



中国科学技术大学
University of Science and Technology of China

计算机程序设计

第一章 预备知识

白雪飞

中国科学技术大学微电子学院

目录



- 引言
- 组织架构
- 计算与计算思维
- 编程语言与程序设计
- 小结



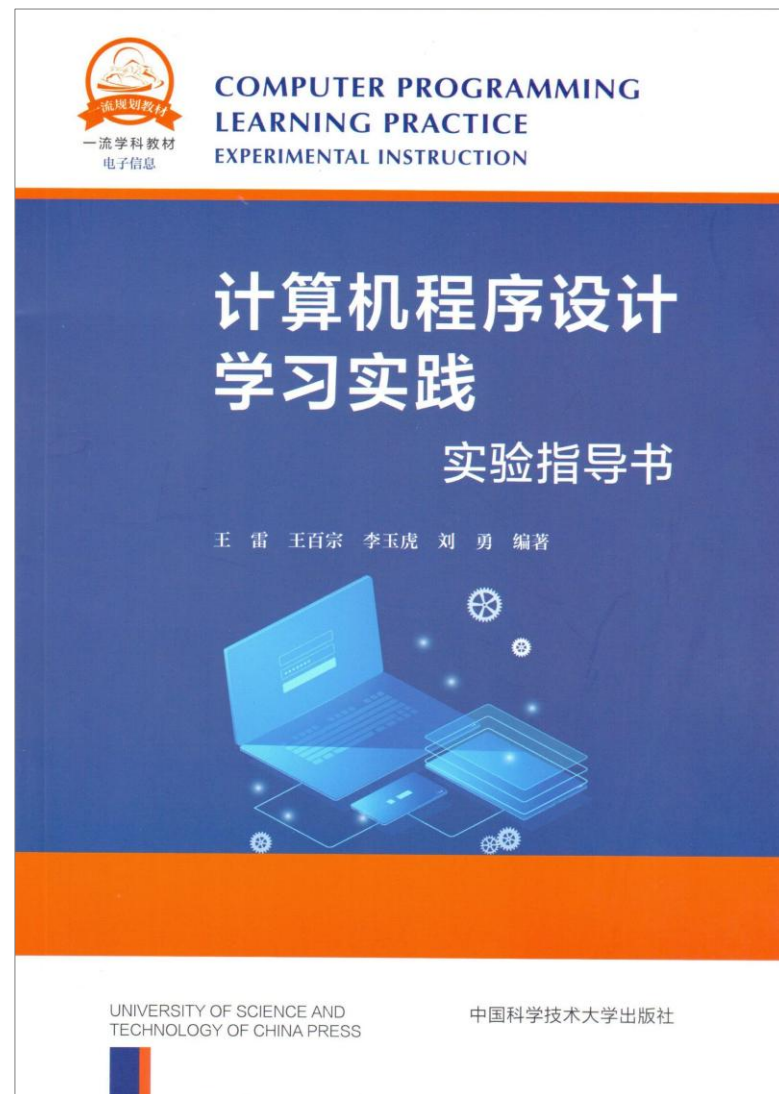
引言

■ 教学目标

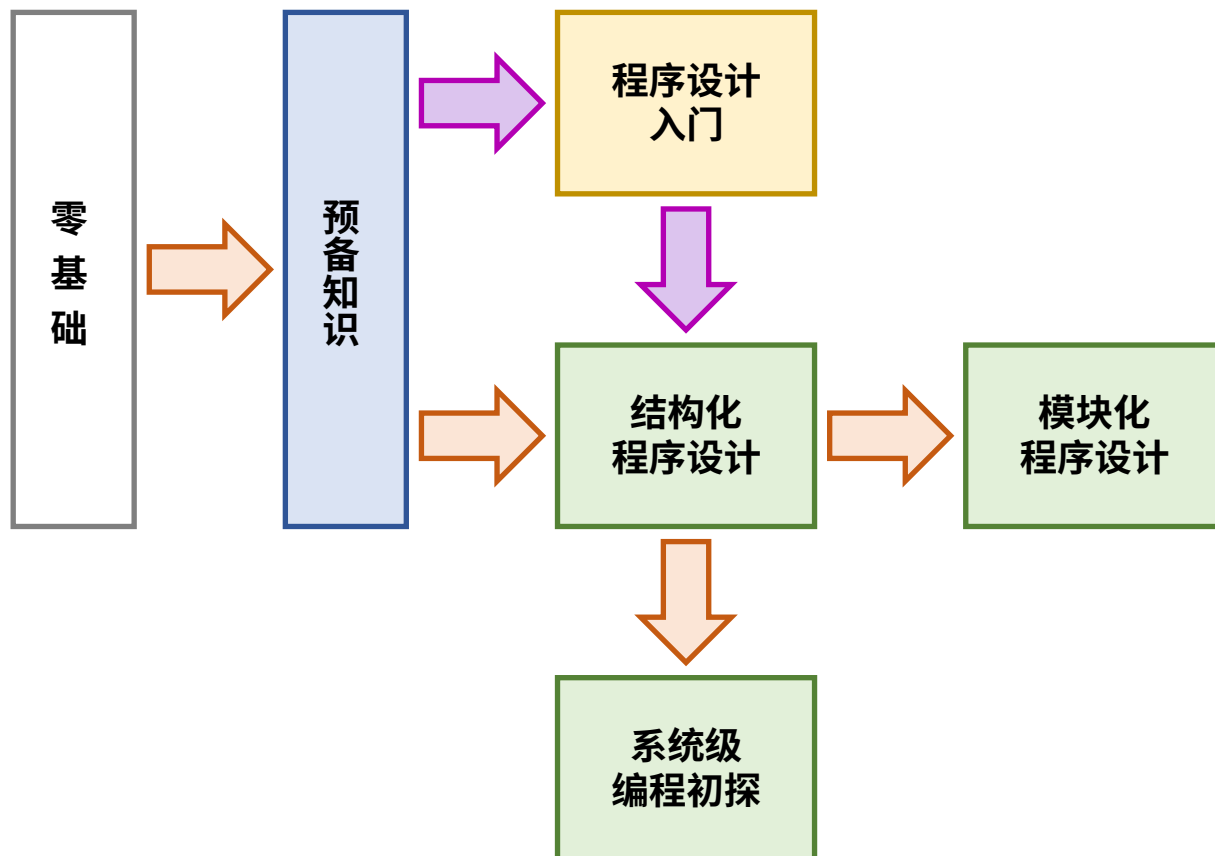
- 从零基础开始培养编程实践能力
- 在编程实践中锻炼培养计算思维与自主学习能力
- 理解与掌握C语言的常用语法元素和程序设计方法
- 利用编程方法解决复杂的问题，提高计算效率

■ 考核方式

- 平时作业 ($\leq 20\%$): 作业量各班自定，依据作业完成情况
- 上机实验 ($\geq 20\%$): 实验题各班自定，依据程序检查情况、实验报告完成情况
- 上机考试 ($\geq 20\%$): 考试内容和时间各班自定，依据考试系统评分
- 期末笔试 ($\geq 40\%$): 课程组统一安排
 - 题型：单选、多选、程序填空、读程序写运行结果、算法描述
 - 算法描述题：给出程序功能，要求画流程图或写伪代码
 - 不再考手写程序题
- 本课程由于有实验和机试，不安排补考，不及格只能重修



组织架构



课程内容组织架构



计算与计算思维

- 计算与进制
- 计算机体系结构
- 信息与编码
- 文件与地址
- 算法与逻辑
- 计算思维



Babylonian Tablet Showing the Sexagesimal Number Approximating $\sqrt{2}$, 1800–1600 BCE

■ 数制与进制

- 数制：使用一组数字符号表示数的体系
- 进制：二进制、十进制、十二进制、二十进制、六十进制等
- 非进制：罗马数字
- 计算的基础是进制

■ R 进制

- 基数：所使用的基本符号的数量， R
- 符号：从0开始的最小的 R 个自然数， $0, 1, \dots, R-1$
- 权值：符号在数中不同位置的单位， R 的整数次幂

$$A = a_{n-1} a_{n-2} \cdots a_2 a_1 a_0 . a_{-1} a_{-2} \cdots a_{-m} = \sum_{i=-m}^{n-1} (a_i \cdot R^i)$$

计算机中的进制

■ 二进制编码

- 计算机中所有信息都采用二进制编码进行存储和处理
- 计算机与用户交流采用人们熟悉和便于阅读的形式
- 计算机硬件和软件完成上述形式与二进制编码之间的转换

■ 二进制的优点

- 技术实现简单
- 运算规则简单
- 抗干扰能力强
- 便于逻辑运算
- 二进制数的位数较长

计算机中常用的进制

进制	英语	基数	基本符号	后缀
二进制	Binary	2	0, 1	B
八进制	Octal	8	0, 1, 2, 3, 4, 5, 6, 7	Q
十进制	Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	D
十六进制	Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	H

■ R 进制转换为十进制

- 按照进制的公式在十进制下计算

■ 十进制转换为 R 进制

- 整数部分：除基取余法
- 小数部分：乘基取整法

■ 二、八、十六进制之间的转换

- 每三位二进制数对应一位八进制数
- 每四位二进制数对应一位十六进制数
- 二进制数分组不足三位或四位时用“0”补足
- 二进制数分组不应越过小数点

八、十六进制数与二进制分组

八进制	二进制	十六进制	二进制
0	000	0	0000
1	001	1	0001
2	010	2	0010
3	011	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
		8	1000
		9	1001
		A	1010
		B	1011
		C	1100
		D	1101
		E	1110
		F	1111

■ 例1.3-1: R 进制转换为十进制举例。

$$\begin{aligned}(8\text{FC}7.2\text{A})_{16} &= 8 \times 16^3 + 15 \times 16^2 + 12 \times 16^1 + 7 \times 16^0 + 2 \times 16^{-1} + 10 \times 16^{-2} \\ &= (36807.1640625)_{10}\end{aligned}$$

■ 例1.3-2: 二进制转换为八、十六进制举例。

$$(1011010.1)_2 = (\textcolor{red}{00}1, 011, 010.\textcolor{red}{100})_2 = (132.4)_8$$

$$(1011010.1)_2 = (\textcolor{red}{0}101, 1010.\textcolor{red}{1000})_2 = (5\text{A}.8)_{16}$$

■ 例1.3-3: 八、十六进制转换为二进制举例。

$$(25.63)_8 = (\textcolor{red}{0}10, 101.110, \textcolor{red}{0}11)_2 = (10101.110011)_2$$

$$(\text{F}7.28)_{16} = (1111, \textcolor{red}{0}111.\textcolor{red}{00}10, \textcolor{red}{1000})_2 = (11110111.00101)_2$$

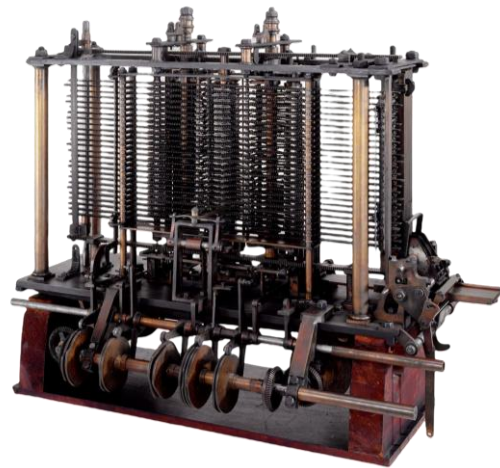
进制转换举例



■ 例1.3-4：将十进制数 $(194.6875)_{10}$ 转换为二进制形式。

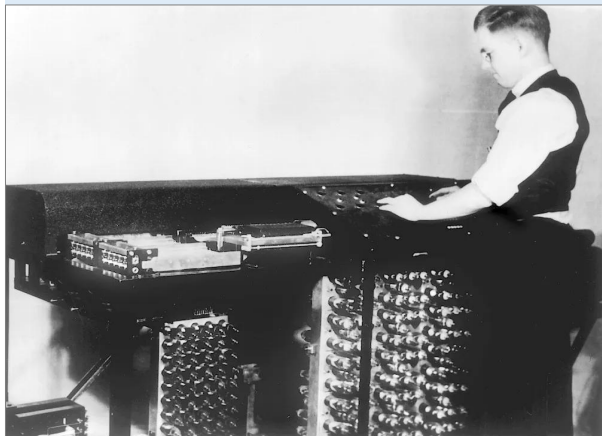
		余数		(小数点)		整数部分		
2	194	...	0	(低位)				0.6875
2	97	...	1			1	...	$\times \quad 2$
2	48	...	0					1.3750
2	24	...	0					0.3750
2	12	...	0			0	...	$\times \quad 2$
2	6	...	0					0.7500
2	3	...	1			1	...	$\times \quad 2$
2	1	...	1	(高位)				1.5000
0								0.5000
								$\times \quad 2$
								1.0000

- 计算与进制
- 计算机体系结构
- 信息与编码
- 文件与地址
- 算法与逻辑
- 计算思维



*Portion of the Calculating Machine of the
Analytical Engine, Built by Charles Babbage,
1834–1871*

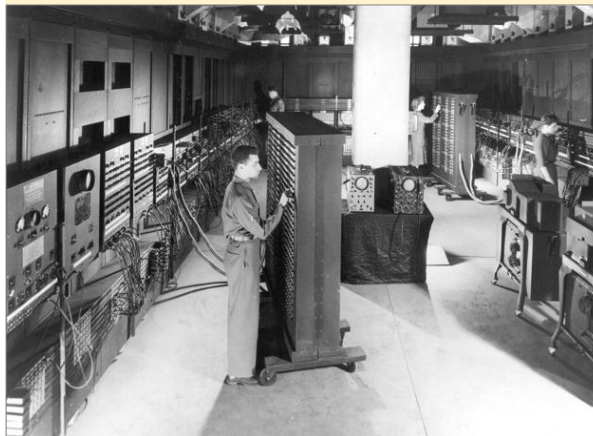
电子计算机的诞生和早期发展



第一台电子计算机

阿塔纳索夫—贝瑞计算机
Atanasoff-Berry Computer (ABC),
1942

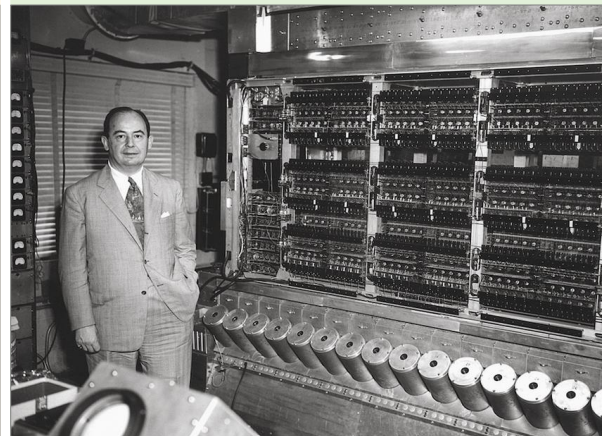
不可编程
仅设计用于求解线性方程组



第一台通用电子计算机

电子数值积分计算机
Electronic Numerical Integrator and
Computer (ENIAC), 1946

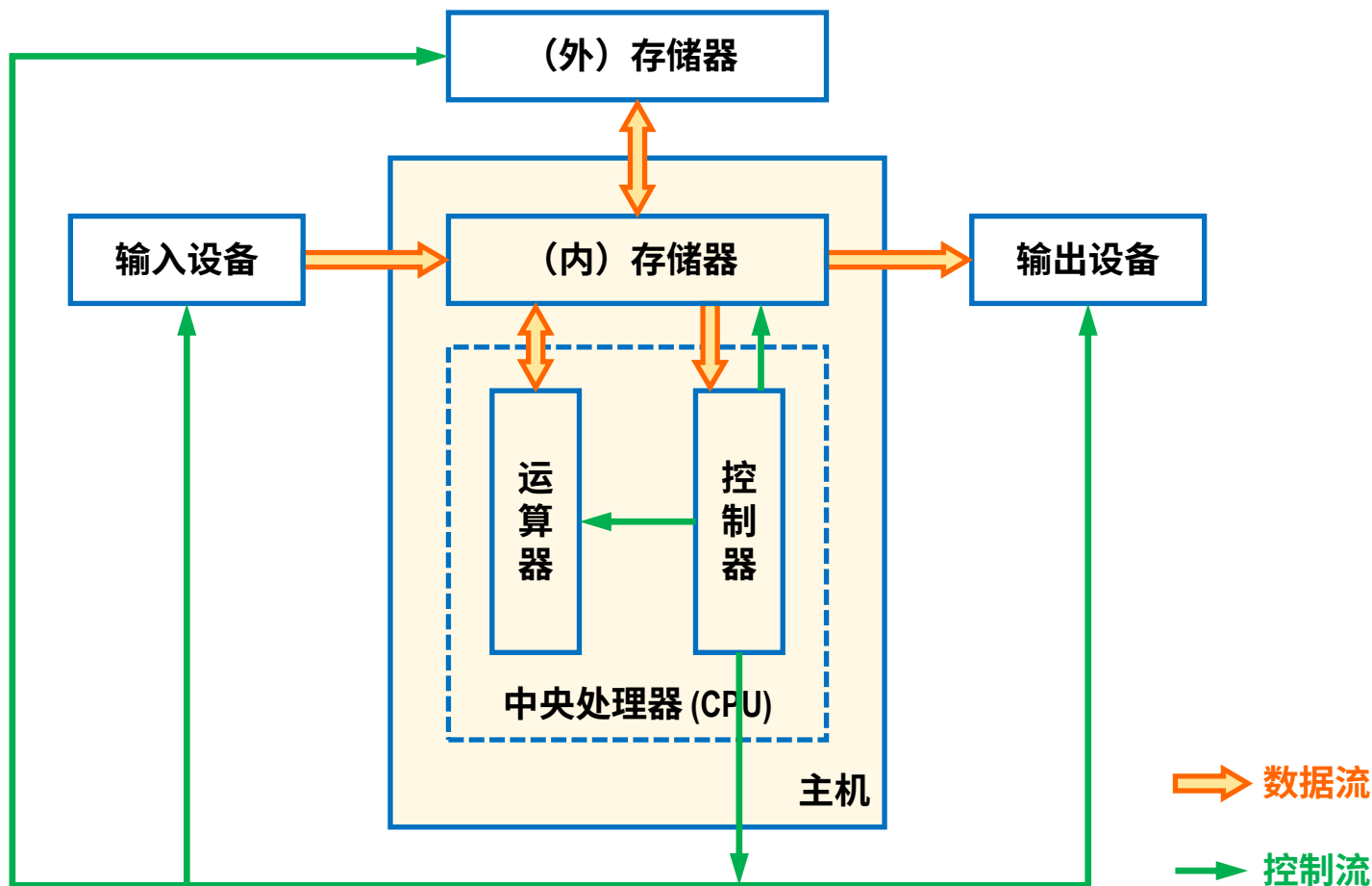
能够重新编程
解决各种计算问题



第一个冯·诺依曼结构计算机方案

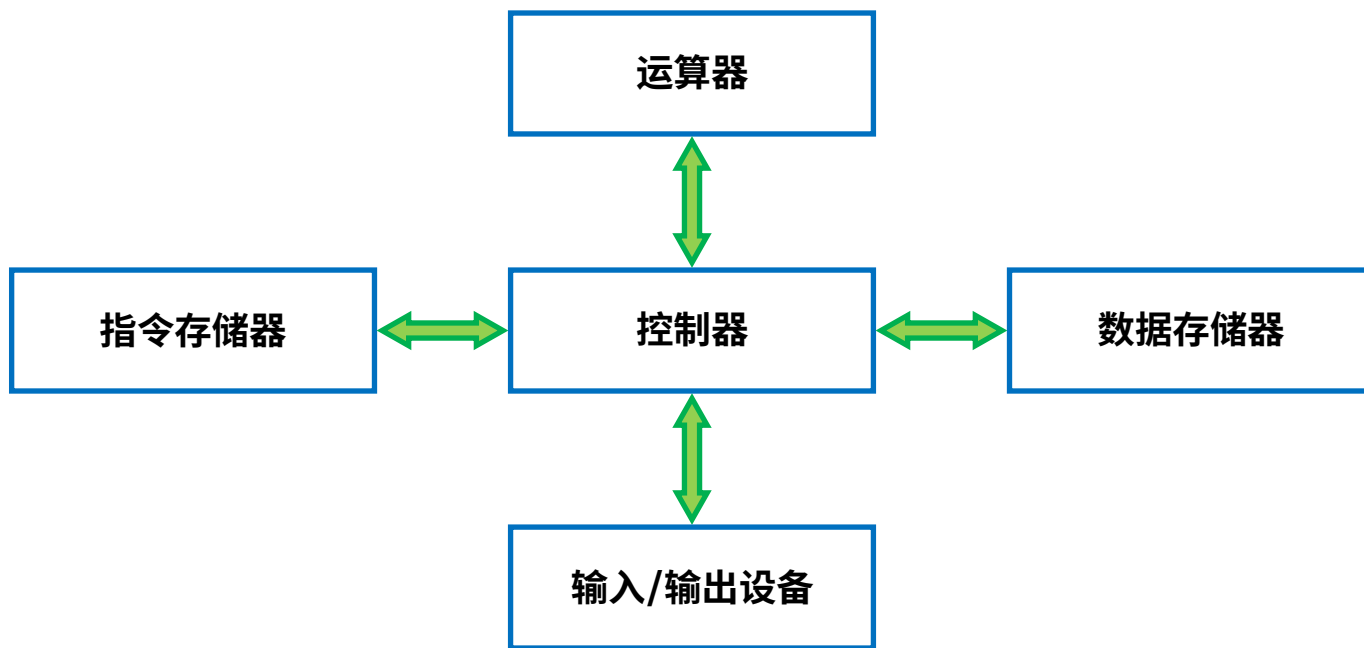
电子离散变量自动计算机
Electronic Discrete Variable Automatic
Computer (EDVAC), 1945–1951

引入了电子计算机制造
和程序设计的新思想



冯·诺依曼结构 (Von Neumann Architecture)、普林斯顿结构 (Princeton Architecture)

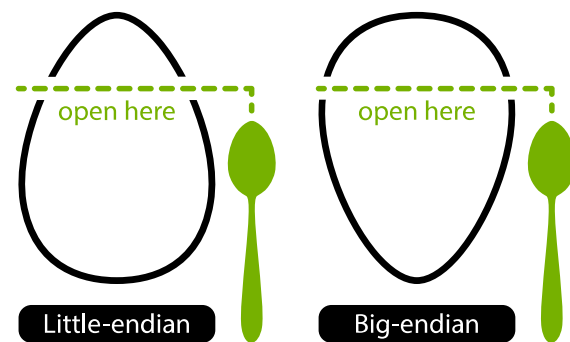
包含五个基本部件，二进制编码，存储程序原理



哈佛结构 (Harvard Architecture)

程序指令存储和数据存储分开，数据和指令的存取可以同时进行

- 计算与进制
- 计算机体系结构
- 信息与编码
- 文件与地址
- 算法与逻辑
- 计算思维



Little-Endian vs. Big-Endian

■ 信息 (Information)

- 现实世界与计算机世界的交互内容
- 信息的表现形式多不胜数，信息的种类不计其数

■ 编码 (Encoding)

- 将信息表示为二进制数据的过程
- 计算机本质上只能识别二进制数据

■ 信息与编码的类型

- 数字信息的编码：整数、实数
- 文字信息的编码：各种文字、字符、符号
- 其他信息的编码：声音、图像、视频、各种专用数据

信息的计量单位

■ 位、比特 (Bit, b)

- 二进制的一位，信息的最小单位
- 用于信息编码和信息传输，如音视频码率、网络传输速度

■ 字节 (Byte, B)

- 二进制的八位，1 byte = 8 bits
- 用作信息组织和存储的基本单位，如内存、硬盘、文件的大小

■ 二进制数量级前缀

- 千 (kilo-, k/K, kibi-, Ki) $1 \text{ Ki} = 2^{10} = 1024$
- 兆 (mega-, M, mebi-, Mi) $1 \text{ Mi} = 2^{20} = 1024 \text{ Ki} = 1,048,576$
- 吉 (giga-, G, gibib-, Gi) $1 \text{ Gi} = 2^{30} = 1024 \text{ Mi} = 1,073,741,824$
- 太 (tera-, T, tebi-, Ti) $1 \text{ Ti} = 2^{40} = 1024 \text{ Gi} = 1,099,511,627,776$
- 拍 (peta-, P, pebi-, Pi) $1 \text{ Pi} = 2^{50} = 1024 \text{ Ti} = 1,125,899,906,842,624$

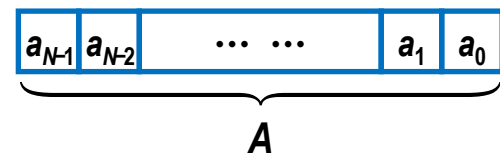
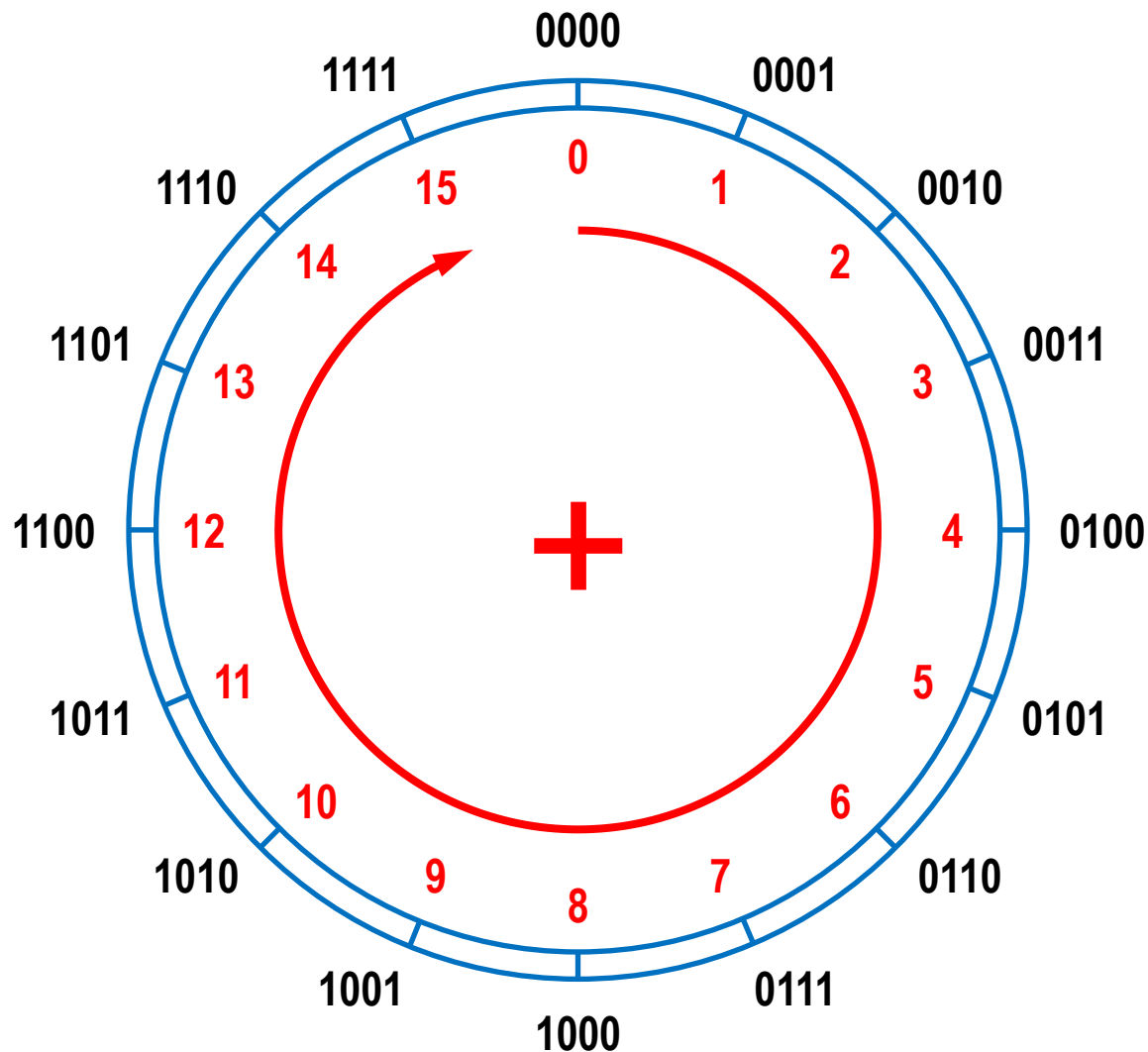
数字的编码

■ 整数

- 无符号数 (Unsigned Number)
 - 没有符号位，只能表示非负数
- 有符号数 (Signed Number)
 - 最高位用作符号位，可以表示正数、负数、零
 - 表示形式：原码、反码、补码、移码

■ 实数

- 定点数 (Fixed-Point Number)
 - 小数点隐含在某个固定位置，表示为整数和隐含的小数长度
- 浮点数 (Floating-Point Number)
 - 小数点位置可以浮动，表示为有效数字乘以基数的整数次幂



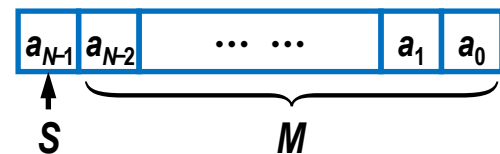
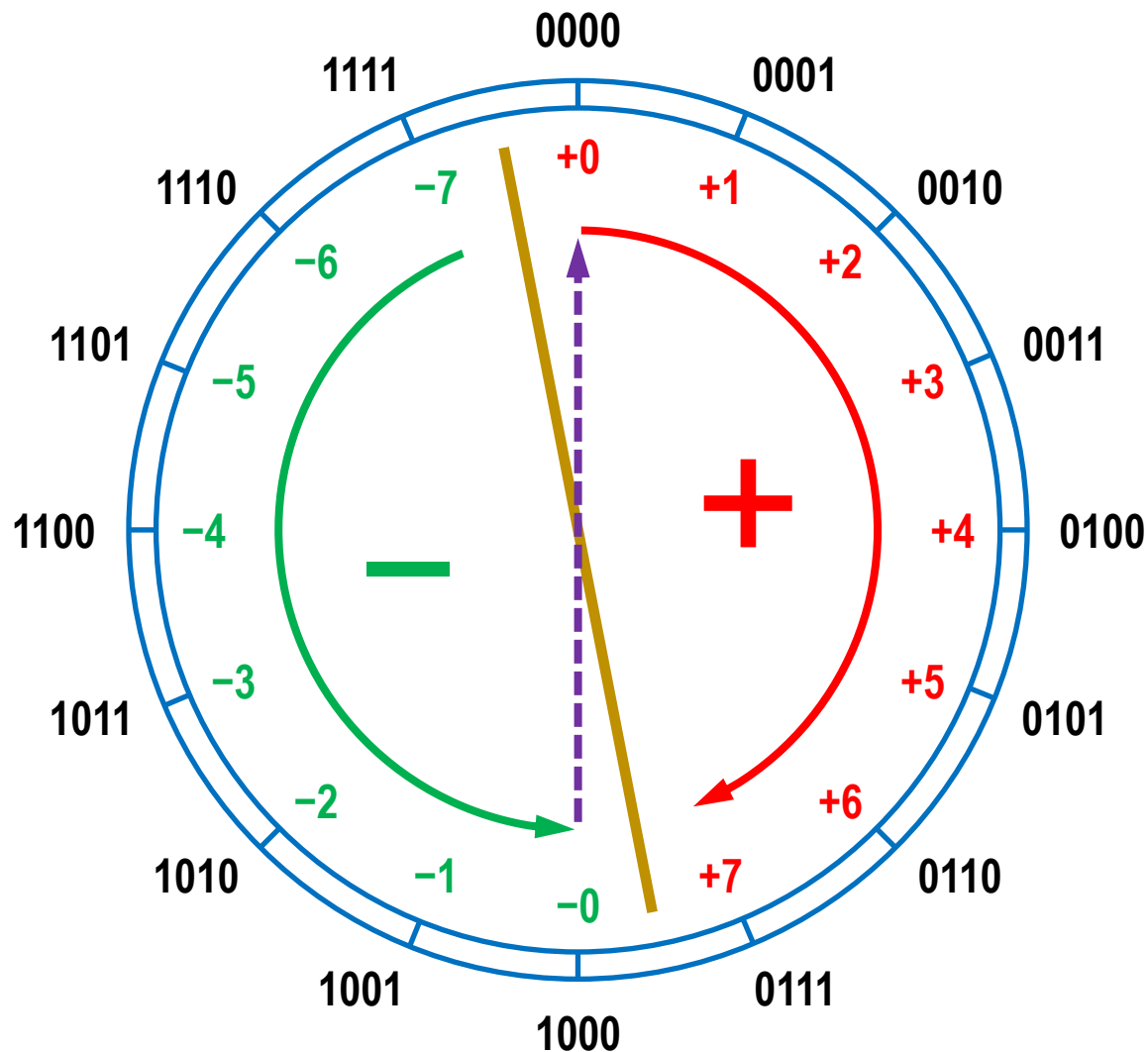
$$A = \sum_{i=0}^{N-1} a_i 2^i$$

表达范围:

$$0 \sim 2^N - 1$$

无符号数的编码

有符号数的原码



$$A = (-1)^S \cdot M$$

表达范围:

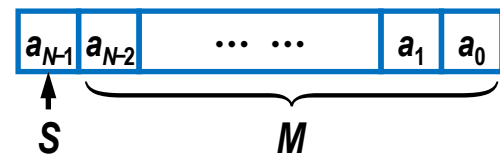
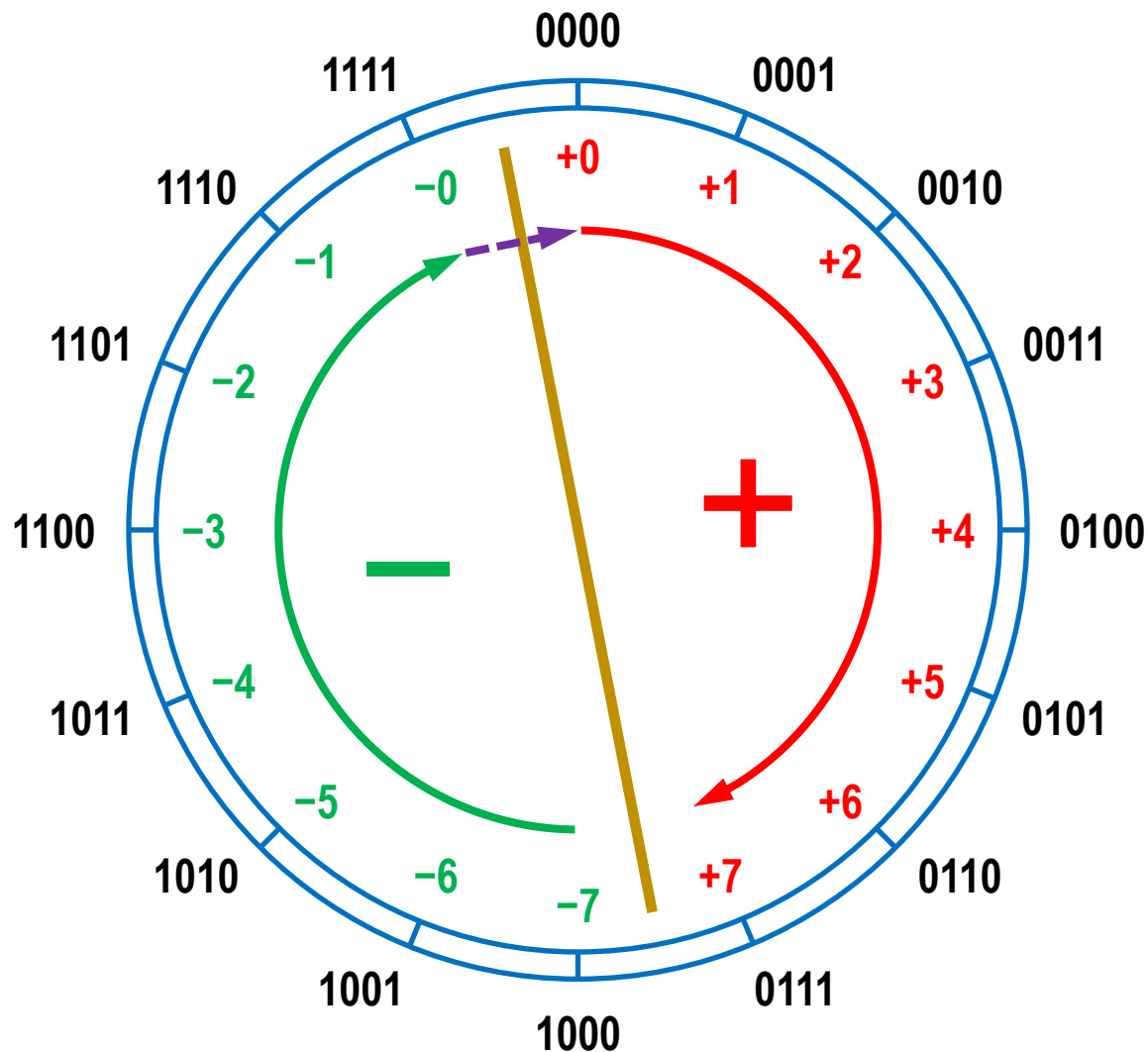
$$-(2^{N-1} - 1) \sim 2^{N-1} - 1$$

正数: 符号位为“0”

负数: 符号位为“1”

有符号数的原码
(Signed Magnitude)

有符号数的反码



$$A = -(2^{N-1} - 1) \cdot S + M$$

表达范围:

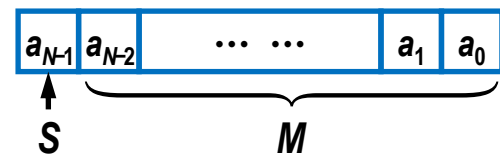
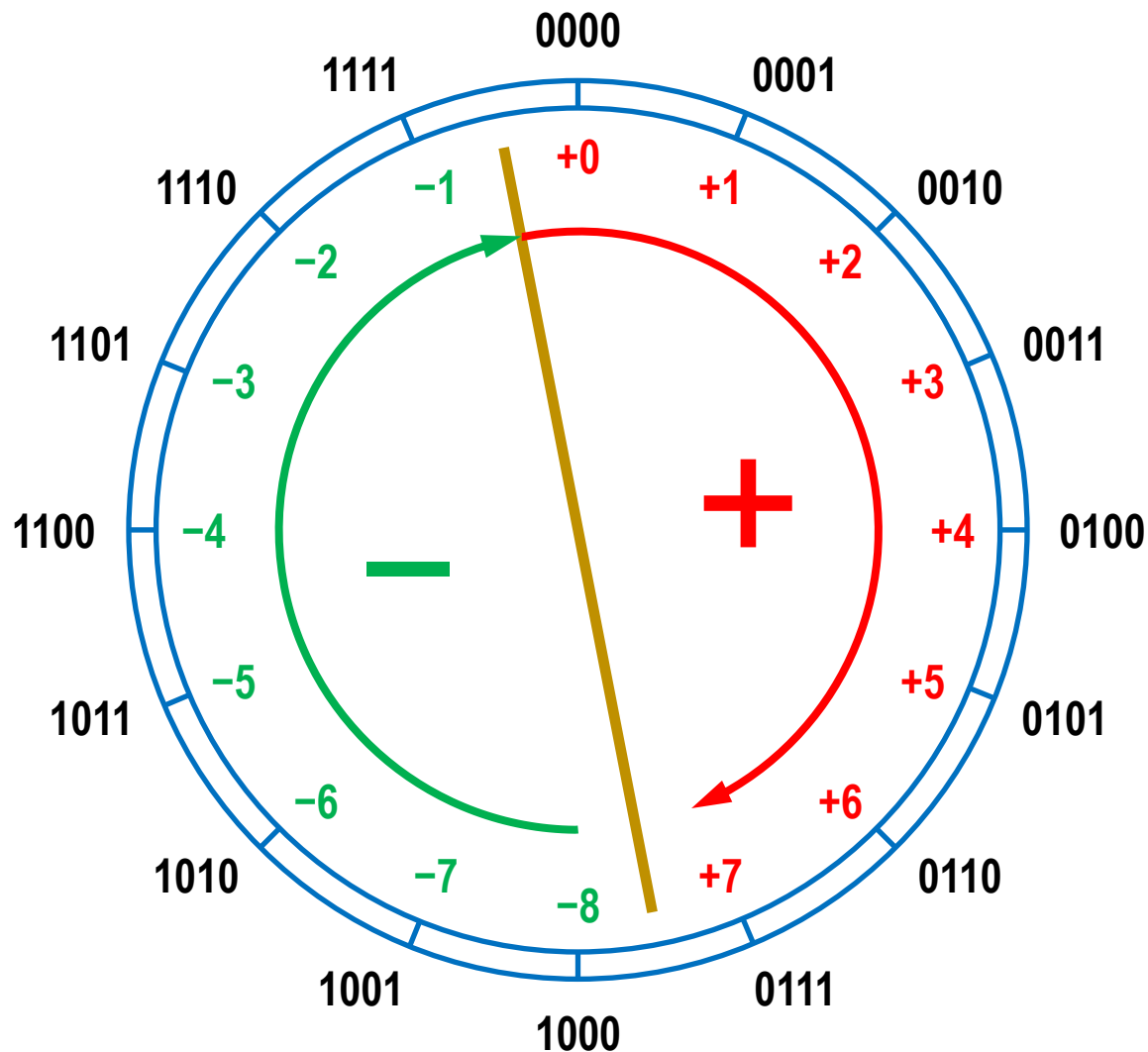
$$-(2^{N-1} - 1) \sim 2^{N-1} - 1$$

非负数: 同原码

相反数: 按位取反

有符号数的反码
(Ones' Complement)

有符号数的补码



$$A = -2^{N-1} \cdot S + M$$

表达范围:

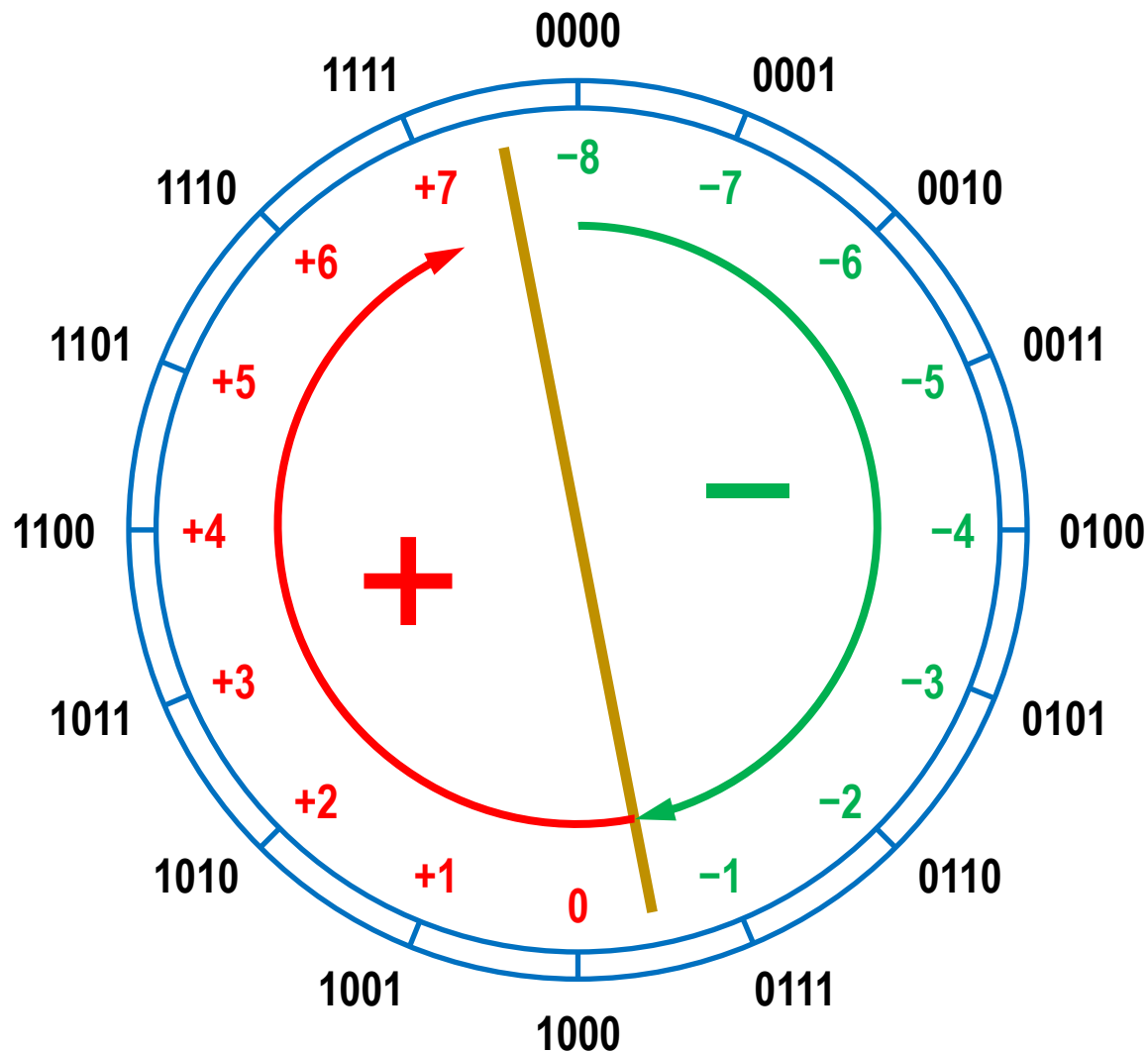
$$-2^{N-1} \sim 2^{N-1} - 1$$

非负数: 同原码

相反数: 按位取反加一

有符号数的补码
(Two's Complement)

有符号数的移码



$$\underbrace{a_{N-1} a_{N-2} \dots a_1 a_0}_{A+2^{N-1}}$$

$$A = \sum_{i=0}^{N-1} a_i 2^i - 2^{N-1}$$

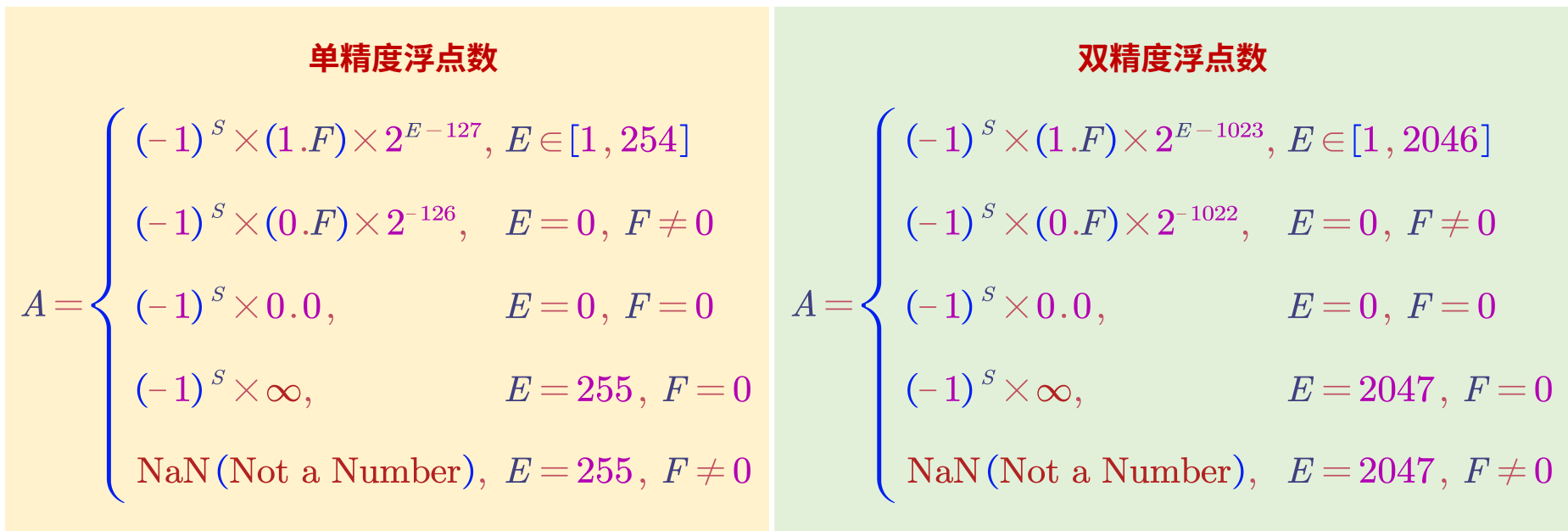
表达范围:

$$-2^{N-1} \sim 2^{N-1} - 1$$

标准移码:

补码最高位取反

有符号数的标准移码
(Offset Binary, Excess- 2^{N-1})



32

■ 西文字符编码

- ASCII (American Standard Code for Information Interchange)
- 标准ASCII码：七位二进制数，取值范围0~127，占用一字节，最高位为“0”
- 扩展ASCII码：八位二进制数，取值范围0~255，占用一字节

■ 汉字编码

- GB 2312-1980：双字节编码，共收录6,763个汉字、682个外语字符和图符
- GB 18030-2022：变长多字节编码，共收录汉字87,887个，支持少数民族文字

■ 统一码 (Unicode)

- <https://www.unicode.org>
- 提供单一、综合的字符集，编码一切现代与大部分历史文献的字符
- 最新版本收录159,801个字符 (Unicode 17.0.0, 2025.09.09)

ASCII码表



十六进制	符号	十六进制	符号	十六进制	符号	十六进制	符号	十六进制	符号	十六进制	符号	十六进制	符号	十六进制	符号
00	(NUL)	10	(DLE)	20	(SPACE)	30	0	40	@	50	P	60	`	70	p
01	(SOH)	11	(DC1)	21	!	31	1	41	A	51	Q	61	a	71	q
02	(STX)	12	(DC2)	22	"	32	2	42	B	52	R	62	b	72	r
03	(ETX)	13	(DC3)	23	#	33	3	43	C	53	S	63	c	73	s
04	(EOT)	14	(DC4)	24	\$	34	4	44	D	54	T	64	d	74	t
05	(ENQ)	15	(NAK)	25	%	35	5	45	E	55	U	65	e	75	u
06	(ACK)	16	(SYN)	26	&	36	6	46	F	56	V	66	f	76	v
07	(BEL)	17	(ETB)	27	'	37	7	47	G	57	W	67	g	77	w
08	(BS)	18	(CAN)	28	(38	8	48	H	58	X	68	h	78	x
09	(HT)	19	(EM)	29)	39	9	49	I	59	Y	69	i	79	y
0A	(LF)	1A	(SUB)	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	(VT)	1B	(ESC)	2B	+	3B	;	4B	K	5B	[6B	k	7B	{
0C	(FF)	1C	(FS)	2C	,	3C	<	4C	L	5C	\	6C	l	7C	
0D	(CR)	1D	(GS)	2D	-	3D	=	4D	M	5D]	6D	m	7D	}
0E	(SO)	1E	(RS)	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	(SI)	1F	(US)	2F	/	3F	?	4F	O	5F	_	6F	o	7F	(DEL)

- [illegible]

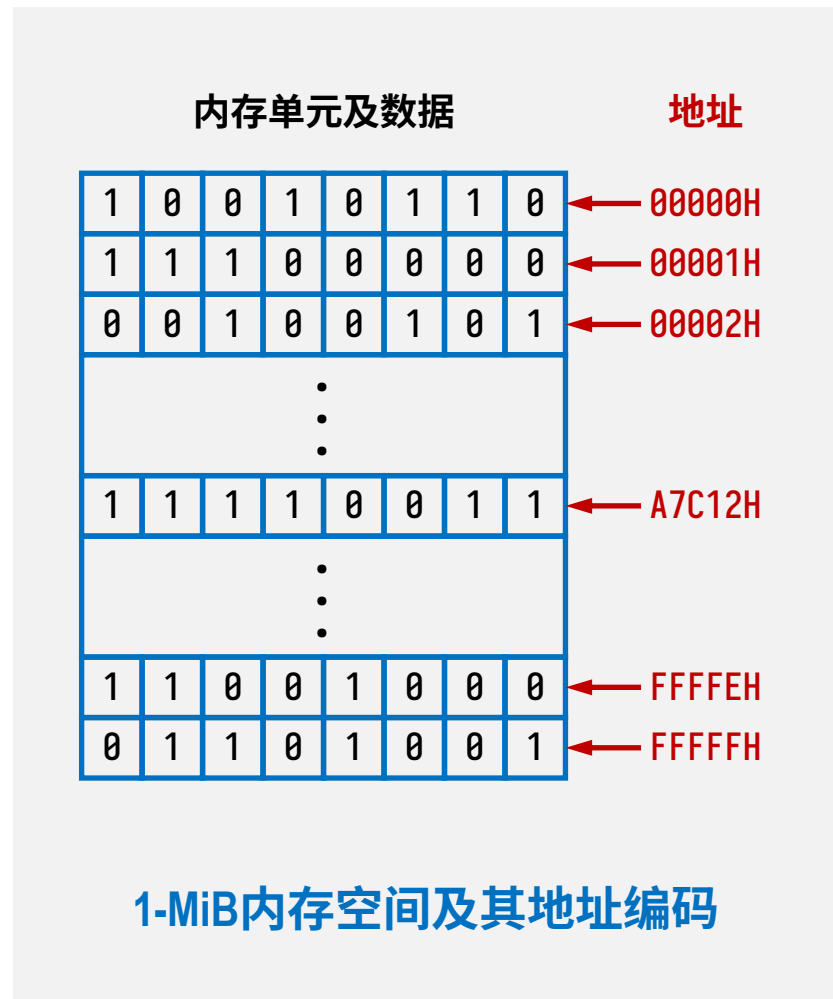
35

■ 文件 (File)

- 存储管理外存储器中信息的单位
- 文本文件、二进制文件
- 命名方式: 文件名.扩展名
- 扩展名一般用于表示文件类型

■ 地址 (Address)

- 存储的计量单位通常是字节
- 存储器中每个字节分配一个地址
- 用于定位存储器中存储的内容
- 地址使用二进制编码
- 地址也是一种数据类型



- 计算与进制
- 计算机体系结构
- 信息与编码
- 文件与地址
- 算法与逻辑
- 计算思维



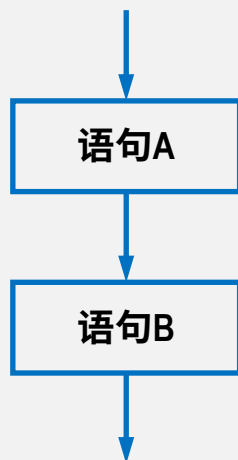
Königsberg Seven Bridge Problem

■ 算法 (Algorithm)

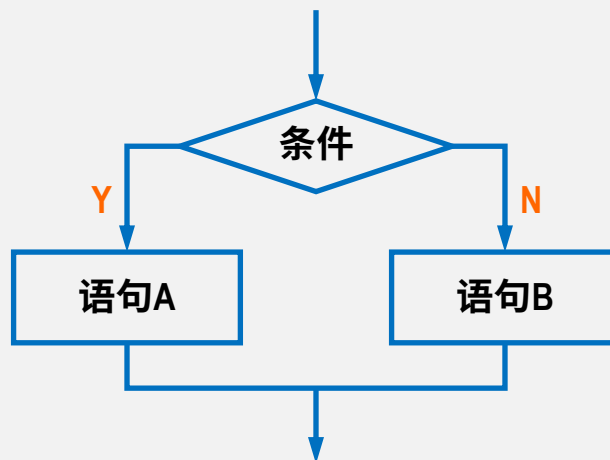
- 解题方案的准确而完整的描述
- 用于解决问题的有限指令序列
- 算法 + 数据结构 = 程序

■ 逻辑 (Logic)

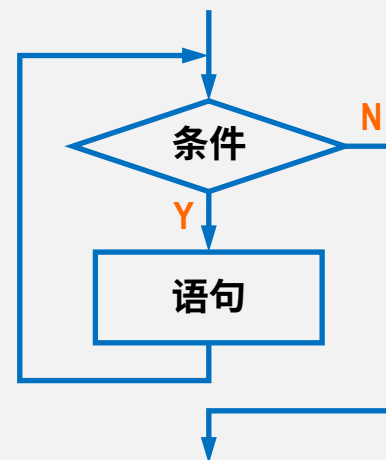
- 顺序结构
- 选择结构（分支结构）
- 重复结构（循环结构）



顺序结构



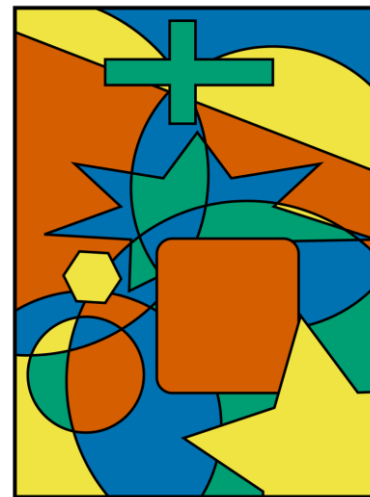
分支结构



循环结构

基本逻辑结构

- 计算与进制
- 计算机体系结构
- 信息与编码
- 文件与地址
- 算法与逻辑
- 计算思维



*Four Color Theorem, Proved
Using a Computer, 1976*

■ 计算思维的定义

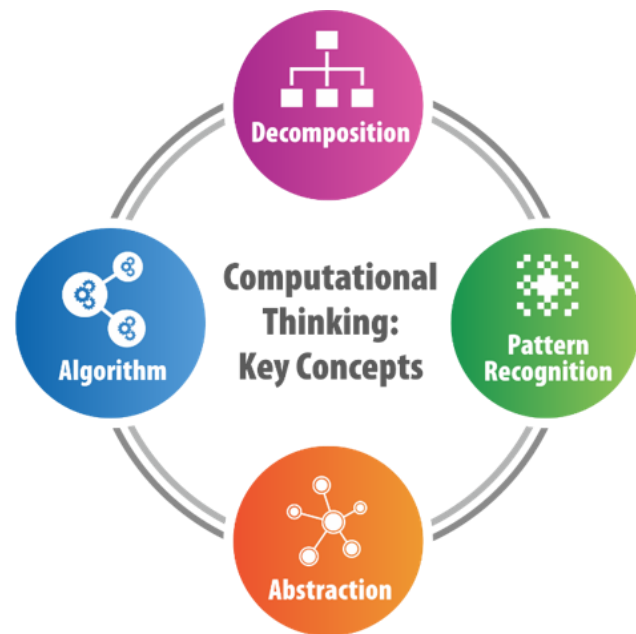
- 运用计算机科学的基础概念进行问题求解、系统设计、人类行为理解的一系列思维活动

■ 计算思维的核心特征

- 抽象 (Abstraction): 问题表示和算法设计
- 自动化 (Automation): 根据算法完成计算任务

■ 计算思维的过程

- 分解问题、模式识别、抽象化、算法实现
- 建模、映射、编程、求解



计算思维
(Computational Thinking, CT)



编程语言与程序设计

- 低级语言
- 高级语言
- 编程语言分类
- 编程方法
- 操作系统与程序
- C语言程序设计



*Ada Lovelace (1815–1852),
the First Computer Programmer?*

■ 机器语言

- 机器指令的集合
- 指令根据计算机硬件电路设计，与机器硬件相关联

■ 汇编语言

- 使用容易理解记忆的字母和单词代替机器语言的二进制序列

■ 低级语言的特点

- 依赖于硬件
- 执行效率高
- 编程效率低
- 难以移植到不同的硬件系统上

机器语言

B8 20 4E
05 16 14
BB 00 20
01 D8
89 C3
01 D8
B8 1A 00
BB 26 00
00 D8
00 DC
00 C7
B4 00
00 D8
04 9C

汇编语言

MOV AX, 4E20H
ADD AX, 1416H
MOV BX, 2000H
ADD AX, BX
MOV BX, AX
ADD AX, BX
MOV AX, 001AH
MOV BX, 0026H
ADD AL, BL
ADD AH, BL
ADD BH, AL
MOV AH, 0
ADD AL, BL
ADD AL, 9CH

机器语言与汇编语言

- 低级语言
- 高级语言
- 编程语言分类
- 编程方法
- 操作系统与程序
- C语言程序设计

■ 高级语言的特点

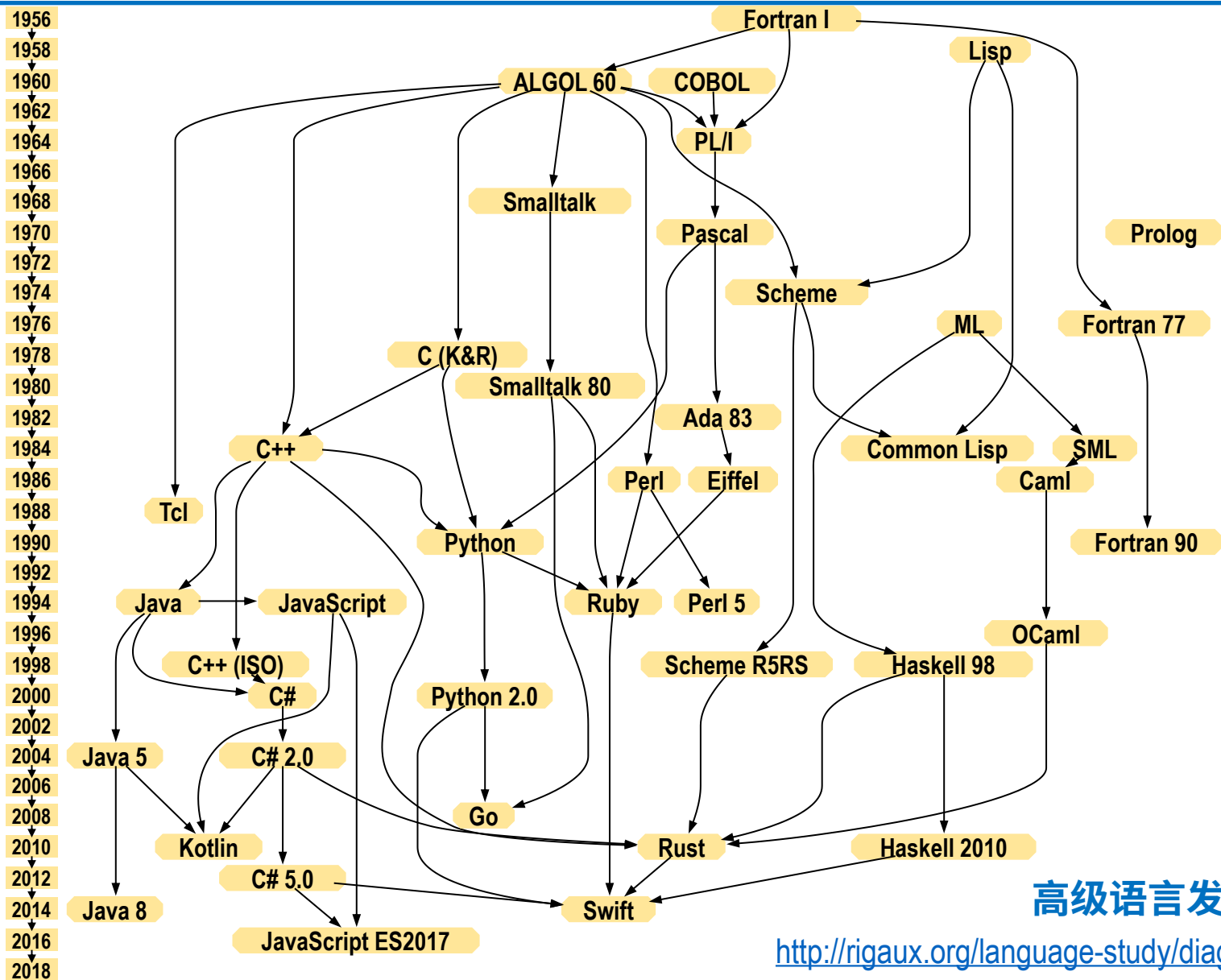
- 屏蔽与硬件相关的底层实现细节，基本脱离机器的硬件系统
- 语句较接近自然语言和数学公式
- 可移植性、易理解性、编程效率显著提高
- 执行效率有所降低

■ 常见高级语言

- C 面向过程，广泛用于系统软件与应用软件开发
- C++ 面向对象，派生自C语言，但并不完全兼容C语言
- C# 面向对象，派生自C和C++，并去除部分复杂特性
- Java 跨平台、面向对象，广泛用于网页应用和移动应用开发
- Python 跨平台、解释型语言，应用广泛，包括网页应用、数据科学等
- Pascal 语法严谨，适合用于结构化编程初学者的教学
- Fortran 第一个通用性高级语言，主要用于数值计算

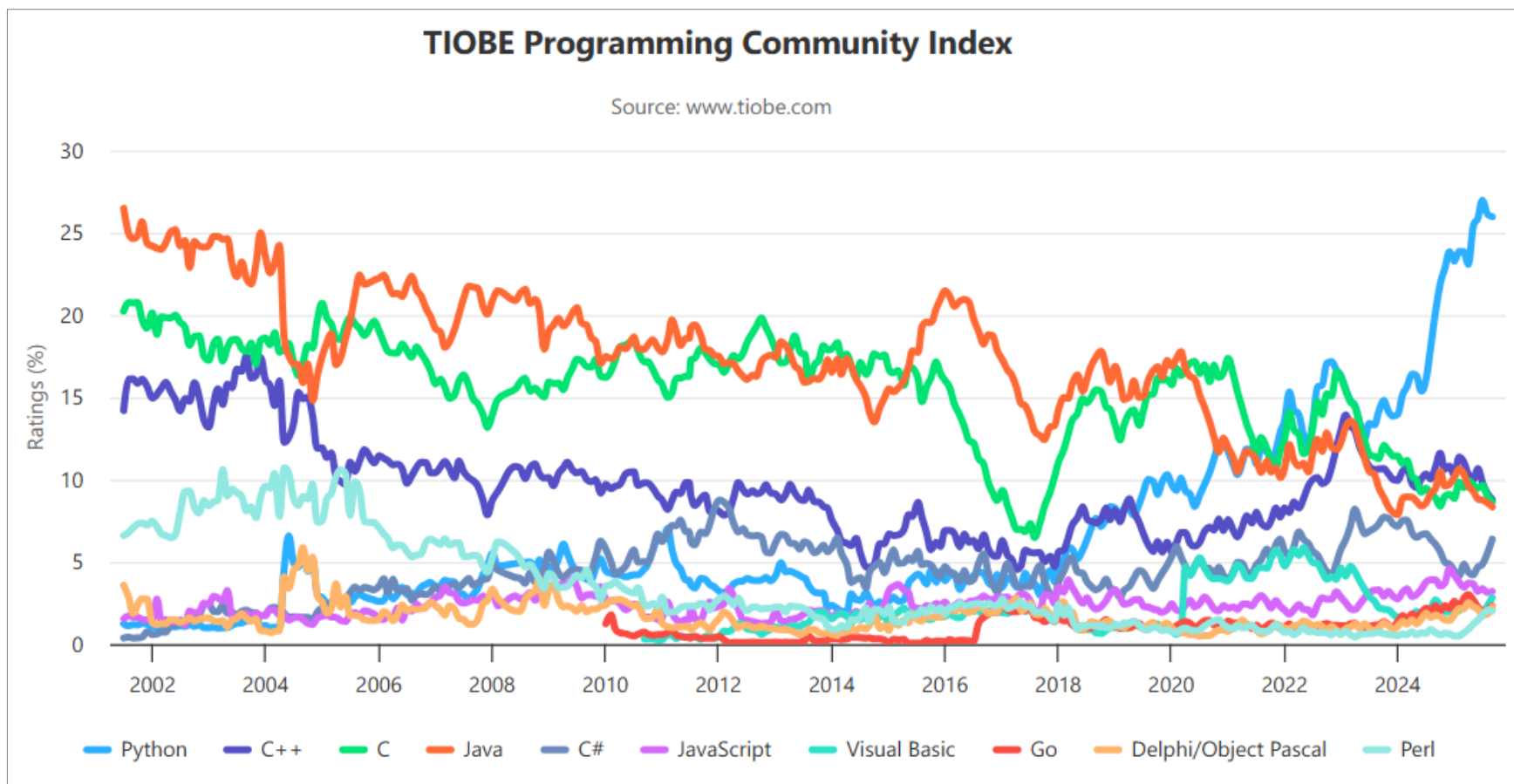
Go

高级语言发展简史



高级语言发展简史

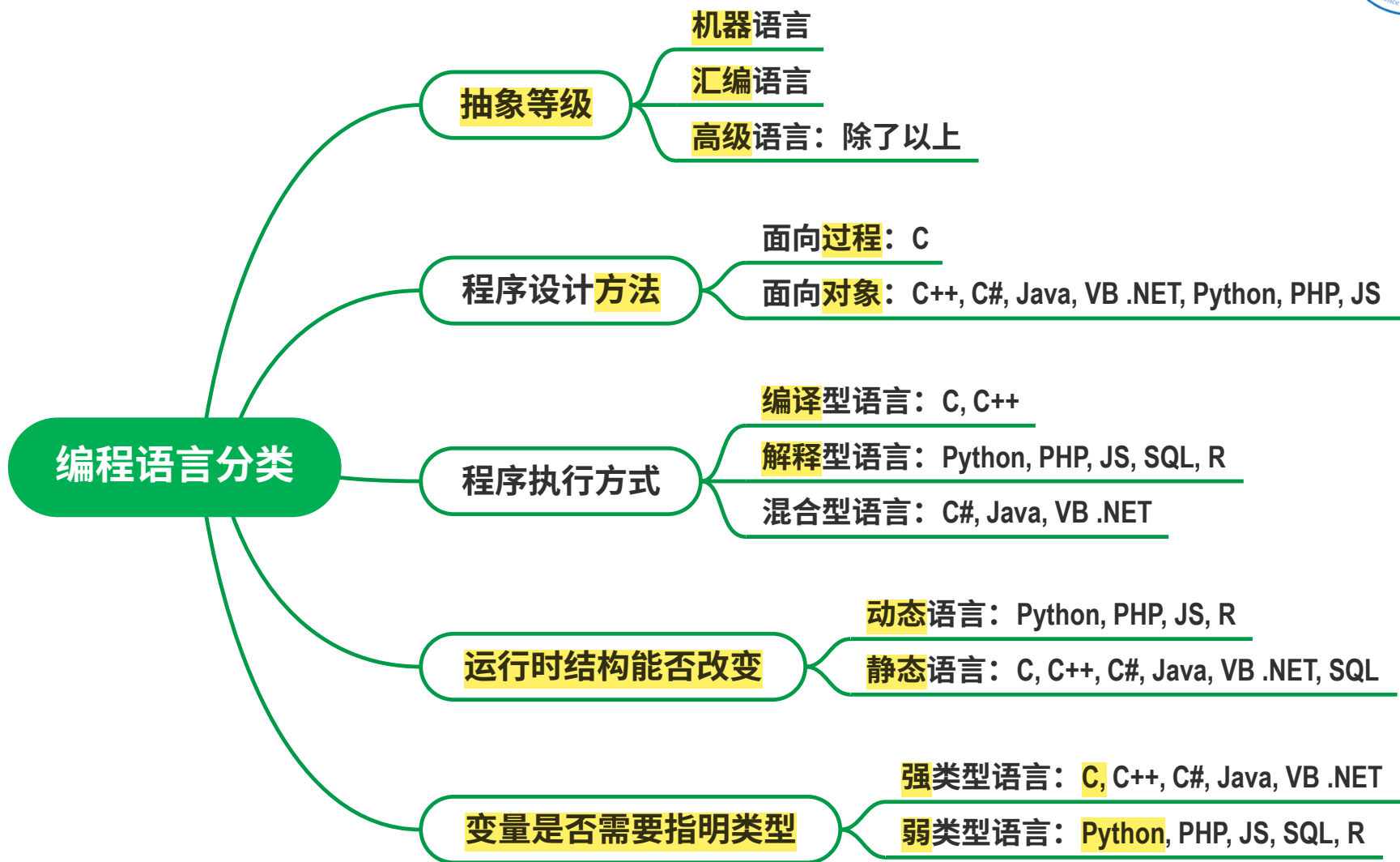
<http://rigaux.org/language-study/diagram.html>



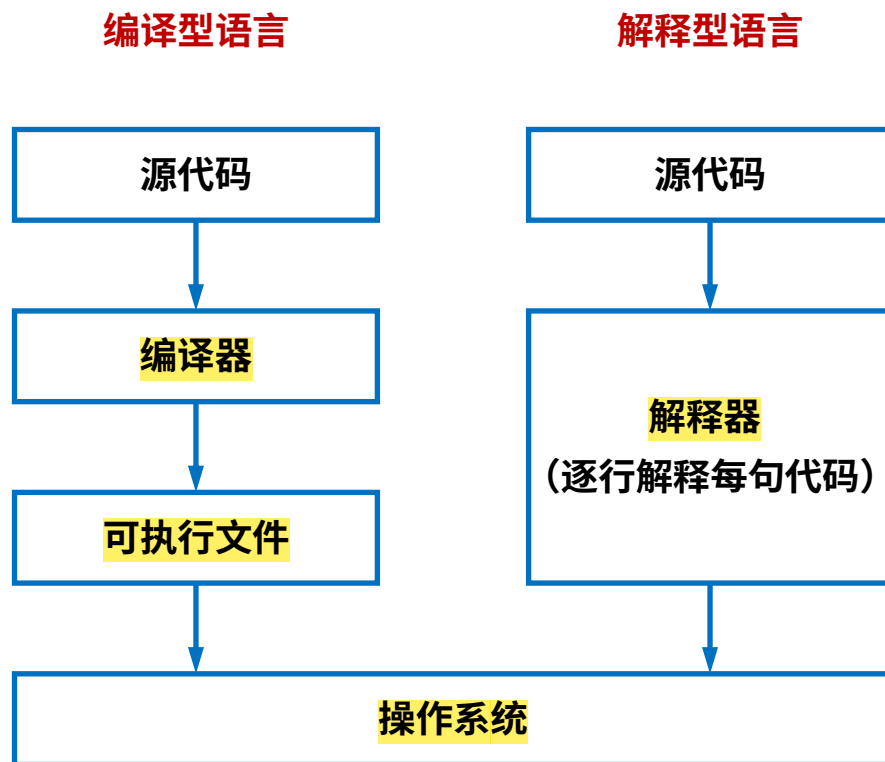
TIOBE编程语言流行榜前十名及其走势图 (2025.09)

<https://www.tiobe.com/tiobe-index>

- 低级语言
- 高级语言
- 编程语言分类
- 编程方法
- 操作系统与程序
- C语言程序设计

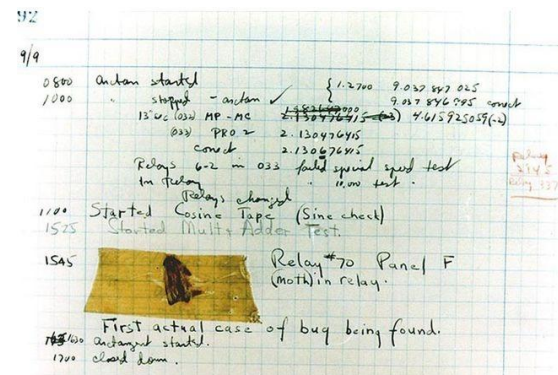


主要编程语言的简单分类



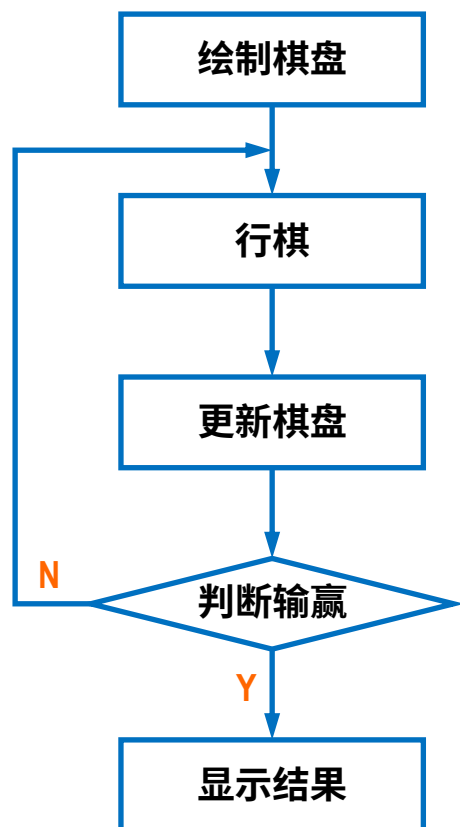
编译型语言与解释型语言

- 低级语言
- 高级语言
- 编程语言分类
- 编程方法
- 操作系统与程序
- C语言程序设计

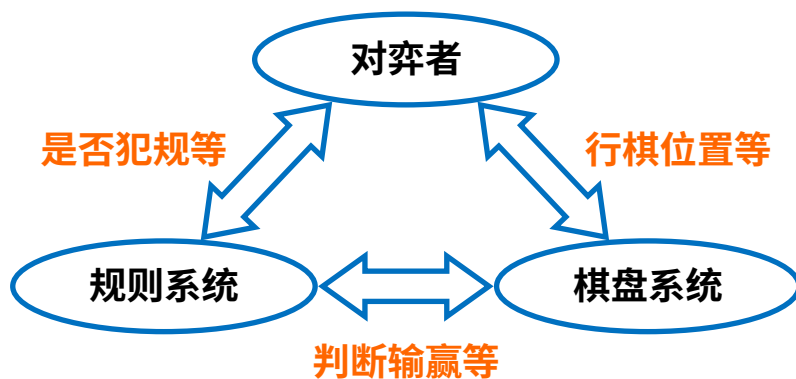


The First Computer “Bug”, 1947

面向过程



面向对象



面向过程和面向对象的对比

面向过程：

围绕解决问题的过程，以事件为中心

面向对象：

围绕问题中的对象以及对象间的关系，以对象为中心

- 低级语言
- 高级语言
- 编程语言分类
- 编程方法
- 操作系统与程序
- C语言程序设计



Some of Linux Distributions

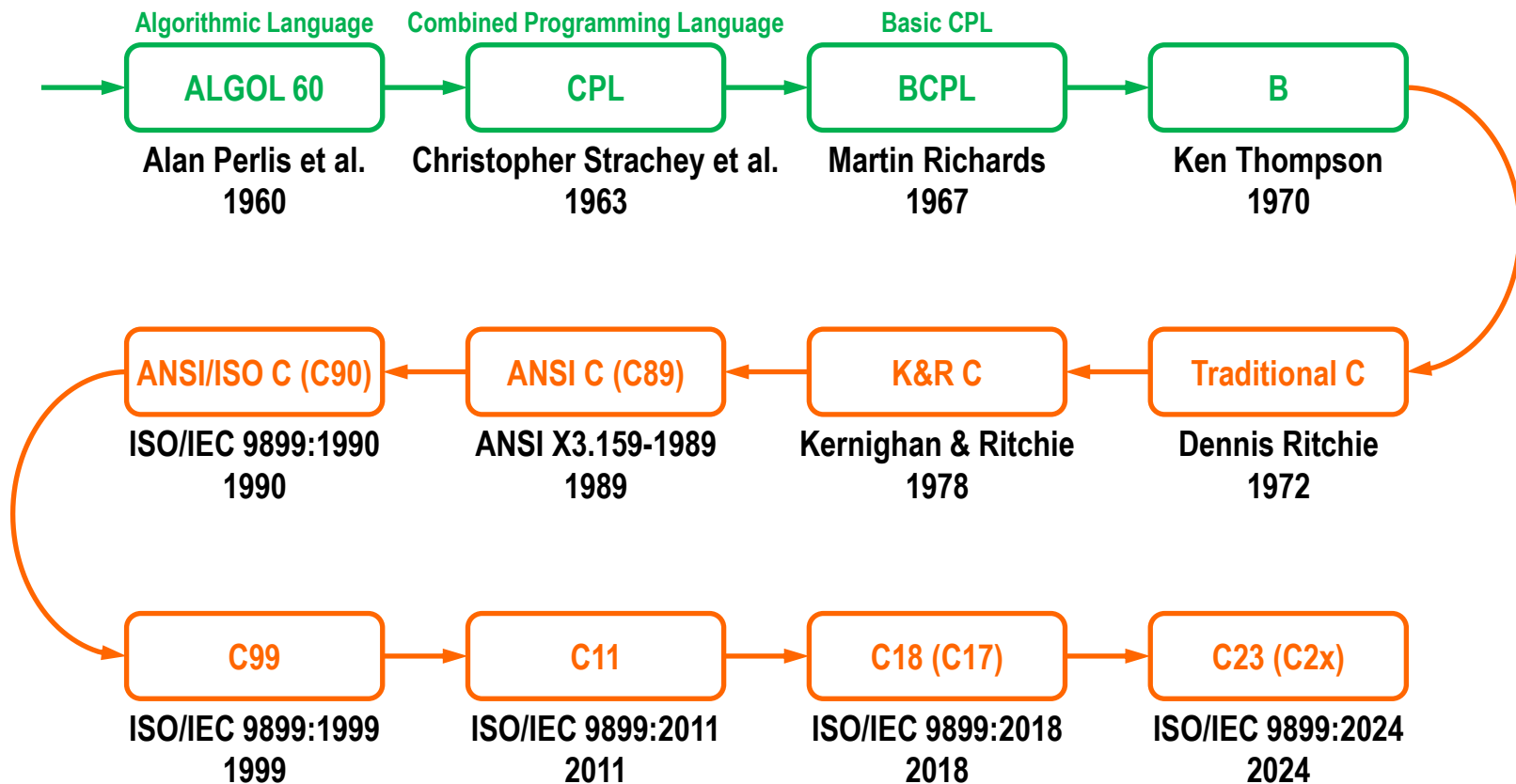
■ 操作系统

- 管理和控制计算机软硬件资源的软件系统
- 操作系统内核：与CPU、内存等核心部件打交道的部分
- 开发编程语言：汇编语言、C语言、C++语言
- 主要操作系统：Windows, Linux, Unix, MacOS, iOS, Android, ...

■ 应用程序

- 操作系统以外的可执行程序
- 应用程序必须在操作系统的支持下才能工作
- 操作系统负责将应用程序加载到内存中运行

- 低级语言
- 高级语言
- 编程语言分类
- 编程方法
- 操作系统与程序
- C语言程序设计



C语言发展简史时间线

■ C语言的特点

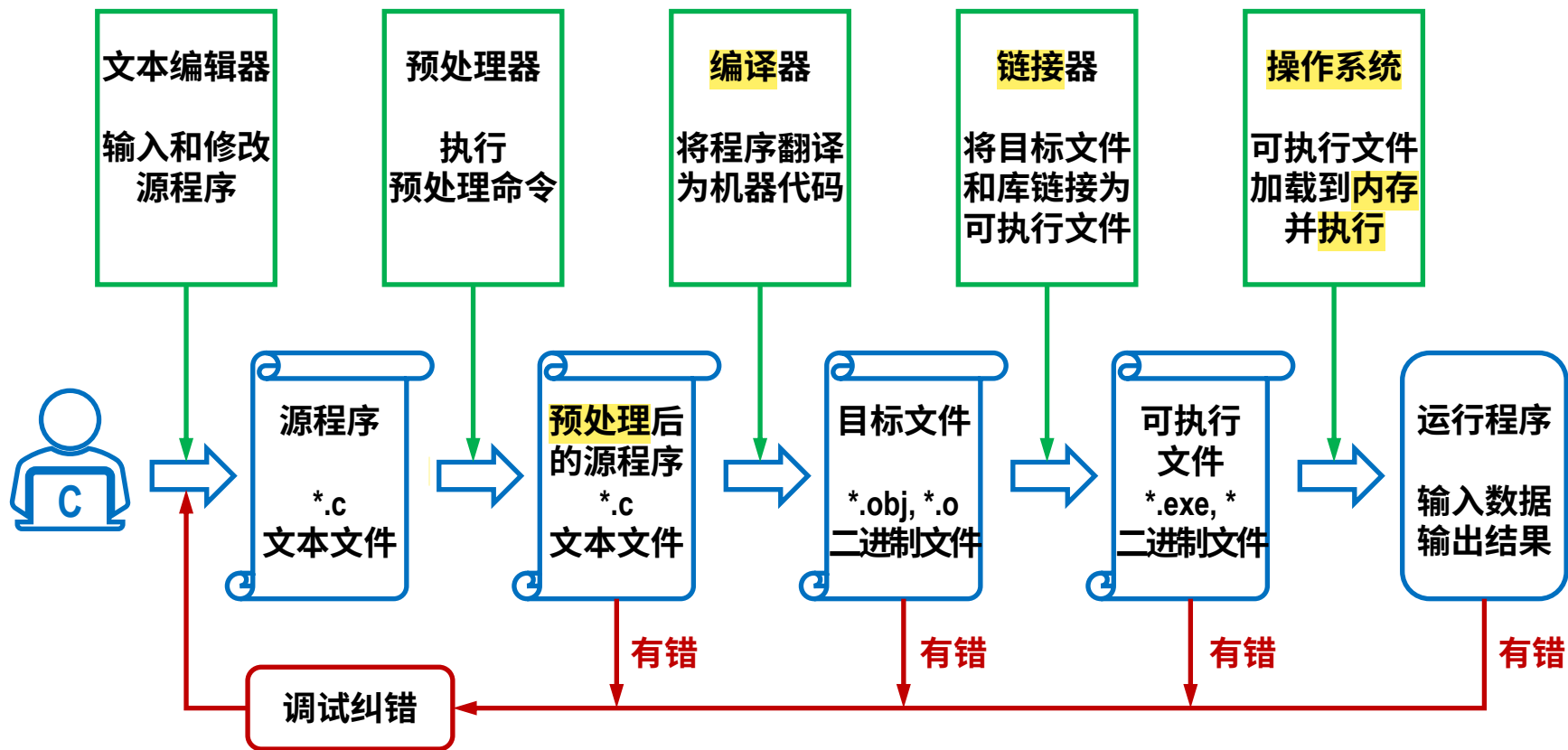
- 概念清楚，语法简洁精炼，表达简单
- 描述能力强，机制灵活，支持结构化程序设计
- 与操作系统结合紧密，生成代码的执行效率高
- 兼具高级语言与低级语言的特点，适用领域宽广

■ 常用C语言编译器

- GCC, MinGW-GCC, Intel C++, Clang, Microsoft Visual C++, et al.

■ 常用集成开发环境 (IDE)

- Code::Blocks, Dev-C++, Visual Studio, Visual Studio Code, Xcode, Code Lite, et al.



C语言程序开发步骤和流程

小 结

■ 主要知识点

- 进制转换的方法：除基取余法、乘基取整法
- 数字的编码方式：无符号数编码、有符号数补码、浮点数编码
- 字符的编码方式：ASCII码
- 计算思维的核心特征：抽象、自动化
- 程序设计的核心内容：建模、算法设计
- C语言程序设计的主要步骤：编写代码、编译、链接
- 包括操作系统在内的所有程序都在内存中运行

■ 能力要求

- 了解计算、信息、数据、指令、程序、内存、地址、抽象、自动化等概念以及它们之间的关联
- 了解主要程序设计语言的层次与应用场景
- 能够针对学习中的疑问自行查找答案，并通过多个信源的对比分析辨析正误



本章结束