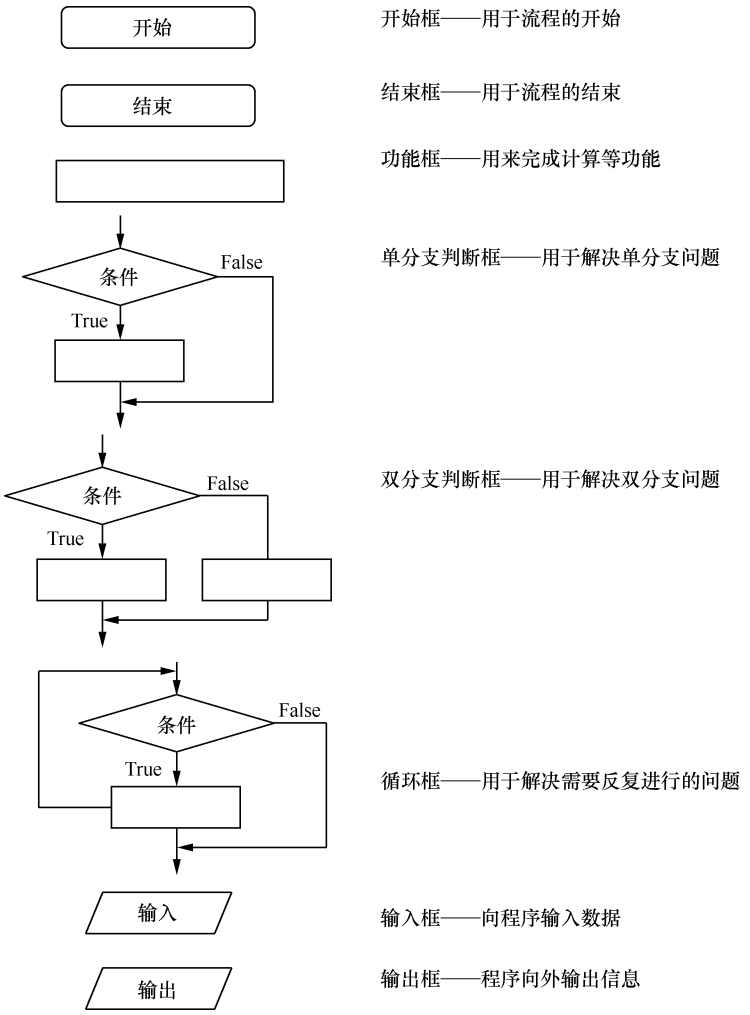

图 5-1 常用的框图符号

【问题 5-1】 计算长方形面积。

分析：长方形面积公式为 $s=a\times b$，只要设置了长 a，宽 b，就能计算出面积 s。

用框图表示的长方形面积算法如图 5-2 所示。

【问题 5-2】 用户输入一个三位自然数，让计算机输出百位数、十位数和个位数。

分析：该问题需要把三位数的百位、十位、个位分离出来。三位数除以 100，其整数部分就是百位数，后两位数除以十，其整数部分就是十位数，因而可画出图 5-3 所示的框图。

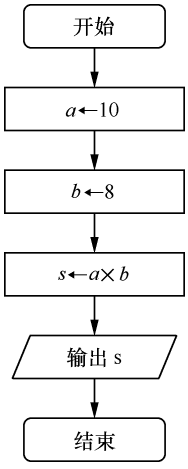

图 5-2 长方形面积算法

图 5-3 求三位数各位数字的算法

【问题 5-3】 由键盘输入一个整数，如果是偶数则输出“偶数”，如果是奇数则输出“奇数”。

分析：一个整数能被 2 整除就是偶数，否则就是奇数。

用框图表示的判断整数奇偶性的算法如图 5-4 所示。

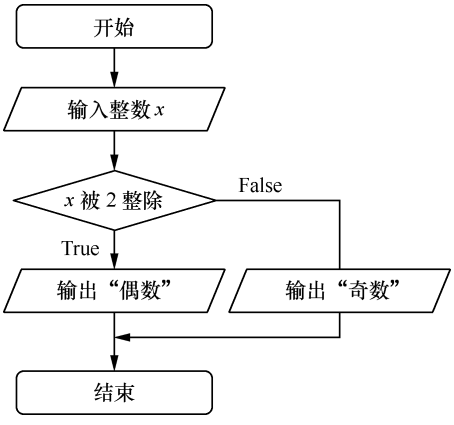

图 5-4 判断整数的奇偶性

【问题 5-4】 编程计算 1＋2＋3＋…＋n 的值，n 由键盘输入。

分析：每次循环累加一个整数值，整数的取值范围为 1～n，需要使用循环。

用框图表示的算法如图 5-5 所示。

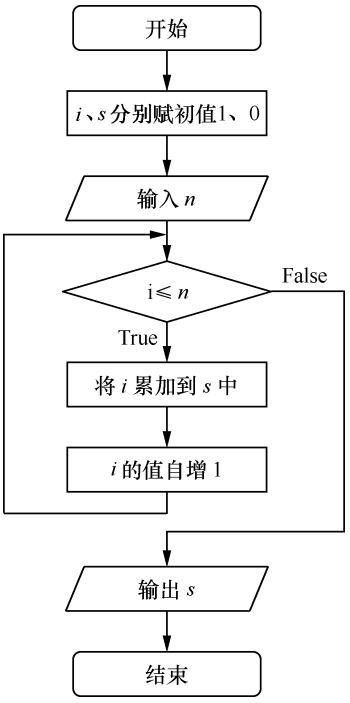

图 5-5 从 1 累加到 n 的算法

5.4 Python 语言

5.4.1 Python 语言简介

Python 是一种面向对象、直译式计算机程序设计语言，也是一种功能强大的通用型语言，由 Guido van Rossum 于 1989 年末开发，经过近 30 年的发展，已经成熟且稳定。它包含一组完善且容易理解的标准库，能够轻松完成很多常见的任务。它的语法非常简捷和清晰，与其他大多数计算机程序设计语言不一样，它采用缩进来定义语句块。

Python 支持命令式编程、面向对象程序设计、函数式编程、泛型编程等多种编程方式。与 Scheme、Ruby、Perl、TCL 等动态语言一样，Python 具备垃圾回收功能，能够自动管理内存。它经常被当作脚本语言，用于处理系统管理任务和 Web 编程，并且非常适合完成各种高阶任务。Python 虚拟机本身几乎可以在所有的操作系统中运行。使用诸如 py2exe、PyPy、PyInstaller 之类的工具可以将 Python 源代码转换成可以脱离 Python 解释器执行的程序。

Python 2.0 于 2000 年 10 月 16 日发布，主要是实现了完整的垃圾回收，并且支持 Unicode。同时，整个开发过程更加透明，Python 社区对开发进程的影响逐渐扩大。Python 3.0 于 2008 年 12 月 3 日发布，此版不完全兼容之前的 Python 代码。不过，很多新特性后来也被移植到旧的 Python 2.6/2.7 版本。之后，还推出了 Python 3.4、Python 3.5、Python 3.6、Python 3.7 等版本。

Python 是完全面向对象的语言，函数、模块、数字、字符串都是对象，并且完全支持继承、重载、派生、多继承，有益于增强代码的复用性。Python 支持重载运算符，因此 Python 也支持泛型设计。虽然 Python 可能被粗略地分类为“脚本语言”(Script Language)，但实际上一些大规模软件开发计划(如 Zope、Mnet、BitTorrent、Google)也广泛地使用它。

Python 本身被设计为可扩展的，并非所有的特性和功能都集成到语言核心。Python 提供了丰富的 API 和工具，以便程序员能够轻松地使用 C、C++ 、Python 语言来编写扩展模块。Python 解释器本身也可以被集成到其他需要脚本语言的程序内，因此，很多人还把 Python 作为一种“胶水语言”(Glue Language)使用，使用 Python 将其他语言编写的程序进行集成和封装。Google 的很多项目都使用 C++ 编写性能要求极高的部分，然后用 Python 调用相应的模块。很多游戏(如 EVE Online)使用 Python 来处理游戏中繁多的逻辑。

在很多操作系统中，Python 是标准的系统组件。大多数 Linux 发行版以及 NetBSD、OpenBSD 和 Mac OS X 都集成了 Python，可以在终端直接运行 Python。有一些 Linux 发行版的安装器使用 Python 语言编写，比如 Ubuntu 的 Ubiquity 安装器，Red Hat Linux 和 Fedora 的 Anaconda 安装器。Gentoo Linux 使用 Python 来编写它的 Portage 包管理系统。Python 标准库包含了多个调用操作系统功能的库。通过 pywin32 这个第三方软件包，Python 能够访问 Windows 的 COM 服务及其他 Windows API。使用 IronPython，Python 程序能够直接调用.Net Framework。一般说来，Python 编写的系统管理脚本在可读性、性能、代码重用度、扩展性几个方面都优于普通的 Shell 脚本。

NumPy、SciPy、Matplotlib 可以让 Python 程序员编写科学计算程序。PyQt、PySide、wxPython、PyGTK 是 Python 快速开发桌面应用程序的利器。

TensorFlow 是谷歌为支持其研究和生产目标创建的项目，于 2015 年发布，它是一款开源机器学习框架，易于在各种平台上使用和部署。它是机器学习中维护得最好和使用广泛的框架之一，对 Python 有良好的支持。

MXNet 是亚马逊(Amazon)选择的深度学习库。它拥有类似于 Theano 和 TensorFlow 的数据流图，有助于实现多 GPU 配置，有类似于 Lasagne 和 Blocks 更高级别的模型构建块，并且可以在任何硬件上运行(包括手机)，对 Python 有良好的支持。

Caffe(Convolutional Architecture for Fast Feature Embedding，用于快速特征嵌入的卷积结构)最初于 2017 年发布，Caffe 是一种专注于表现力、速度和模块性的机器学习框架。该框架采用 C++ 编写，并附带一个 Python 界面。

Keras 是一个开源机器学习库，最初于 2015 年发布，旨在简化深度学习模型的创建。

它用 Python 编写，可以部署在其他人工智能技术之上，如 TensorFlow、Microsoft Cognitive Toolkit(CNTK)和 Theano。

Python 对于各种网络协议的支持都很完善，因此经常被用于编写服务器软件、网络爬虫。第三方库 Twisted 支持异步网络编程和多数标准的网络协议(包含客户端和服务器)，并且提供了多种工具，被广泛用于编写高性能的服务器软件。

适用于 Python 的集成开发环境(Integrated Development Environment，IDE)软件，除了标准二进制发布包所附的 IDLE 之外，还有许多其他选择。其中有些软件设计有语法着色、语法检查、运行调试、自动补全、智能感知等便利功能。由于 Python 的跨平台出身，这些软件往往也具备各种操作系统的版本或一定的移植性。

下面介绍几款专门为 Python 设计的 IDE 软件。

(1) Eric：基于 PyQt 的自由软件，功能强大。支持自动补全、智能感知、自动语法检查、工程管理、SVN/CVS 集成、自动单元测试等功能。调试功能与 Visual Studio 和 Eclipse 类似。目前同时有两个版本：Eric4 支持 Python 2.x，Eric5 支持 Python 3.x。使用前需要先安装相应的 PyQt 版本。

(2) Ulipad：功能较全的免费软件，依赖 wxPython。

(3) IDLE：Python“标准”IDE。一般随 Python 安装，支持较少的编辑功能，调试功能也比较弱，但容易使用。

(4) Komodo 和 Komodo Edit：后者是前者的免费精简版。

(5) PythonWin：包含在 pywin32 内的编辑器，仅适用于 Windows。

(6) SPE(Stani's Python Editor)：功能较多的免费软件，依赖 wxPython。

(7) WingIDE：可能是功能最全的 IDE，但不是免费软件。

(8) Uliweb、Django、Web2py：用于开发网站的 Python 框架。

【例 5-1】 问题 5-4 的 Python 语言程序。

```
#Exp5_1.py
i, s=1, 0                                  #定义两个变量 i 和 s
n=eval(input('n=') )                       #从键盘输入一个整数送给 n
while i<=n:                                #循环语句
      s=s+i                                #对 i 进行累加
      i=i+1
print('1+2+3+…+', n, '=', s)               #输出数据
```

5.4.2 Python 语言基础

1. 赋值语句

赋值语句是 Python 中最基本的语句，用来产生变量。使用赋值语句可定义变量并为其赋初值。

赋值语句的语法如下。

变量=表达式

例如：

```
x=3
y=5
z=x+y
```

2. 模块导入

Python 编程效率高的主要原因是有大量的开源模块，开发者只要导入所需要的模块就能进行编程。导入模块之后，就能使用模块中的函数和类来解决开发中的实际问题。导入模块使用 import 语句，有以下 3 种用法。

1）导入整个模块

一般格式如下。

```
import 模块名 1[, 模块名 2 [, …] ]
```

模块名就是程序文件的前缀，不含.py，可一次导入多个模块。导入模块之后，调用模块中的函数或类时，需要以模块名为前缀，程序的可读性好，具体如下。

```
>>> import math
>>> math.sin(0.5)
0.479425538604203
```

2）与 from 联用导入整个模块

一般格式如下。

```
from 模块名 import *
```

使用这种方式导入模块之后，调用模块中的函数或类时，仅使用函数名或类名，程序简洁但可读性稍差。例如：

```
>>> from math import *
>>> cos(0.5)
0.8775825618903728
```

3）与 from 联用导入一个或多个对象

一般格式为如下。

```
from 模块名 import  对象名 1[, 对象名 2 [, …] ]
```

这种方式只导入模块中的一个或多个对象，调用模块中的对象时，仅使用对象名。例如：

```
>>> from math import exp , sin, cos
>>> exp(1)
2.718281828459045
>>> sin(0.5)
```

```
0.479425538604203
>>> cos(0.5)
0.8775825618903728
```

Python 中的常用模块见表 5-1。

表 5-1　Python 中的常用模块

模　块	说　　明
os	os 模块包装了不同操作系统的通用接口，使用户在不同操作系统下，可以使用相同的函数接口调用操作系统的功能
sys	sys 是系统信息和方法模块，提供了很多实用的变量和方法
math	math 模块定义了标准的数学方法，如 cos(x)、sin(x)等
random	random 模块提供了多种方法来产生随机数
struct	struct 模块可把数字和 bool 值与字节串进行相互转换
pickle	pickle 模块可把对象变为字节串并写入文件中，也可从文件中读出对象
datetime	datetime 模块中有日期时间处理的方法
time	time 模块中有与时间、时钟、计时相关的方法
tkinter	tkinter 模块提供了图形界面开发的方法
mySQLdb	mySQLdb 模块中有操纵 mySQL 数据库的方法，该模块需要下载
urllib	urllib 包提供了高级的接口来实现与 HTTP Server、FTP Server 和本地文件交互的 Client，该模块需要下载

random 的简单用法如下。

```
random.randint(a, b)              #返回 a~b 的随机整数
random.random()                   #返回 0~1 的随机小数
```

例如：

```
>>> import random
>>> random.randint(0,100)
5
>>> random.randint(0,100)
60
>>> random.randint(0,100)
65
>>> random.randint(0,100)
38
>>> random.random()
0.04113355816162545
>>> random.random()
0.8554658837478725
>>> random.random()
0.3029787171878404
```