

普通高等教育“十三五”规划教材

Python程序设计基础

虞 歌◎主编

PYTHON



本书配有iStudy通用实践评价平台使用，
可实现在线学习、练习、测评与考务管理。

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十三五”规划教材

Python 程序设计基础

虞 歌 主编



中国铁道出版社

CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是以 Python 语言 (Python 3 版本) 作为入门语言的程序设计教材, 以崭新的思路进行设计和编排。全书以程序设计零起点读者为主要对象, 以培养程序设计能力为目标, 简洁通俗, 循序渐进, 通过例题, 重点讲解程序设计思想和方法, 力图将 Python 语言基础知识介绍和程序设计能力培养完美结合, 培养读者对程序设计的兴趣, 提高读者程序设计能力水平。

本书共 10 章, 包括程序设计概述, 基本程序设计, 程序的控制结构, 函数, 字符串、列表和元组, 字典和集合, 对象和类, 继承和多态, 异常处理以及文件。

本书适合作为高等学校学生学习程序设计课程的教材, 也可供程序员和编程爱好者参考使用。

图书在版编目 (CIP) 数据

Python 程序设计基础/虞歌主编. —北京: 中国铁道出版社, 2018. 6

普通高等教育“十三五”规划教材

ISBN 978-7-113-24496-5

I. ①P… II. ①虞… III. ①软件工具-程序设计-高等学校-教材 IV. ①TP311. 56

中国版本图书馆 CIP 数据核字 (2018) 第 102639 号

书 名: Python 程序设计基础

作 者: 虞 歌 主编

策 划: 祝和谊

读者热线: (010) 63550836

责任编辑: 周 欣 徐盼欣

封面设计: 崔丽芳

责任校对: 张玉华

责任印制: 郭向伟

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 河北省三河市燕山印刷有限公司

版 次: 2018 年 6 月第 1 版 2018 年 6 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 16.5 字数: 395 千

书 号: ISBN 978-7-113-24496-5

定 价: 45.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 51873659

前言

PREFACE

随着信息产业的迅速发展,软件人才的需求量越来越大。程序设计是软件人才必备的基础知识和技能。

程序设计基础是一门理论与实践密切相关、以培养学生程序设计能力为目标的课程。如何消除学生学习程序设计的畏难情绪,使学生顺利进入程序设计的大门,逐步掌握程序设计思想和方法,提高实践动手能力,是本课程教学的难题。

程序设计既是科学,也是艺术。学习程序设计是一件非常辛苦的事情,要有非常强的耐心和实践精神,需要花费大量的时间,不可能一蹴而就,必须从某个起点开始循序渐进。

本书就是一个很好的起点。本书以程序设计零起点读者为主要对象,采用 Python 语言(Python 3 版本)作为程序设计的描述语言。Python 语言是目前业界广泛使用的程序设计语言,编者确信选用 Python 语言作为程序设计基础课程的教学语言是正确的选择。在教学实践中,编者感到 Python 语言的简洁、灵活和高效,能够带给软件开发者无尽想象的空间,同时也深深感到讲授 Python 语言过程中面临的困难和挑战,意识到在程序设计基础课程中讲授 Python 语言并不是那么容易的。

尽管目前有关学习 Python 语言的书籍很多,但学习 Python 语言仍然让大多数初学者心存畏惧。编者一直从事程序设计方面的教学和科研工作,主讲过多门程序设计课程,积累了丰富的教学经验。在本书编写过程中,编者结合自己教学和使用 Python 语言的经验和感悟,以程序设计为主线,通过例题,简洁通俗讲解程序设计思想和方法,并穿插介绍相关的语言知识,循序渐进地培养学生程序设计能力。本书对那些渴望掌握 Python 语言而又心存畏惧的初学者而言是一个很好的选择。

教学改革的重点之一,就是要抓学生实践动手能力的培养,学生的能力是实现就业的决定因素,而就业率又是体现教育质量的重要指标。杭州师范大学作为国内首家服务外包本科学院以及教育部、商务部在江苏、浙江两省开展地方高校计算机学院培养服务外包人才试点工作单位,我们实施了程序设计课程的教学改革,在教学内容、教学方法、教学手段和考核方式上,基本形成了比较完整的体系,目的就是培养学生的程序设计能力,适应社会对软件服务外包人才培养的需求。本书配有 iStudy 通用实践评价平台,可实现在线学习、练习、测评与考务管理。本书源于教学改革和教学实践,体现了程序设计教学改革的成果。

读者要获取本书的相关资源,请访问中国铁道出版社网站 <http://www.tdpress.com/51eds/>。

本书由杭州师范大学虞歌主编。在本书编写过程中,参考了部分图书资料和网站资料,在此向其作者表示感谢。本书的出版得到了中国铁道出版社的大力支持,在此表示衷心的感谢。

感谢读者选择本书。由于编者水平和经验有限,书中难免有不足之处,恳请读者提出宝贵意见和建议,使本书日臻完善。编者联系方式: yuge@hznu.edu.cn。

编者

2018年4月

与本书配套的数字课程资源使用说明

一、云端在线课堂+教材融合

教材使用者请访问教学平台(<http://dodo.hznu.edu.cn>), 开始课程学习。账号请联系作者或编辑。

二、资源使用

与本书配套的数字课程资源按照章、节知识树的形式组织, 配有电子教案、案例素材、微视频等资源。

1. 电子教案: 教师上课使用的与课程和教材紧密配套的教学 PPT, 可供教师下载使用, 也可供学生学习使用。

2. 案例素材: 每章节的例题源文件、思考与练习答案以及编程题解答源文件, 可供教师下载使用, 也可供学生学习使用。

3. 微视频: 内容涵盖各知识点的讲述和各案例的实际操作讲解, 能够让学生随时随地观看比较直观的视频讲解。

目录

CONTENTS

第 1 章 程序设计概述	1	2.3.2 常量	21
1.1 计算机系统	1	2.4 数值数据类型和运算符	21
1.1.1 计算机硬件系统	1	2.5 字符串和字符简介	25
1.1.2 计算机软件系统	2	2.6 列表、元组和字典简介	29
1.1.3 计算机中的信息表示	3	2.6.1 初识列表	29
1.2 程序设计基础	5	2.6.2 初识元组	30
1.2.1 程序	5	2.6.3 初识字典	31
1.2.2 程序设计语言	5	2.7 内置函数和数学函数	32
1.2.3 程序设计	6	2.7.1 内置函数	32
1.3 Python 语言的发展历史与特点	7	2.7.2 数学函数	33
1.3.1 Python 语言的发展历史	7	2.8 对象和方法简介	36
1.3.2 Python 语言的特点	8	2.9 程序设计风格	38
1.4 Python 程序开发	9	2.9.1 适当的注释	38
1.4.1 基本术语	9	2.9.2 命名习惯	38
1.4.2 Python 程序的开发过程	10	2.9.3 程序编排	39
1.4.3 搭建 Python 编程环境	10	思考与练习	39
1.5 初识 Python 程序	13	编程题	41
1.5.1 第一个 Python 程序	13	第 3 章 程序的控制结构	42
1.5.2 两个整数的加、减、乘、 除、整除程序	14	3.1 单入口单出口的控制结构	42
1.5.3 算术表达式求值程序	15	3.2 布尔类型和关系运算符	43
1.5.4 华氏温度转换为摄氏 温度的程序	15	3.3 成员运算符和身份运算符	45
思考与练习	16	3.4 if 语句	47
编程题	16	3.4.1 单分支 if 语句和双分支 if-else 语句	47
第 2 章 基本程序设计	18	3.4.2 多分支 if-elif-else 语句	48
2.1 计算三角形面积的程序	18	3.5 产生随机数	50
2.2 标识符及其命名规则	19	3.6 逻辑运算符	51
2.2.1 标识符	19	3.7 循环	53
2.2.2 关键字	19	3.8 while 语句	54
2.3 变量和常量	20	3.9 for 语句	57
2.3.1 变量	20	3.10 pass、break 和 continue 语句	60
		3.11 嵌套循环	61

思考与练习	63	5.2.4 列表和函数	133
编程题	65	5.2.5 二维列表	135
第 4 章 函数	68	5.3 元组	137
4.1 函数的定义和调用	68	5.3.1 元组的概念	137
4.2 函数的设计规则	72	5.3.2 元组的基本操作	138
4.2.1 函数头的设计规则	72	5.4 排序和查找	140
4.2.2 函数体的设计规则	72	5.4.1 排序	140
4.3 函数的参数	73	5.4.2 查找	142
4.3.1 位置参数和关键字参数	73	思考与练习	144
4.3.2 默认参数	74	编程题	146
4.3.3 参数传递	75	第 6 章 字典和集合	150
4.3.4 包裹传递参数和解包裹	76	6.1 字典	150
4.4 变量的作用域	78	6.1.1 字典的概念	150
4.5 lambda 表达式	79	6.1.2 字典的基本操作	151
4.6 函数式编程	81	6.2 集合	157
4.7 递归	84	6.2.1 集合的概念	157
4.7.1 递归的定义	84	6.2.2 集合的基本操作	158
4.7.2 运用递归求解问题	86	思考与练习	165
4.7.3 递归和循环	87	编程题	166
4.7.4 尾递归	89	第 7 章 对象和类	168
4.8 海龟图	90	7.1 面向对象程序设计	168
4.8.1 设置画布	90	7.2 使用类编写程序	169
4.8.2 控制海龟	92	7.2.1 声明类	169
4.8.3 控制画笔	95	7.2.2 创建对象	170
4.8.4 分形图形	100	7.3 对象作为函数参数	171
4.9 模块化代码	101	7.4 对象列表	171
思考与练习	102	7.5 隐藏数据域	172
编程题	104	7.6 类变量和类方法	174
第 5 章 字符串、列表和元组	107	7.7 静态方法	176
5.1 字符串	107	7.8 使用属性	177
5.1.1 字符编码和字符串	107	7.9 运算符重载和特殊方法	179
5.1.2 字符串的基本操作	109	7.10 定制类	182
5.1.3 正则表达式	115	7.11 迭代器	185
5.2 列表	121	7.12 生成器	188
5.2.1 列表的概念	121	7.13 处理日期和时间	190
5.2.2 列表的基本操作	122	7.13.1 datetime 类	190
5.2.3 复制列表	131	7.13.2 timedelta 类	194

思考与练习	198	9.7 自定义异常类	231
编程题	200	9.8 断言	232
第 8 章 继承和多态	203	思考与练习	233
8.1 继承的概念	203	编程题	233
8.2 单继承	204	第 10 章 文件	236
8.3 覆盖方法	206	10.1 文件的概念	236
8.4 object 类	207	10.2 文件操作	237
8.5 多态和动态绑定	207	10.3 文件读写	238
8.6 鸭子类型	209	10.3.1 文本文件读写	238
8.7 与对象和类相关的内置函数	210	10.3.2 二进制文件读写	241
8.8 类之间的关系	212	10.4 对象序列化	242
8.9 collections 模块	215	10.4.1 pickle 模块	242
8.10 多继承	218	10.4.2 JSON	243
思考与练习	220	10.5 内存文件	246
编程题	221	10.5.1 StringIO	246
第 9 章 异常处理	224	10.5.2 BytesIO	247
9.1 程序设计错误	224	10.6 CSV 文件	247
9.2 什么是异常	225	10.7 电子表格文件	250
9.3 如何处理异常	225	思考与练习	253
9.4 内置异常类	226	编程题	253
9.5 抛出异常	228	参考文献	255
9.6 进一步讨论异常处理	229		

在学习和使用计算机时，一开始就必须建立正确的计算机系统观点。计算机的组成不仅与硬件有关，而且涉及软件。我们撰写文档、畅游因特网时，使用的文字处理器、浏览器都是在计算机上运行的软件。而软件是使用程序设计语言开发出来的。Python 就是一种流行且功能强大的程序设计语言。本书将带领读者学习使用 Python 语言。

1.1 计算机系统

1.1.1 计算机硬件系统

1. 计算机硬件结构

计算机硬件是计算机系统中所有物理装置的总称。计算机硬件系统有 5 个基本组成部分，即控制器、运算器、存储器、输入设备和输出设备。控制器和运算器构成了计算机硬件系统的核心——中央处理器（central processing unit, CPU）。存储器可分为内存储器 and 外存储器，简称内存和外存。计算机硬件系统各个基本组成部分之间是用总线相连接的。总线是计算机硬件系统内部传输各种信息的通道。

图 1.1 给出了一般的计算机硬件结构图。

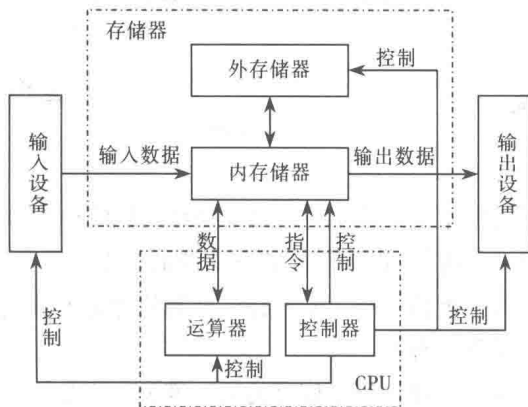


图 1.1 计算机硬件结构图

2. 中央处理器

中央处理器由控制器和运算器组成，是计算机硬件系统的核心。

(1) 计算机硬件系统的各个基本组成部分能够有条不紊地协调工作，都是在控制器的控制下完成的。程序是完成既定任务的一组指令序列，每条指令都规定了计算机所要执行的一种基本操作。控制器按照程序规定的流程依次执行指令，最终完成程序所要实现的任务。

(2) 运算器在控制器的控制下，进行算术运算和逻辑运算。运算器内部有算术逻辑运算单元 (arithmetical logical unit, ALU) 及存放运算数据和运算结果的寄存器。

3. 存储器

存储器由内存和外存组成，其主要功能是存放程序和数据。

(1) 内存由大规模或超大规模集成电路芯片构成。内存分为随机存取存储器 (random access memory, RAM) 和只读存储器 (read-only memory, ROM) 两种。RAM 用于存放正在运行的程序和数据，一旦关闭计算机 (断电)，RAM 中的信息就会丢失。ROM 中的信息一般只能读出而不能写入，断电后，ROM 中的原有信息保持不变，在计算机重新开机后，ROM 中的信息仍可被读出。内存由许多存储单元构成，所有的存储单元都按顺序编号，这些编号称为内存地址。内存中所有存储单元的总和称为内存的存储容量。

(2) 外存用于存放大量的需要长期保存的程序和数据。计算机若要运行存储在外存中的某个程序，必须将它从外存读入内存中。外存按存储介质材料可以分为磁存储器、光存储器和闪存 (flash memory) 存储器。磁存储器中最常用的是硬盘 (hard disk)。光存储器中最常用的是 CD (compact disk) 和 DVD (digital versatile disk)。闪存存储器包括固态硬盘 (solid state disk, SSD)、闪存盘 (又称优盘、U 盘等)。

4. 输入设备和输出设备

(1) 输入设备用于向计算机输入程序和数据。最常用的输入设备是键盘 (keyboard) 和鼠标 (mouse)。

(2) 输出设备用于输出计算机的处理结果。最常用的输出设备是显示器 (monitor) 和打印机 (printer)。

通常把控制器、运算器和内存称为计算机的主机，把输入设备、输出设备和外存称为计算机的外围设备。

1.1.2 计算机软件系统

1. 计算机软件的概念

计算机软件是计算机系统的重要组成部分，如果把计算机硬件看作计算机的“躯体”，那么计算机软件就是计算机系统的“灵魂”。没有任何软件支持的计算机称为“裸机”，只是一些物理设备的堆积，几乎不能工作。只有配备了一定的软件，计算机才能发挥其作用。而计算机功能的强弱也与其所配备的软件的丰富程度有关。

计算机软件是计算机系统中与硬件相互依存的另一部分，它是程序、数据及其相关文档的完整集合。

在程序正常运行过程中,需要输入一些必要的的数据。文档是与程序开发、维护和使用有关的图文材料。程序和数据必须装入计算机内部才能工作。文档一般是给人看的,不一定装入计算机。

2. 计算机软件的分类

计算机软件一般可以分为系统软件和应用软件两大类。

(1) 系统软件居于计算机系统中最靠近硬件的一层,其他软件都是通过系统软件发挥作用的。系统软件与具体的应用领域无关。

(2) 应用软件是指为解决某一领域的具体问题而开发的软件产品。随着计算机应用领域的不断拓展和广泛普及,应用软件的作用越来越大。

3. 操作系统

最重要的系统软件是操作系统(operating system, OS)。操作系统能对计算机系统硬件和软件资源进行有效的管理和控制,合理地组织计算机的工作流程,为用户提供一个使用计算机的工作环境,起到用户和计算机之间的接口作用。

只有在操作系统的支持下,计算机系统才能正常运行,如果操作系统遭到破坏,计算机系统就无法正常工作。

操作系统有如下主要功能。

(1) 任务管理。对中央处理器的资源进行分配,并对其运行进行有效的控制和管理。

(2) 存储管理。有效管理计算机系统存储中的存储器,为程序运行提供良好的环境,按照一定的策略将存储器分配给用户使用,并及时回收用户不使用的存储器,提高存储器的利用率。

(3) 设备管理。按照一定的策略分配和管理输入/输出设备,以保证输入/输出设备高效、有条不紊地工作。

(4) 文件管理。文件是一组相关信息的集合。程序和数据都是以文件的形式存放在计算机外存上的。每个文件必须有一个名字,通过文件名,可以找到对应的文件。文件管理的主要任务是支持文件的存储、查找、删除和修改等操作,保证文件的安全,方便用户使用。

(5) 作业管理。作业是指要求计算机完成的某项任务。作业管理的主要任务是作业调度和作业控制,目的是为用户使用计算机系统提供良好的操作环境,让用户有效地组织工作流程。

微软(Microsoft)公司的Windows操作系统、苹果(Apple)公司的Mac OS X操作系统及Linux操作系统都是目前常用的操作系统。

1.1.3 计算机中的信息表示

1. 二进制数字表示

在十进制系统中有10个数——0、1、2、3、4、5、6、7、8、9,而在二进制系统中只有2个数——0和1。

无论是什么类型的信息,包括数字、文本、图形图像及声音、视频等,在计算机内部都采用二进制形式来表示。

尽管计算机内部均用二进制形式来表示各种信息,但计算机与外部的交往仍采用人们熟悉和便于阅读的形式。计算机的外部信息需要经过转换变为二进制信息后,才能被计算机所接收;同样,计算机的内部信息也必须经过转换后才能恢复信息的“本来面目”。这种转换通常是由计算机

自动实现的。

二进制数往往会很长，读写比较烦琐。因此，常用八进制数或十六进制数来代替表述二进制数。

在八进制系统中有 8 个数——0、1、2、3、4、5、6、7。每个八进制数字相当于 3 位二进制数。

在十六进制系统中有 16 个数——0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。字母 A、B、C、D、E、F 对应十进制数 10、11、12、13、14、15。每个十六进制数字相当于 4 位二进制数。

2. 信息存储单位

(1) 位 (bit)，是计算机内部存储信息的最小单位。一个二进制位只能表示 0 或 1，要想表示更大的数，需把更多的位组合起来。

(2) 字节 (byte)，简记为 B，是计算机内部存储信息的基本单位。一个字节由 8 个二进制位组成，即 $1\text{ B}=8\text{ bit}$ 。在计算机中，其他经常使用的信息存储单位还有千字节 (kilobyte, KB)、兆字节 (megabyte, MB)、吉字节 (gigabyte, GB) 和太字节 (terabyte, TB)，其中 $1\text{ KB}=1\ 024\text{ B}$ ， $1\text{ MB}=1\ 024\text{ KB}$ ， $1\text{ GB}=1\ 024\text{ MB}$ ， $1\text{ TB}=1\ 024\text{ GB}$ 。

(3) 字 (word)，一个字通常由一个字节或若干字节组成，是计算机进行信息处理时一次存取、处理的数据长度。字长是衡量计算机性能的一个重要指标，字长越长，计算机一次所能处理信息的实际位数就越多，运算精度就越高，最终表现为计算机的处理速度越快。常用的字长有 8 位、16 位、32 位和 64 位等。

3. 非数字信息的表示

文本、图形图像及声音、视频之类的信息，称为非数字信息。在计算机中用得最多的非数字信息是文本字符。由于计算机只能处理二进制数，这就需要用二进制的 0 和 1 按照一定的规则对各种文本字符进行编码。

计算机内部按照一定的规则表示文本字符的二进制编码称为机内码。

(1) 西文字符的编码。字符的集合称为“字符集”。西文字符集由字母、数字、标点符号和一些特殊符号组成。字符集中的每一个符号都有一个数字编码，即字符的二进制编码。计算机中使用最广泛的西文字符集是美国标准信息交换码 (American Standard Code for Information Interchange, ASCII) 字符集，其编码称为 ASCII 码，已被国际标准化组织 (International Organization for Standardization, ISO) 采纳，作为国际通用的信息交换标准代码。ASCII 字符集采用一个字节 (8 位) 表示一个字符，所以可以表示 256 个字符。

(2) 中文字符的编码。汉字在计算机中也只能采用二进制编码。汉字的数量大、字形复杂、同音字多。汉字的总数超过 6 万个，常用的也有数千个，显然用一个字节 (8 位) 编码是不够的。GB 2312—1980 是我国于 1981 年颁布的国家标准信息交换用汉字编码字符集，其二进制编码称为国标码。国标码用两个字节 (16 位) 表示一个汉字。GB 2312—1980 字符集的组成：第一部分为字母、数字和各种符号，共 682 个；第二部分为一级常用汉字，按汉语拼音排列，共 3 755 个；第三部分为二级常用汉字，按偏旁部首排列，共 3 008 个。总的汉字数为 6 763 个。GB 2312—1980 字符集的汉字有限，某些汉字无法表示。随着计算机应用的普及，这个问题日渐突出。我国对 GB 2312—1980 字符集进行了扩充，形成了 GB 18030 字符集 (GB 18030—2005)。GB 18030 完全

包含了 GB 2312—1980，共有汉字 27 484 个。

(3) Unicode 编码。随着因特网的迅速发展，信息交换的需求越来越大，不同的编码越来越成为信息交换的障碍，于是 Unicode 编码应运而生。Unicode 编码是由国际标准化组织 ISO 于 20 世纪 90 年代初制定的一种字符编码标准，用多个字节表示一个字符，世界上几乎所有的书面语言都能用单一的 Unicode 编码表示。这样，ASCII 字符与其他字符（如中文字符）的编码就统一起来了，简化了字符处理的过程。

1.2 程序设计基础

1.2.1 程序

1. 程序的定义

广义地说，程序是指为进行某项活动所规定的途径。

我们平时所说的日程安排、会议议程等，都是程序的实例。例如，学校要召开运动会，就需要事先编排好程序，从开幕式到闭幕式，每一项活动的时间、地点、人物、设施、规则、管理、协调等都必须有详细、周密的安排。

2. 程序的执行

程序的执行通常有 3 种方式。例如，在正常情况下，运动会按照程序所设定的顺序执行，这称为程序的顺序执行方式；如果遇到意外，例如下雨、运动员受伤等，还必须要准备相应的应急程序，也就是准备两套或多套方案供选择执行，这就是程序的选择执行方式；而当一项比赛有多组多人反复进行时，只需要一套程序反复执行即可，这就是程序的循环执行方式。

3. 计算机程序

算法是解决某个问题所需要的方法和步骤。

如果以计算机作为工具解决某个问题，必须将解决问题的方法和步骤（算法）告诉计算机。因为人无法与计算机直接交流，所以必须使用程序将算法表示成计算机能够理解的形式，然后让计算机执行程序来完成指定的任务。

计算机程序就是人们为解决某个问题用计算机可以识别的指令合理编排的一系列操作步骤。

1.2.2 程序设计语言

1. 程序设计语言的定义

程序设计语言又称编程语言，是编写计算机程序所使用的语言。程序设计语言是人与计算机交互的工具。人要把需要计算机完成的工作告诉计算机，就需要使用程序设计语言编写程序，让计算机去执行。

2. 程序设计语言的发展

没有程序设计语言的支持，计算机并无实用价值。由于程序设计语言的重要性，从计算机问世至今，人们一直在为研制更好的程序设计语言而努力。程序设计语言的数量在不断增加，各种新的程序设计语言不断问世。

程序设计语言的发展过程是其功能不断完善、描述问题的方法越来越贴近人类思维方式的过程。越接近自然语言的程序设计语言,就越“高级”,反之就越“低级”。越低级的语言,学习和使用难度就越大。

程序设计语言包括机器语言、汇编语言和高级语言。汇编语言和机器语言一般被称为低级语言。

(1) 机器语言是计算机诞生和发展初期使用的语言。机器语言采用二进制编码形式,由 0、1 组成,如 0000010000001110,是计算机唯一可以直接识别、直接运行的语言。机器语言的执行效率高,但不易记忆和理解,编写的程序难以修改和维护,所以现在很少直接用机器语言编写程序。

(2) 为了减轻编写程序的负担,20 世纪 50 年代初发明了汇编语言。汇编语言和机器语言基本上是一一对应的,但在表示方法上作了根本性的改进,引入了助记符,例如,用 ADD 表示加法,用 MOVE 表示传送等。汇编语言比机器语言更加直观,容易记忆,提高了编写程序的效率。计算机不能够直接识别和运行用汇编语言编写的程序,必须通过一个翻译程序将汇编语言转换为机器语言后方可执行。

(3) 世界上第一个高级程序设计语言是 20 世纪 50 年代由 John Backus 领导的一个小组研制的 FORTRAN 语言。高级语言与人们日常熟悉的自然语言和数学语言更接近,便于学习、使用、阅读和理解。高级语言的发明,大大提高了编写程序的效率,促进了计算机的广泛应用和普及。计算机不能够直接识别和运行用高级语言编写的程序,必须通过一个翻译程序将高级语言转换为机器语言后方可执行。常用的高级语言有 C、C++、Java、C#和 Python 等。

3. 语言处理程序

计算机只能直接执行机器语言程序,用汇编语言或高级语言编写的程序都不能直接在计算机上执行。因此,计算机必须配备一种工具,它的任务是把用汇编语言或高级语言编写的程序翻译成计算机可直接执行的机器语言程序,这种工具就是“语言处理程序”。语言处理程序包括汇编程序、解释程序和编译程序。

(1) 汇编程序把汇编语言编写的程序翻译成计算机可直接执行的机器语言程序。

(2) 解释程序对高级语言编写的程序逐条进行翻译并执行,最后得出结果。也就是说,类似于外文翻译中的口译,解释程序对高级语言编写的程序是一边翻译、一边执行的。下次执行同样的程序时,还必须重新翻译。

(3) 编译程序把高级语言编写的程序一次性翻译成计算机可直接执行的机器语言程序。类似于外文翻译中的笔译,翻译一次就可以反复使用,以后每次运行同样的程序时,无须重新翻译。

1.2.3 程序设计

1. 程序设计的定义

程序设计又称编程,是指编写计算机程序解决某个问题的过程。专业的程序设计人员常被称为程序员。

进行程序设计必须具备 4 个方面的知识。

(1) 领域知识。这是给出解决某个问题算法的基础。例如,要解决素数判断的问题,必须了解素数的概念以及素数判断的数学方法,这就是领域知识。如果程序员不具备解决某个问题的领域知识,是不可能编写出解决该问题的计算机程序的。

(2) 程序设计方法。程序设计方法是指合理编排计算机程序内部逻辑的方法。程序员在具备领域知识的基础上, 必须掌握某种程序设计方法, 运用适当的思维方式, 构造出解决某个问题的算法。

(3) 程序设计语言。要使用计算机解决某个问题, 程序员必须掌握某种程序设计语言(如 Python 语言), 运用程序设计语言将算法转换为计算机程序。

(4) 程序设计工具。程序员在程序设计时, 为了提高程序设计的效率和程序的质量, 通常需要使用某种程序设计工具。

2. 程序设计过程

程序设计过程应当包括分析、设计、编码、测试、维护等不同阶段。

(1) 分析阶段的主要任务是了解问题的背景, 理解问题的需求。

(2) 设计阶段的主要任务是针对问题的需求, 设计出解决问题的算法。

(3) 编码阶段的主要任务是运用某种程序设计语言, 将算法转换为程序。

(4) 测试阶段的主要任务是对程序进行严格的测试, 保证程序的质量。只有通过测试的程序才能够交付使用。

(5) 程序在交付使用后, 就进入了维护阶段。人都有可能犯错误, 程序是人的智力产品, 无论经过怎样的严格测试, 程序中通常还是会存在错误的。在程序交付使用后, 也有可能发现程序中的错误, 就需要对程序进行修改。在程序使用过程中, 可能会对程序的功能提出新的需求, 为了实现新的需求, 也需要对程序进行修改。诸如此类的修改工作通常称为对程序的“维护”。

程序设计初学者常犯的错误是拿到问题就开始编程, 忽略了程序设计过程中的分析和设计阶段, 这会严重影响程序设计的质量。初学程序设计时, 教材中的题目都比较简单, 从这些题目中很难体会到上述程序设计过程的作用。尽管如此, 也不应该直接开始编程, 而应该首先分析问题的需求, 设计出合理的算法。

3. 程序设计好坏的界定

程序设计有非常严格的语法规则和很强的逻辑顺序, 所以需要熟练掌握和深入理解相应的语法规则, 并进行大量的逻辑思维训练和编程实践, 才能够设计出好用并可靠的计算机程序。

在保证程序正确的前提下, 可读性、易维护、可移植和高效是程序设计的首要目标。

可读性是指程序清晰, 具有良好的书写风格, 没有太多繁杂的技巧, 能使他人容易读懂。可读性是程序维护的基础, 如果很难读懂程序, 则无法修改程序。

易维护是指当需求发生变化时, 能够比较容易地扩展和增强程序的功能。

可移植是指编写的程序在各种类型的计算机和操作系统上都能正常运行, 且运行结果一致。

1.3 Python 语言的发展历史与特点

1.3.1 Python 语言的发展历史

1. Python 语言的起源

Python 的作者吉多·范罗苏姆(Guido von Rossum)是荷兰人。1982年在阿姆斯特丹大学获

得数学和计算机科学硕士学位。

吉多在荷兰阿姆斯特丹的国家数学和计算机科学研究所工作期间，参与了 ABC 语言的开发。

ABC 语言以教学为目的，设计目标是“让用户感觉更好”，希望程序设计语言变得容易阅读、容易使用、容易记忆和容易学习，以此来激发人们学习编程的兴趣。就吉多本人看来，ABC 语言非常优美和强大，是专门针对非专业程序员设计的。但是，ABC 语言并没有成功，究其原因，吉多认为是非开放造成的。

Python 语言是从 ABC 语言发展起来的。1989 年的圣诞节期间，吉多为了在阿姆斯特丹打发时间，决心开发一种新的程序设计语言，作为 ABC 语言的一种继承。之所以选中 Python 作为语言的名字，是因为他是 BBC 电视剧——《蒙提·派森的飞行马戏团》(Monty Python's Flying Circus) 的爱好者。

Python 的设计哲学：优雅、简单、易于理解。

Python 的语法很多来自于 C 语言，但又受到 ABC 语言的强烈影响。

2. Python 语言的发展

Python 是一种用途广泛、解释型、面向对象的程序设计语言。

最初的 Python 完全由吉多本人开发。Python 得到了吉多同事们的欢迎，他们迅速反馈使用意见，并参与 Python 的改进。吉多和一些同事组成了 Python 的核心团队。随后，Python 拓展到了研究所之外，Python 开始流行。

吉多维护了一个邮件列表 (maillist)，Python 用户通过邮件进行交流。Python 用户来自许多领域，有不同背景，对 Python 有不同需求。Python 开放且易于扩展，当用户不满足于 Python 现有功能时，很容易对 Python 进行拓展改造，将不同领域的优点带给 Python。这些用户将改动发给吉多，由吉多决定是否将这些改动加入到 Python 中。

2000 年发布了 Python 2.0 版本。从 Python 2.0 开始，Python 也从邮件列表的开发方式，转为完全开源的开发方式。Python 社区气氛已经形成，不断壮大，进而拥有了自己的新闻组 (newsgroup)、网站以及基金。工作由整个社区分担，Python 获得了更加高速的发展。

2008 年发布了 Python 3.0 版本。Python 3.0 对语言做了全面的清理和整合，修正了原来语言里的许多缺陷。

Python 现在由 Python 软件基金会 (Python software foundation, PSF) 主导开发和管理。PSF 是一个非营利的国际组织。其网址为 www.python.org。

吉多仍然是 Python 的主要开发者，决定整个 Python 语言的发展方向。Python 社区经常称呼他是“仁慈的独裁者”，意思是他仍然关注 Python 的开发进程，并在必要的时候做出决定。

目前 Python 处于 2.0 和 3.0 共存的情况。Python 2.7 已经被确定为 Python 2.0 的最后版本，PSF 只会对其进行有限的修改完善。PSF 已经集中精力发展 Python 3.0。

Python 2.0 最终还是会被 Python 3.0 所代替。本书使用 Python 3.0 来讲解程序设计。

1.3.2 Python 语言的特点

1. Python 语言的优点

(1) 语言简洁、紧凑，压缩了一切不必要的语言成分。简单易学。

(2) 高效。Python 的底层是用 C 语言编写的，很多标准库和第三方库也都是用 C 语言编写的。

(3) 免费、开源。Python 是自由/开放软件 (free/libre and open source software, FLOSS) 之一。使用者可以自由地发布这个软件的副本、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。

(4) 高层语言。编写 Python 程序时无须考虑诸如如何管理程序使用的内存之类的底层细节。

(5) 可移植性好。Python 是跨平台语言，可以运行在 Windows、Mac OS X 和各种 Linux 系统上。“一次编写，到处运行”。

(6) 面向对象。Python 既支持面向过程的编程，也支持面向对象的编程。

(7) 可扩展性。借助 Python 提供的接口，可以使用 C/C++ 语言对 Python 进行功能性扩展，既可以利用 Python 简洁灵活的语法，又可以获得与 C/C++ 几乎相同的执行性能。

(8) 具有非常丰富的库，除了标准库以外，还有许多第三方高质量的库，而且几乎都是开源的。

(9) 拥有一个积极健康且提供强力支持的社区。这就是为什么 Python 如此优秀的原因之一——它由一群希望看到一个更加优秀的 Python 的人创造并经常改进。

2. Python 语言的不足

(1) 运行速度慢。因为 Python 是解释型语言，源代码在执行时会逐行翻译成 CPU 能理解的机器码，这个翻译过程非常耗时，所以很慢。

(2) 源代码不能加密。如果要发布 Python 程序，实际上就是发布源代码。

(3) 存在版本兼容问题。Python 的 2.0 版和 3.0 版是不兼容的。

3. 有效地使用 Python 语言

(1) 遵循 Python 语言的标准版本。

(2) 采用切合实际的编程规范。编程规范就是程序设计风格。即使程序设计语言本身没有强制要求，程序员也必须遵循规范并坚持使用。良好的编程规范使程序设计风格一致，易于阅读和修改。

(3) 充分利用库函数。不要重复发明轮子 (don't reinvent the wheel)。程序员必须能够熟练使用 Python 本身提供的标准库函数以及第三方经过严格测试的库函数。合理地使用库函数可以减少程序错误，节省编程工作量。

(4) 使用程序设计工具。为了提高程序设计的效率和程序的质量，程序员必须能够熟练使用相应的程序设计工具，充分利用调试功能来修正错误。

1.4 Python 程序开发

1.4.1 基本术语

源程序文件又称源代码文件，简称源文件，是指用 Python 语言编写的、有待翻译的程序，并将程序以文件的形式保存在计算机外存（如硬盘）中。Python 语言源程序文件的文件扩展名通常为 .py。