



工业和信息化“十三五”
人才培养规划教材



国家软件与集成电路公共服务平台 指定教材
信息技术紧缺人才培养工程

黑马程序员 ● 编著



附微课
视频

有问题，就找问答精灵！



Python

程序设计现代方法



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

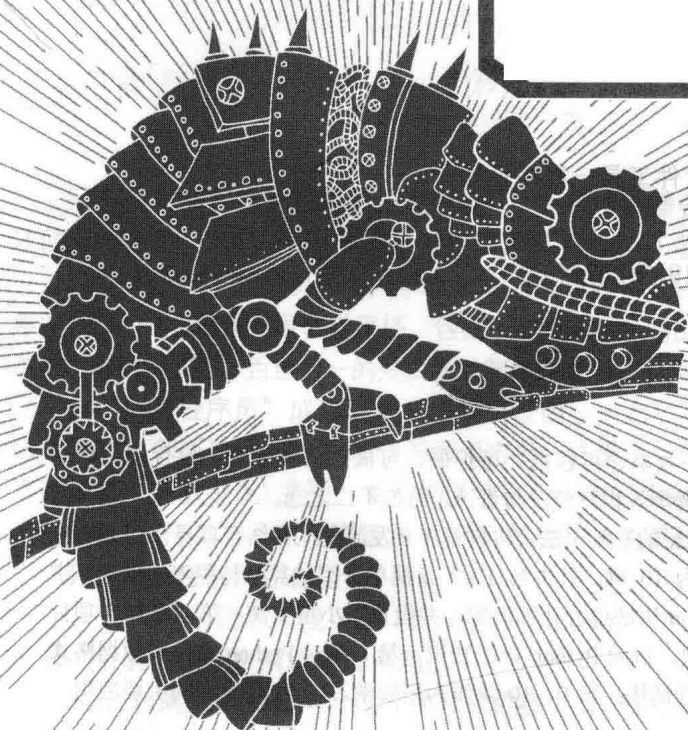


工业和信息化“十三五”
人才培养规划教材

NITE

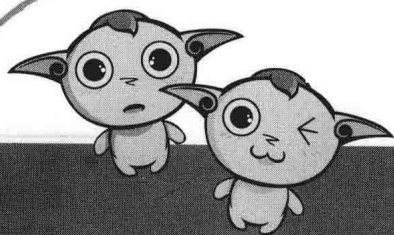
国家软件与集成电路公共服务平台
信息技术紧缺人才培养工程 指定教材

黑马程序员 ● 编著



附微课
视频

有问题，就找问答精灵！



Python

程序设计现代方法

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Python程序设计现代方法 / 黑马程序员编著. — 北京 : 人民邮电出版社, 2019.9
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-51089-1

I. ①P… II. ①黑… 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087640号

内 容 提 要

Python 作为编程语言的一种, 具有高效率、可移植、可扩展、可嵌入、易于维护等优点; 同时 Python 语法简洁, 代码高度规范, 是初学者步入程序开发与设计之路的不二之选。

本书在 Windows 环境下介绍 Python 3.x 的基础语法, 讲解程序开发流程, 并结合项目介绍 Python 常用模块与通用的程序设计方法。本书分为 10 章, 其中第 1、2 章简单介绍计算机与程序的概念, 讲解 Python 的基础语法; 第 3~6 章对 Python 语法进行详细讲解, 并设置了小型实例; 第 7 章结合项目介绍程序设计方法, 同时讲解 Pygame 模块, 巩固 Python 语法知识; 第 8 章对 Python 文件和数据格式化等知识进行讲解, 为大型项目的开发做好铺垫; 第 9、10 章作为拓展学习, 对 Python 的主要应用——数据分析与可视化、网络爬虫进行介绍。

本书附有配套视频、源代码、习题、教学课件等资源, 为帮助初学者更好地学习本书中的内容, 我们还提供了在线答疑, 希望得到更多读者的关注。

本书既可作为高等院校本、专科计算机相关专业及其他工科专业的 Python 教材, 也可作为自学者使用的辅助教材, 是一本适用于程序开发初学者的入门级教材。

◆ 编 著 黑马程序员

责任编辑 范博涛

责任印制 马振武

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

大厂聚鑫印刷有限责任公司印刷

◆ 开本: 787×1092 1/16

印张: 16.25

2019 年 9 月第 1 版

字数: 402 千字

2019 年 9 月河北第 1 次印刷

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

江苏传智播客教育科技股份有限公司（简称传智播客）是一家致力于培养高素质软件开发人才的科技公司，“黑马程序员”是传智播客旗下高端IT教育品牌。

“黑马程序员”的学员多为大学毕业后，想从事IT行业，但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格，包括了严格的技术测试、自学能力测试，还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保学员质量，并降低企业的用人风险。

自“黑马程序员”成立以来，教学研发团队一直致力于打造精品课程资源，不断在产、学、研3个层面创新自己的执教理念与教学方针，并集中“黑马程序员”的优势力量，针对性地出版了计算机系列教材80多册，制作教学视频数十套，发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发IT系列教材，还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务

1. 请同学们登录在线平台 <http://yx.ityxb.com>，进入“高校学习平台”，免费获取海量学习资源。帮助高校学生解决学习问题。

2. 针对高校学生在学习过程中存在的压力等问题，我们还面向大学生量身打造了IT技术女神——“播妞学姐”，可提供教材配套源码、习题答案以及更多学习资源。同学们快来关注“播妞学姐”的微信公众号 [boniu1024](https://www.weixin.com/q/boniu1024)。



“播妞学姐”微信公众号

为教师提供的配套服务

针对高校教学，“黑马程序员”为IT系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源，高校老师请登录在线平台 <http://yx.ityxb.com> 进入“高校教辅平台”或关注码大牛老师微信/QQ：2011168841，获取配套资源，也可以扫描下方二维码，加入专为IT教师打造的师资服务平台——“教学好助手”，获取最新的教学辅助资源。



“教学好助手”微信公众号

前言

Preface

随着计算机的普及与智能设备的发展，人们对操作系统、应用程序、游戏等各种软件的需求量越来越大，各种软件都离不开程序开发，因此社会对各种程序的开发人员，如 Python、C、C++、Java、PHP 等开发人员的需求量也不断提升。2016 年，AlphaGo 击败人类职业围棋选手，引起了人工智能和 Python 语言的热潮；2018 年 3 月，Python 成为我国计算机等级考试二级考试新增科目，再次提升了 Python 语言的重要性。

◆ 为什么选择本书

Python 语言语法简单，但语言只是工具，程序开发与设计并非只是对语言的学习，编程思维与程序设计思想才是重中之重。掌握基础语法和理论只是第一步，若想拥有编程能力，必须动手实践；若要编写优秀的代码，更应该结合程序设计思维。

本书在讲解时采用理论与实践相结合的方式，我们为每章配备了实践案例，先对相关知识进行讲解，再以实践案例对相关知识进行巩固。本书语言通俗易懂，相关案例精练实用，旨在帮助读者学习理论知识的同时，提高学习兴趣，强化动手能力。

◆ 如何使用本书

本书在 Windows 平台基础上对 Python 3.x 的语法及程序设计的相关知识进行讲解，全书分为 10 章，各章内容分别如下。

第 1 章首先介绍了程序的载体——计算机的相关知识，包括计算机的诞生、发展和工作原理，其次介绍了计算机语言及执行方式，然后简单介绍了 Python 语言，包括该语言的发展史、2.x 版本和 3.x 版本之间的区别、语言的特点及应用领域，之后介绍了在 Windows 系统中配置 Python 开发环境、运行 Python 程序的方式，最后简单介绍了程序的基本编写方法。通过对本章内容的学习，希望读者能对计算机有所了解，理解人类通过程序使用计算机的过程，成功搭建 Python 开发环境，掌握运行 Python 程序的方式，并了解程序开发与编写方法。

第 2 章结合实例首先介绍 Python 程序的要素，包括 Python 程序的代码风格、变量、输入/输出语句、结构控制语句及函数式编程思想，其次介绍了模块化编程思想、模块的导入和使用方法，最后介绍了 Python 中的绘图模块——turtle。通过对本章内容的学习，希望读者能够熟悉程序设计的流程，了解 Python 程序要素，掌握模块化编程思想，并能利用 turtle 模块绘制简单图形。

第 3 章主要介绍 Python 的数据类型的相关内容，包括数字类型和字符串，其次介绍了数学模块 math。通过对本章内容的学习，希望读者能够熟练地使用基本数据类型，为后续的开发打好基础。

第4章主要讲解程序表示方法、分支结构、循环结构及异常处理。通过对本章内容的学习，希望读者可对程序表示方法有所了解，并能熟练运用不同的结构控制程序流程，运用异常处理结构处理异常。

第5章主要介绍与函数相关的知识，包括函数的概念、定义、调用过程、参数传递、作用域及特殊形式的函数——匿名函数和递归函数。本章也对代码抽象与模块化设计的思想进行了简单介绍。通过对本章内容的学习，希望读者能熟悉函数的相关知识，并能掌握定义函数和使用函数的方法。

第6章主要介绍 Python 的组合数据类型，包括列表、元组、集合和字典。通过对本章内容的学习，希望读者能够熟悉组合数据类型的分类及特点，并能在程序中熟练运用组合数据类型表示和存储数据。

第7章结合项目——数字推盘，介绍了 MVC 设计模式、自顶向下的设计方法、自底向上的实现方法及 Python 游戏模块 pygame 的基础用法。通过本章的学习，希望读者能够掌握 pygame 模块的用法，了解 MVC 设计模式，并能熟练使用自顶向下方法设计程序。

第8章主要讲解文件和数据格式化相关的知识，包括计算机中文件的定义、文件的基本操作、文件迭代、文件操作模块 os 及数据维度和数据格式化等。通过对本章内容的学习，希望读者能够了解计算机中文件的意义，熟练读取、更改文件，熟悉文件操作模块，并了解常见的数据组织形式。

第9章介绍数据分析的概念、科学计算工具 numpy、数据可视化工具 matplotlib 的模块 pyplot、数据分析工具 pandas，并结合实例演示数据分析工具的用法。通过对本章内容的学习，读者能够掌握数据分析工具的用法，具备使用数据分析工具分析数据的能力。

第10章讲解网络爬虫相关的知识，包括网络爬虫的概念、原理、实现过程、实现网络爬虫功能的第三方模块 requests 和 Beautiful Soup 4，并结合实例演示如何开发简单的爬虫项目。通过对本章内容的学习，希望读者能够了解爬虫的基本原理，具备开发简单爬虫项目的能力。

读者在学习的过程中，务必要勤于练习，确保真正掌握所学知识。读者若在学习的过程中遇到无法解决的困难，不要纠结，继续往后学习，或可豁然开朗。

◆ 致谢

本书的编写和整理工作由传智播客教育科技股份有限公司完成，主要参与人员有吕春林、高美云、王晓娟、郑瑶瑶等，全体人员在近一年的编写过程中付出了很多，在此一并表示衷心的感谢。

◆ 意见反馈

尽管我们做了最大努力，但书中难免会有不妥之处，欢迎各界专家和读者朋友们来信给予宝贵意见，我们将不胜感激。您在阅读本书时，如发现任何问题，可以通过电子邮件与我们联系。

请发送电子邮件至 itcast_book@vip.sina.com。

黑马程序员
2019年4月于北京

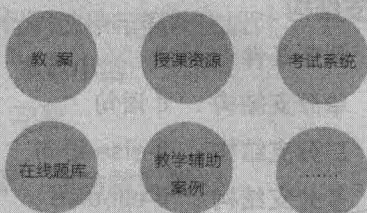
目录

Content

专属于老师及学生的在线教育平台
<http://yx.ityxb.com/>

让 IT 教学更简单

教师获取教材配套资源



添加微信/QQ

2011168841

让 IT 学习更有效

学生获取配套源码

关注微信公众号“播妞学姐”
获取教材配套源码



专属大学生的圈子

第 1 章 Python 概述	001
1.1 计算机与计算机语言	002
1.1.1 计算机的诞生与发展	002
1.1.2 计算机语言概述	006
1.1.3 翻译执行	008
1.2 Python 语言概述	009
1.2.1 Python 语言发展史	009
1.2.2 Python 语言的特点	011
1.2.3 Python 的应用领域	012
1.2.4 Python 版本的区别	012
1.3 Python 环境配置	014
1.3.1 安装 Python 解释器	014
1.3.2 Python 程序的运行方式	017
1.3.3 运行 Python 程序	018
1.4 集成开发环境	021
1.4.1 PyCharm 的下载和安装	021
1.4.2 PyCharm 的使用	024
1.5 程序的开发与编写	027
1.5.1 程序开发流程	027
1.5.2 程序编写的基本方法	028
1.6 本章小结	029
1.7 习题	030
第 2 章 Python 实例设计与分析	032
2.1 实例 1：货币兑换	033
2.2 代码风格	035
2.2.1 缩进	035
2.2.2 注释	035
2.3 变量	036

2.3.1 标识符和关键字	036	3.6.3 字符串操作符	075
2.3.2 数据类型	037	3.6.4 字符串处理函数	076
2.3.3 变量赋值	037	3.6.5 字符串处理方法	076
2.3.4 字符串索引和切片	038	3.7 实例 5：过滤敏感词	080
2.4 基本输入 / 输出	039	3.8 本章小结	081
2.4.1 input() 函数	039	3.9 习题	081
2.4.2 eval() 函数	039	第 4 章 流程控制	082
2.4.3 print() 函数	040	4.1 程序表示方法	083
2.5 结构控制	041	4.1.1 程序流程图	083
2.5.1 顺序结构	041	4.1.2 程序的基本结构	084
2.5.2 分支结构	042	4.1.3 使用程序流程图描述程序	086
2.5.3 循环结构	043	4.2 分支结构	087
2.6 函数式编程	043	4.2.1 判断条件	088
2.7 实例 2：Turtle Star	044	4.2.2 单分支结构：if 语句	089
2.8 模块化编程	045	4.2.3 二分支结构：if-else 语句	089
2.8.1 模块的导入和使用	045	4.2.4 多分支结构：if-elif-else 语句	090
2.8.2 模块的分类	046	4.2.5 分支嵌套	093
2.9 绘图模块——turtle	049	4.3 循环结构	095
2.10 本章小结	054	4.3.1 条件循环：while 循环	095
2.11 习题	054	4.3.2 遍历循环：for 循环	096
第 3 章 数字类型和字符串	056	4.3.3 循环控制	097
3.1 数字类型	057	4.4 异常处理	099
3.1.1 整型	057	4.5 实例 6：猜数字	100
3.1.2 浮点型	058	4.6 本章小结	102
3.1.3 复数类型	059	4.7 习题	102
3.1.4 布尔类型	060	第 5 章 函数	104
3.2 数字类型的运算	060	5.1 函数概述	105
3.2.1 数值运算符	060	5.2 函数的基础知识	106
3.2.2 运算符优先级	064	5.2.1 函数的定义	106
3.2.3 数字类型转换	065	5.2.2 函数的调用	107
3.3 实例 3：模拟商家收银抹零行为	065	5.3 函数的参数传递	108
3.4 数学模块——math	066	5.3.1 参数的位置传递	108
3.5 实例 4：三天打鱼两天晒网	070	5.3.2 参数的关键字传递	109
3.6 字符串	070	5.3.3 参数的默认值传递	109
3.6.1 字符串的定义方式	071		
3.6.2 字符串格式化	072		

5.3.4 包裹传递	110	第 7 章 程序设计之数字推盘	146
5.3.5 解包裹传递	110	7.1 数字推盘游戏简介	147
5.3.6 混合传递	111	7.2 游戏模块——pygame	149
5.4 函数的返回值	112	7.2.1 pygame 的初始化和退出	149
5.5 变量作用域	112	7.2.2 创建游戏窗口	150
5.5.1 局部变量	113	7.2.3 游戏循环与游戏时钟	152
5.5.2 全局变量	113	7.2.4 图形和文本绘制	153
5.6 函数的特殊形式	114	7.2.5 元素位置控制	158
5.6.1 匿名函数	115	7.2.6 动态效果	161
5.6.2 递归函数	115	7.2.7 事件与事件处理	164
5.7 时间处理模块——datetime	117	7.3 游戏框架	167
5.8 实例 7：模拟钟表	121	7.4 自顶向下的设计	167
5.9 代码抽象与模块化设计	125	7.4.1 顶层设计	168
5.10 本章小结	126	7.4.2 子层设计	169
5.11 习题	127	7.4.3 第三层设计	174
第 6 章 组合数据类型	128	7.4.4 模块整合	178
6.1 组合数据类型概述	129	7.5 自底向上的实现	179
6.1.1 序列类型	129	7.6 运行数字推盘	180
6.1.2 集合类型	130	7.7 本章小结	182
6.1.3 映射类型	131	7.8 习题	182
6.2 序列类型	131	第 8 章 文件和数据格式化	183
6.2.1 切片	131	8.1 文件概述	184
6.2.2 列表	133	8.2 文件的基本操作	186
6.2.3 元组	135	8.2.1 文件的打开和关闭	186
6.3 实例 8：生成验证码	136	8.2.2 读文件	188
6.4 集合类型	137	8.2.3 写文件	189
6.4.1 集合的常见操作	137	8.2.4 文件读写位置	191
6.4.2 集合关系测试	138	8.2.5 管理文件与目录	193
6.5 字典	140	8.3 文件迭代	194
6.5.1 字典类型介绍	140	8.4 实例 10：用户登录	195
6.5.2 字典的常见操作	140	8.5 数据维度与数据格式化	201
6.6 中文分词模块——jieba	141	8.5.1 基于维度的数据分类	201
6.7 实例 9：《西游记》人物出场统计	143	8.5.2 一二维数据的存储与读写	202
6.8 本章小结	145	8.5.3 多维数据的格式化	205
6.9 习题	145	8.6 Python 中的 json 模块	206
		8.7 本章小结	207

8.8 习题	207	9.6 本章小结	233
第 9 章 数据分析与可视化	209	9.7 习题	233
9.1 数据分析概述	210	第 10 章 网络爬虫	235
9.1.1 数据分析的流程	210	10.1 网络爬虫概述	236
9.1.2 数据分析常用工具	211	10.2 爬虫爬取网页的流程	236
9.2 科学计算	212	10.3 抓取网页数据	237
9.2.1 numpy 之数组对象 ndarray	212	10.3.1 浏览网页过程	237
9.2.2 numpy 的基本操作	214	10.3.2 使用 requests 模块抓取 网页	239
9.3 数据可视化	216	10.4 解析网页数据	240
9.3.1 数据可视化概述	216	10.4.1 网页数据结构分析	240
9.3.2 pyplot 之绘图区域	217	10.4.2 解析网页的过程和技术	241
9.3.3 pyplot 之图表与风格控制	219	10.4.3 使用 Beautiful Soup 4 解析网页 数据	242
9.4 实例 11：各学科成绩评估分析	223	10.5 实例 12：龙港房地产爬虫	244
9.5 数据分析	226	10.6 本章小结	248
9.5.1 pandas 数据结构	226	10.7 习题	249
9.5.2 pandas 的基本使用	228		

P ython 程序设计现代方法

第 1 章

Python 概述



学习目标

- ★了解计算机语言的分类，熟悉高级语言的翻译执行过程
- ★了解 Python 版本的区别以及 Python 语言的特点和应用领域
- ★熟练搭建 Python 开发环境
- ★掌握 Python 程序的运行方式
- ★了解程序开发流程及编写方法

计算机是 20 世纪最伟大的发明之一，自诞生至今，计算机的相关技术和行业得到了蓬勃发展。目前，计算机已经成为陪伴人类生活、工作、学习的重要伙伴，我们编撰文档、网上交友，听音乐、打游戏、绘图等都离不开计算机。计算机的正常工作离不开程序，程序是一系列计算机指令，不同的程序可为人们提供不同的服务。程序的编写离不开编程语言。接下来，本章将从计算机的诞生与发展入手，带领大家逐步认识计算机、计算机程序以及 Python 语言。

1.1 计算机与计算机语言

1.1.1 计算机的诞生与发展

虽然代表通用电子计算机产生的 ENIAC 诞生于 1946 年，但“计算机”这一天才想法在 10 年之前已被提出。1936 年，年仅 24 岁的英国数学家、逻辑学家图灵（Alan Turing）向伦敦权威的数学杂志投送了一篇题为《论数字计算在决断难题中的应用》的论文，并在该论文的附录中描述了一种可以辅助数学研究的机器。图灵设想该机器可以模拟人类用纸笔进行数学运算的过程，他将这个过程视为下列两项简单操作。

- (1) 在纸上写上或擦除某个符号。
- (2) 将注意力从纸的一个位置移动到另一个位置。

人在运算的每个阶段又会根据以下两点来决定下一步的动作。

- (1) 人当前所关注的纸上某个位置的符号。
- (2) 人当前思维的状态。

为了模拟人力运算过程，图灵将构想出的机器分为以下几个组成部分。

- (1) 一条无限长的纸带。这条纸带被划分为连续的小格子，每个格子包含一个来自有限字母表的符号，格子从左至右依次被编号为 0, 1, 2, …，纸带的右端无限伸展。

- (2) 一个读写头。读写头可以在纸带上左右移动、读取当前所指格子上的符号，并能改变格子中的符号。

- (3) 一个状态寄存器。用来存储机器当前所处的状态。图灵机的状态是有限的，且有一个称为“停机状态”的特殊状态。

- (4) 一套控制规则。可根据机器当前的状态以及读写头当前所指格子中存储的符



号来确定读写头下一步的动作，并改变寄存器的值，使机器进入一个新的状态。

人们将图灵描述的机器模型称为“图灵机”（Turing Machine）。图灵认为图灵机可以模拟人类所能进行的任何计算过程。图灵机的结构模型如图 1-1 所示。

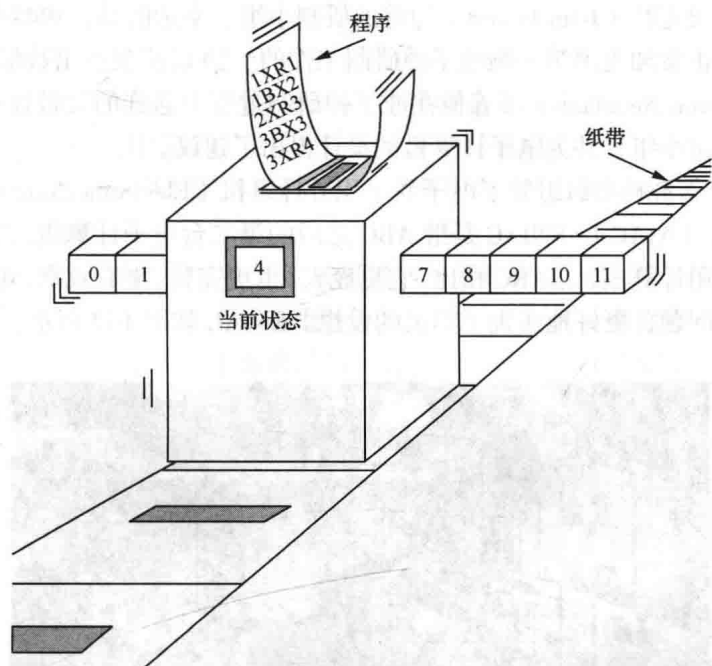


图 1-1
图灵机模型

一台图灵机可以计算一个事先设定的问题，任意一台图灵机的描述都是有限的，因此可以某种方式将其编码为字符串。假设以 $\langle M \rangle$ 表示图灵机 M 的编码，现有一台图灵机 U ，它可以接收任意一台图灵机 M 的编码 $\langle M \rangle$ ，模拟 M 的运作，那么图灵机 U 就被称为通用图灵机（Universal Turing Machine）。通用图灵机可以解决各种各样的问题。

虽然图灵只是提出了图灵机的设想，并未将其加以实现，但这一设想解决了纯数学基础理论问题，并证明了研制通用数字计算机的可行性，为后来通用计算机的出现奠定了理论基础。图灵创造性地指出了人类科学发展的新方向，他也因这一贡献而被称为“计算机科学之父”。

1939 年，美国的阿坦那索夫（John Atanasoff）和其助手贝瑞（Clifford E. Berry）设计并组装了世界上第一台电子数字计算设备 ABC（Atanasoff-Berry Computer），这台设备不可编程，仅设计用于求解线性方程组，并在 1942 年成功进行了测试。

ABC 是世界上第一台电子计算机，它初步实现了图灵机的设想，但受战争影响，阿坦那索夫未能完整实现设计之初的全部理念，尽管如此，这台机器仍为现代计算机的产生奠定了基础。

第二次世界大战期间，各国迫切希望研发新型大炮和导弹，美国陆军军械部也为此设立了“弹道研究实验室”，并要求该实验室每天为陆军炮弹部队提供 6 张射击表以便对导弹的研制技术进行鉴定。但每张射击表的产生都需要进行大量且复杂的运算，即便实验室雇用 200 多名计算员加班加点工作也需两个多月才能完成一张。这显

然根本无法满足研发需求。

1942年,宾夕法尼亚大学莫尔电机工程学院的莫希利(John Mauchly)提出了试制一台电子计算机的设想(莫希利对电子计算机的设想实际上源于阿坦那索夫),他期望使用计算机提高运算效率。美国军方得知这一设想后给予了极大支持,并成立了以莫希利、埃克特(John Eckert)为首的研制小组。幸运的是,1944年,时任弹道研究所顾问、正参加美国第一颗原子弹研制工作的、20世纪最杰出数学家之一的冯·诺依曼(John von Neumann)带着他在原子弹研制过程中遇到的大量计算问题于研制中期加入了研制小组,并为电子计算机的设计提出了建议。

1946年,莫希利小组组装了电子数字积分计算机(Electronic Numerical Integrator And Computer, ENIAC)。ENIAC是继ABC之后的第二台电子计算机,也被认为是世界上第一台通用计算机,与ABC相比,它更庞大,也更完善、更有效率,可以重新编程、解决各种计算问题,更好地实现了图灵的设计。ENIAC如图1-2所示。

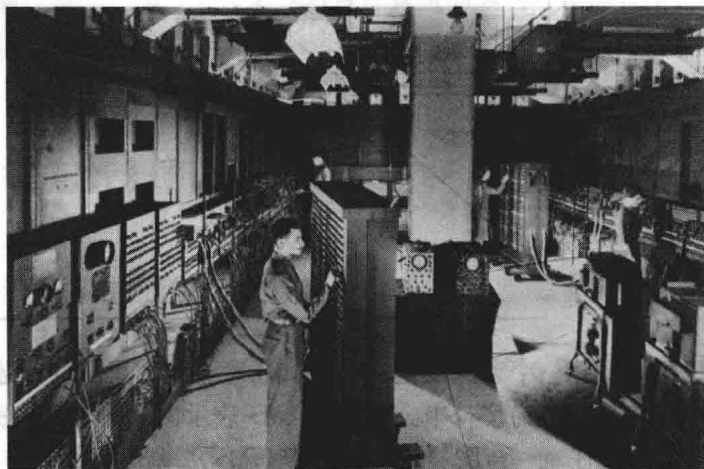


图 1-2
ENIAC

ENIAC存在两个问题:一是没有存储器,二是程序用布线接板进行控制,每次更换程序需要重新搭接布线接板,这一过程甚至要耗费几天,大大降低了计算效率。

1945年,冯·诺依曼和他的研制小组经过讨论,发表了一个全新的电子计算机方案——电子离散变量自动计算机(Electronic Discrete Variable Automatic Computer, EDVAC)。1950年,EDVAC在宾夕法尼亚诞生。

在EDVAC设计方案中,冯·诺依曼提出以下要点。

- (1) 将程序本身视为数据,以存储数据的方式,将程序预先存放到存储器中。
- (2) 计算机的数制采用二进制,计算机应该按照程序顺序执行。

冯·诺依曼的这些理论被称为冯·诺依曼体系结构,后来人们设计的计算机基本都沿用此体系结构。冯·诺依曼因此项贡献被称为“计算机之父”,成为计算机与计算机科学发展史中与图灵并驾齐驱的奠基人。

早期产生的计算机并未将程序存放到存储器中,所有计算机程序的编制都在计算机外部实现,直到冯·诺依曼带队建造了计算机EDVAC,计算机的体系结构才发生改变。自1950年EDVAC产生之后,世界各地设计的计算机基本都遵循冯·诺依曼体系结构。冯·诺依曼型计算机的发展大致分为五代。



1. 第一代计算机

第一代计算机（1950—1959年）体积庞大，通常被锁在房子中，只有操作者与计算机专家才能进入；使用真空电子管作为电子开关，造价昂贵，一般为大型企业拥有。此时的计算机操作难度较大，操作者需了解计算机的电子结构细节才能进行编程。

2. 第二代计算机

第二代计算机（1959—1965年）采用晶体管代替了真空电子管，此举减小了计算机的体积，也节省了开支，使计算机成为中小型企业有能力负担的设备。此阶段出现了比较高级的程序设计语言，编程人员无须了解计算机的电子结构细节，只需掌握语言便可编写程序，编程任务和计算机的运算任务自此被分离开来。

3. 第三代计算机

第三代计算机（1965—1975年）采用集成电路代替了晶体管，计算机的体积和成本再度降低，小企业可以负担的小型计算机开始出现。另外市面亦有程序出售，计算机用户可直接购买程序，而不必自己编写程序，软件行业就此诞生。

4. 第四代计算机

第四代计算机（1975—1985年）使用单块电路板搭载整个计算机子系统，计算机的体积再度缩小。由于与第一代计算机相比，第四代计算机的体积已经非常微小，此时的计算机也被称为微型计算机。此阶段诞生了第一个桌面计算机——Altair 8800，Altair 8800如图1-3所示。



图 1-3
Altair 8800

5. 第五代计算机

1985年至今产生的计算机被称为第五代计算机。第五代计算机包括台式机、笔记本电脑、平板电脑等设备。与之前的计算机相比，第五代计算机在容量、体积以及性能等方面都有了很大的提升。

虽然计算机已历经60余年的发展，计算机的体积、性能、价格都有了很大的改变，但改变的主要是硬件或软件，计算机模型并未发生改变，依然沿用冯·诺依曼模型。可以说，冯·诺依曼是当之无愧的计算机之父。

多学一招：冯·诺依曼模型

基于冯·诺依曼型体系结构设计的计算机必须具有如下功能。

- (1) 将需要的程序和数据发送至计算机中。
- (2) 可长期记忆程序、数据、中间结果以及最终运算结果。
- (3) 具有完成各种算术、逻辑运算和数据传送等数据加工处理的能力。

(4) 能够根据需要控制程序走向, 并能根据指令控制机器的各部件协调运作。

(5) 能够按照要求将处理结果输出给用户。

以上列出的这些功能与冯·诺依曼体系结构对计算机模型的设计有关。此模型将计算机分为存储器、控制器、运算器、输入设备和输出设备 5 个部分。这 5 个部分之间的关系如图 1-4 所示。

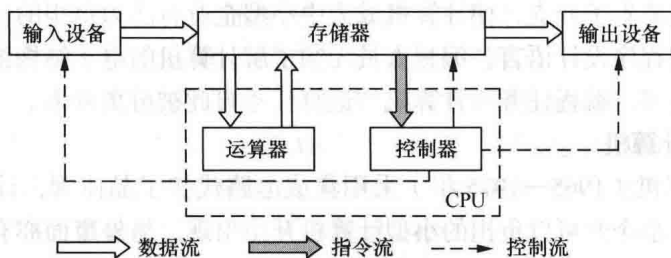


图 1-4
冯·诺依曼模型

1.1.2 计算机语言概述

计算机擅长接受指令, 但不能识别人类的语言, 人类为保证计算机可以准确地执行指定的命令, 需要使用计算机语言向计算机发送指令。计算机语言是用于编写计算机指令, 即编写程序的语言, 其本质是根据事先定义的规则编写的预定语句的集合。

计算机语言分为 3 类: 机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是由 0、1 组成的二进制代码表示的指令。这类语言可以被 CPU 直接识别, 具有灵活、高效等特点。但机器语言有个不可忽视的缺点: 可移植性差。各公司生产的不同系列、不同型号的计算机使用的机器语言是不同的, 编程人员使用机器语言为一台计算机编写程序之前必须先熟记此台计算机的全部指令代码和代码的含义, 写出的程序只能在同一款机器中使用, 且不直观、容易出错, 错误又难以定位。一段表示两个整数相加的机器指令如下所示:

```
0001 1111 1110 1111
0010 0100 0000 1111
0001 1111 1110 1111
0010 0100 0001 1111
0001 0000 0100 0000
0001 0001 0100 0001
0011 0010 0000 0001
0010 0100 0010 0010
0001 1111 0100 0010
0010 1111 1111 1111
0000 0000 0000 0000
```

机器语言是第一代编程语言, 早期的计算机语言只有机器语言, 但如今已罕有人学习和使用。

2. 汇编语言

汇编语言用带符号或助记符的指令和地址代替二进制代码, 因此汇编语言也被称



为符号语言。使用汇编语言编写实现两个整数相加的程序，具体代码与说明如表 1-1 所示。

表 1-1 汇编代码示例与说明

代 码				说 明
LOAD	RF	Keyboard		从键盘获取数据，存到寄存器 F 中
STORE	Number1	RF		把寄存器 F 中的数据存入 Number1
LOAD	RF	Keyboard		从键盘获取数据，存到寄存器 F 中
STORE	Number2	RF		把寄存器 F 中的数据存入 Number2
LOAD	R0	Number1		把 Number1 中的内容存入寄存器 0
LORD	R1	Number2		把 Number2 中的内容存入寄存器 1
ADD1	R2	R0	R1	寄存器 0 和寄存器 1 中内容相加，结果存入寄存器 2
STORE	Result	R2		把寄存器 2 中的内容存入 Result
LOAD	RF	Result		把 Result 中的值存入寄存器 F
STORE	Monitor	RF		把寄存器 F 中的值输出到显示器
HALT				停止

表 1-1 所示的每一条汇编指令对应前面所示的机器语言编写的一行代码。与机器语言相比，汇编语言的可读性有所提高，但汇编语言是一种面向机器的低级语言，是一种为特定计算机或同系列计算机专门设计的语言。换言之，为一种设备编写的汇编指令只能用于和此台设备同系列、具有同型号 CPU 的设备中，可移植性仍然很差，对编程人员的要求仍然较高。但也正因为汇编语言与机器的相关性，它可以较好地发挥机器的特性。此外，汇编语言保持了机器语言的优点，它也可以直接访问和控制计算机硬件，占用内存少，且执行速度快。

汇编语言是第二代编程语言，在某些行业和领域中，汇编语言是必不可少的语言；对底层程序设计人员而言，汇编语言是必须了解的语言。需要注意的是，汇编语言无法被计算机识别，在执行之前需要先使用被称为“汇编程序”的特殊程序将汇编语言代码翻译成机器语言代码。

3. 高级语言

由于与硬件相关性较高，且符号与助记符量大又难以记忆，编程人员在开发程序之前需要花费相当多精力去了解、熟悉设备的硬件，以及目标设备的助记符。为了从硬件中脱身，专注程序功能的研发，提高程序开发效率，一些编程人员开始研究高级语言。高级语言与设备硬件结构无关，它更接近自然语言，对数据的运算和程序结构表述得更加清晰、直观，人们阅读、理解和学习编程语言的难度也大大降低。

高级语言并非一种语言，而是诸多编程语言的统称。常见的高级语言有 Python、C、C++、Java、JavaScript、PHP、Basic、C# 等。例如 Python 语言中，实现两个整数相加的代码具体如下：

```
num1 = int(input("加数：")) # 从键盘获取内容后转换为整型，存入 num1
num2 = int(input("被加数：")) # 从键盘获取内容后转换为整型，存入 num2
result = num1 + num2 # 计算 num1+num2，并将结果存入 result
print(result) # 将结果 result 打印到屏幕
```

比较以上 Python 代码与汇编代码，显然 Python 代码更加简洁直观。此外高级语