第 1 章 绪论

郑 莉 清华大学

教材：C++语言程序设计（第5版） 郑莉 清华大学出版社

目录

计算机程序

- 计算机的工作是用程序来控制的
- 程序是指令的集合。
- 指令是计算机可以识别的命令。

机器语言与汇编语言

- 由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。
  - 计算机发展的初期，软件工程师们只能用机器语言来编写程序。这一阶段，在人类的自然语言和计算机编程语言之间存在着巨大的鸿沟。
- 汇编语言将机器指令映射为一些可以被人读懂的助记符，如ADD、SUB等。
  - 此时编程语言与人类自然语言间的鸿沟略有缩小，但仍与人类的思维相差甚远。因为它的抽象层次太低，程序员需要考虑大量的机器细节。

高级语言

- 高级语言屏蔽了机器的细节，提高了语言的抽象层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这使得在书写程序时可以联系到程序所描述的具体事物。

面向对象的语言

- 出发点：
  - 更直接地描述客观世界中存在的事物(对象)以及它们之间的关系。
- 特点：
  - 是高级语言。
  - 将客观事物看作具有属性和行为的对象。
  - 通过抽象找出同一类对象的共同属性和行为，形成类。
  - 通过类的继承与多态实现代码重用
- 优点：
  - 使程序能够比较直接地反映问题域的本来面目，软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

最早的程序

- 目的：用于数学计算
- 主要工作：设计求解问题的过程
- 缺点：对于庞大、复杂的程序难以开发和维护

面向过程的结构化程序设计方法

- 设计思路
  - 自顶向下、逐步求精。采用模块分解与功能抽象，自顶向下、分而治之。
- 程序结构：
  - 按功能划分为若干个基本模块。
  - 各模块间的关系尽可能简单，功能上相对独立；每一模块内部均是由顺序、选择和循环三种基本结构组成。
  - 其模块化实现的具体方法是使用子程序。
- 优点：
  - 有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务，便于开发和维护。

面向过程的结构化程序设计方法（续）

- 缺点：可重用性差、数据安全性差、难以开发大型软件和图形界面的应用软件
  - 把数据和处理数据的过程分离为相互独立的实体。
  - 当数据结构改变时，所有相关的处理过程都要进行相应的修改。
  - 每一种相对于老问题的新方法都要带来额外的开销。
  - 图形用户界面的应用程序，很难用过程来描述和实现，开发和维护也都很困难。

面向对象的方法

- 将数据及对数据的操作方法封装在一起，作为一个相互依存、不可分离的整体——对象。
- 对同类型对象抽象出其共性，形成类。
- 类通过一个简单的外部接口，与外界发生关系。
- 对象与对象之间通过消息进行通信。
- 优点：
  - 程序模块间的关系更为简单，程序模块的独立性、数据的安全性就有了良好的保障。
  - 通过继承与多态性，可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

面向对象的基本概念——对象

- 一般意义上的对象：
  - 是现实世界中一个实际存在的事物。
  - 可以是有形的（比如一辆汽车），也可以是无形的（比如一项计划）。
  - 是构成世界的一个独立单位，具有
    - 静态特征：可以用某种数据来描述
    - 动态特征：对象所表现的行为或具有的功能
- 面向对象方法中的对象：
  - 是系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。
  - 属性：用来描述对象静态特征的数据项。
  - 行为：用来描述对象动态特征的操作序列。

类

- 分类——人类通常的思维方法
- 分类所依据的原则——抽象
  - 忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽象的概念。
  - 例如，石头、树木、汽车、房屋等都是人们在长期的生产和生活实践中抽象出的概念。
- 面向对象方法中的"类"
  - 具有相同属性和服务的一组对象的集合
  - 为属于该类的全部对象提供了抽象的描述，包括属性和行为两个主要部分。
  - 类与对象的关系：
    犹如模具与铸件之间的关系，一个属于某类的对象称为该类的一个实例。

封装

- 把对象的属性和服务结合成一个独立的系统单元。
- 尽可能隐蔽对象的内部细节。对外形成一个边界（或者说一道屏障），只保留有限的对外接口使之与外部发生联系。

继承

- 继承对于软件复用有着重要意义，是面向对象技术能够提高软件开发效率的重要原因之一。
- 定义：特殊类的对象拥有其一般类的全部属性与服务，称作特殊类对一般类的继承。
- 例如：将图形作为一个一般类，圆形便是一个特殊类。

多态性

- 多态是指在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或行为在一般类及其各个特殊类中具有不同的语义。
- 例如：
  - 图形类：计算图形面积->圆类计算圆面积
    ->矩形类计算机型面积

面向对象的软件工程

- 面向对象的软件工程是面向对象方法在软件工程领域的全面应用。
  - 面向对象的分析（OOA）
  - 面向对象的设计（OOD）
  - 面向对象的编程（OOP）
  - 面向对象的测试（OOT）
  - 面向对象的软件维护（OOSM）

分析

- 系统分析阶段应该扼要精确地抽象出系统必须做什么，但是不关心如何去实现。
- 面向对象的系统分析，直接用问题域中客观存在的事物建立模型中的对象，对单个事物及事物之间的关系，都保留他们的原貌，不做转换，也不打破原有界限而重新组合，因此能够很好地映射客观事物。

设计

- 针对系统的一个具体实现运用面向对象的方法。包括两方面的工作：
  - 把OOA模型直接搬到OOD，作为OOD的一部分
  - 针对具体实现中的人机界面、数据存储、任务管理等因素补充一些与实现有关的部分。

编程

OOP工作就是用一种面向对象的编程语言把OOD模型中的每个成分书写出来，是面向对象的软件开发最终落实的重要阶段。

测试

- 测试的任务是发现软件中的错误。
- 在面向对象的软件测试中继续运用面向对象的概念与原则来组织测试，以对象的类作为基本测试单位，可以更准确地发现程序错误并提高测试效率。

维护

- 将软件交付使用后，工作并没有完结，还要根据软件的运行情况和用户的需求，不断改进系统。
- 使用面向对象的方法开发的软件，其程序与问题域是一致的，因此，在维护阶段运用面向对象的方法可以大大提高软件维护的效率。

基本术语

- 源程序：
  - 用源语言写的，有待翻译的程序
- 目标程序：
  - 也称为"结果程序"，是源程序通过翻译程序加工以后所生成的程序。
- 翻译程序：
  - 是指一个把源程序翻译成等价的目标程序的程序。

三种不同类型的翻译程序

- 汇编程序：
  - 其任务是把用汇编语言写成的源程序，翻译成机器语言形式的目标程序。
- 编译程序：
  - 若源程序是用高级程序设计语言所写，经翻译程序加工生成目标程序，那么，该翻译程序就称为"编译程序"。
- 解释程序：
  - 这也是一种翻译程序，同样是将高级语言源程序翻译成机器指令。它与编译程序不同点就在于：它是边翻译边执行的，即输入一句、翻译一句、 执行一句，直至将整个源程序翻译并执行完毕。

程序的开发过程

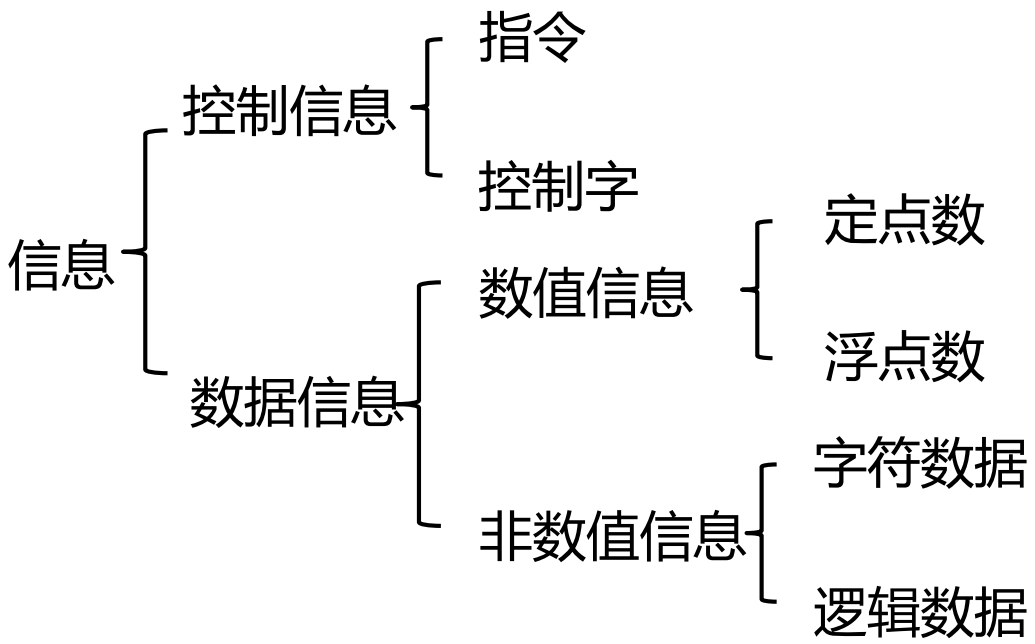
- 编辑
  - 将源程序输入到计算机中，生成后缀为cpp的磁盘文件。
- 编译
  - 将程序的源代码转换为机器语言代码。
- 连接
  - 将多个源程序文件以及库中的某些文件连在一起，生成一个后缀为exe的可执行文件。
- 运行调试

信息的表示与存储

自学

计算机中的信息

- 计算机内部的信息可以分成二大类：
  - 数据信息——计算机程序加工的对象
  - 控制信息——指挥计算机操作



信息的存储单位

- 位(bit，b)：度量数据的最小单位，表示一位二进制信息。
- 字节(byte，B)：由八位二进制数字组成(1 byte = 8 bit)。
  - 千字节　1 KB = 1024 B
  - 兆字节　1 MB = 1024 K
  - 吉字节　1 GB = 1024 M

计算机的数字系统

- 计算机采用的是二进制数字系统。
- 基本符号：0、1
- 进位原则：逢二进一
- 优点：
  - 易于物理实现
  - 二进制数运算简单
  - 机器可靠性高
  - 通用性强
- 缺点：对人来说可读性差

程序设计中常用的数制

进制	基 数	进位原则	基本符号
二进制	2	逢 2 进 1	0, 1
八进制	8	逢 8 进 1	0, 1, 2, 3, 4, 5, 6, 7
十进制	10	逢 10 进 1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
十六进制	16	逢 16 进 1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

R 进制→十进制

- 各位数字与它的权相乘，其积相加。
- 例如:

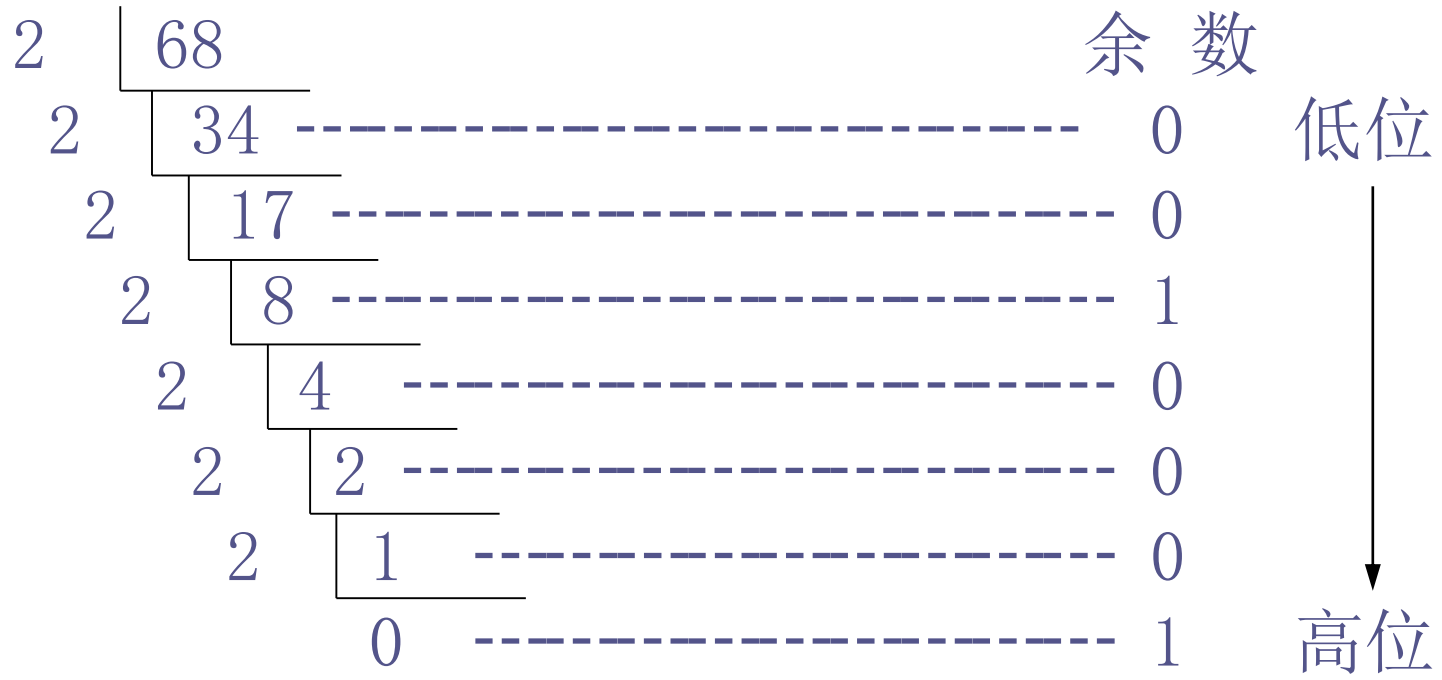
$(11111111.11)_2$
$=1\times2^7+1\times2^6+1\times2^5+1\times2^4+1\times2^3+1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$
$=(255.75)_{10}$

$(3506.2)_8=3\times8^3+5\times8^2+0\times8^1+6\times8^0+2\times8^{-1}=(1862.25)_{10}$

$(0.2A)_{16}=2\times16^{-1}+10\times16^{-2}=(0.1640625)_{10}$

十进制→ R 进制

- 十进制整数转换成R进制的整数：“除R取余”法，例如：

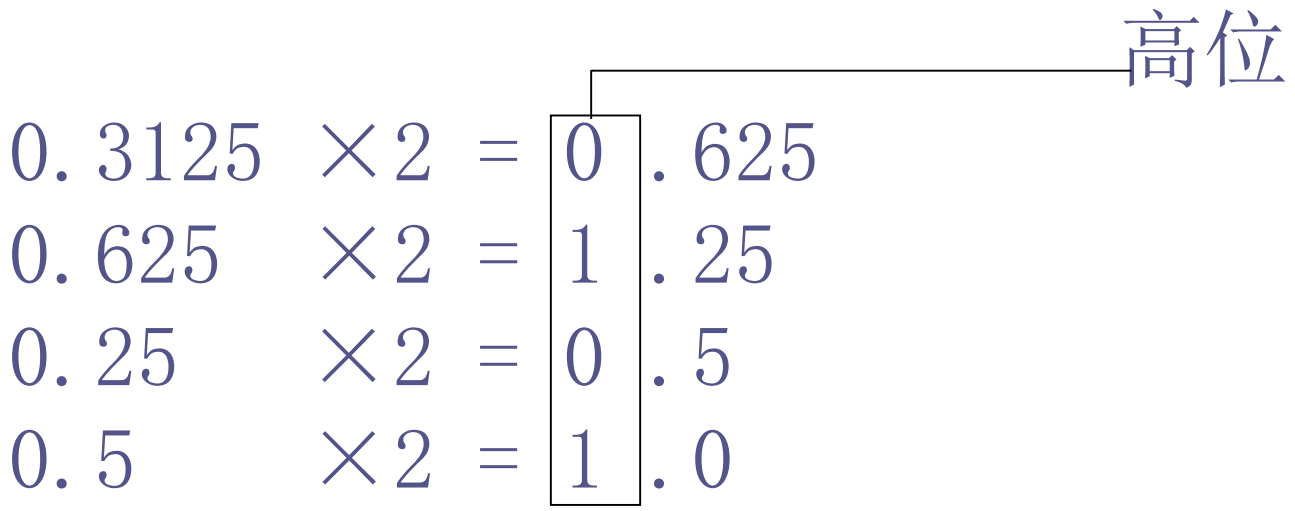


所以 $6810 = 1000100_2$

十进制→ R 进制（续）

- 十进制小数转换成R进制小数

  “乘 R 取整”法，例如：



所以 $0.3125_{10} = 0.0101_2$

二、八、十六进制的相互转换

- 每位八进制数相当于三位二进制数
- 每位十六进制数相当于四位二进制数

$(1011010.10)_2=(\underline{001}\ \underline{011}\ \underline{010}\ .\underline{100})_2=(132.4)_8$

$(1011010.10)_2=(\underline{0101}\ \underline{1010}\ .\underline{1000})_2=(5A.8)_{16}$

$(F7)_{16}=(\underline{1111}\ \underline{0111})_2=(11110111)_2$

二进制数的编码表示:原码

- "符号——绝对值表示"的编码
  例如：
  X=+0101011　　$[X]_{原}$= 0 0101011
  X=- 0101011　　$[X]_{原}$= 1 0101011
  符号位
- 缺点：
  - 零的表示不惟一：
    $[+0]_{原}$ =000...0 $[-0]_{原}$ =100...0
  - 进行四则运算时，符号位须单独处理，且运算规则复杂。

二进制数的编码表示:反码

- 正数的反码与原码表示相同。
- 负数的反码与原码有如下关系：

  符号位相同(仍用1表示)，其余各位取反(0变1，1变0)。例如：

  X=-1100110　$[X]_{原}$ =11100110　$[X]_{反}$ =10011001

  X=+0000000　$[X]_{原}$ =00000000　$[X]_{反}$ =00000000
- 反码中零的表示也不惟一

  X=-0000000　$[X]_{原}$ =10000000　$[X]_{反}$ =11111111
- 反码只是求补码的中间码

二进制数的编码表示:补码

- 模数：
  - n位整数(包括一位符号位)，则它的模数为 2^n。 n位小数，小数点前一位为符号位，则它的模数为 2。
- 补数：
  - 一个数减去另一个数，或者说一个数加上一个负数，等于第一个数加上第二个数的补数。例（时钟指针）：

    8+(-2)=8+10 (mod 12)
  - 一个二进制负数可用其模数与真值做加法 (模减去该数的绝对值) 求得其补码。

二进制数的编码表示:补码（续）

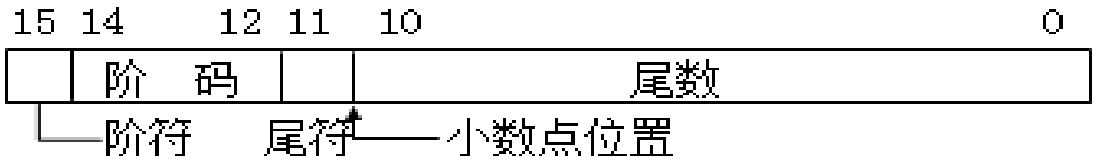
- 计算机中的补码表示法
  - 负数的补码由该数反码的末位加 1 求得
  - 对补码再求补即得到原码
- 补码运算规则
  - 符号位可作为数值参加运算
  - 减法运算可转换为加法运算：
    - 加上一个负数等于加上该数的补码
  - 补码运算的结果仍为补码
  - 运算结果溢出：
    - 负数之和得正数，或正数之和得负数

实数的浮点表示

- 计算机中通常采用浮点方式表示小数
  一个数 N 用浮点形式表示可以写成：
  $N=M\times 2^{E}$
  - E表示2的幂，称为数N的阶码。阶码确定了数N的小数点的位置，其位数反映了该浮点数所表示的数的范围。
  - M表示数N的全部有效数字，称为数N的尾数。其位数反映了数据的精度。

实数的浮点表示（续）

- 浮点数的具体格式随不同机器而有所区别。例如，假设有一台16位机，其二进制浮点数组成为阶码4位，尾数12位，则浮点数格式如下：



下面是一个实际的例子，其中阶码，尾数分别用补码和原码表示

0	0 1 0	1	110..... 0	表示 $(-0.11\times10^{10})_2$
1	1 0 1	0	110..... 0	表示 $(0.11\times10^{-11})_2$

数的表示范围

- 机器中数的表示范围与数据位数及表示方法有关。一个M位整数(包括一位符号位)：
  - 如果采用原码或反码表示法，能表示的最大数为$2^{m-1}-1$，最小数为$-(2^{m-1}-1)$。
  - 若用补码表示，表示范围为$-2^{m-1} \sim 2^{m-1}-1$。
- n位定点小数
  - 采用原码或反码表示时，范围为$-(1-2^{-n}) \sim (1-2^{-n})$
  - 采用补码表示时，范围为$-1 \sim (1-2^{-n})$。
- 浮点数的表示范围由阶码位数和尾数位数决定
  - 若阶码用r位整数(补码)表示，尾数用n位定点小数(原码)表示，则浮点数范围是：$-(1-2^{-n})\times 2^{(2^{r-1}-1)} \sim +(1-2^{-n})\times 2^{(2^{r-1}-1)}$

非数值信息的表示

- 字符在计算机中是通过编码来表示的
  - ASCII码是一种常用的西文字符编码：用7位二进制数表示一个字符，最多可以表示2^7=128个字符
  - EBCDIC码：用8位二进制数表示一个字符，最多可以表示2^8=256个字符
  - "信息交换用汉字编码字符集·基本集"(GB2312-80标准)，简称国标码：是应用较为广泛的汉字编码。二字节码，用二个七位二进制数编码表示一个汉字。

课后任务

- 按课表上实验课
  - 课表第1周没有实验课
  - 第二周完成实验一
- 完成雨课堂平台作业