

C++ 面向对象程序设计

人工智能 学院

田原

课程纲要

① 课程安排

①

第一章 绪论

计算机程序设计语言的发展

面向对象的方法

面向对象的软件开发

信息表示与存储

程序开发基本概念

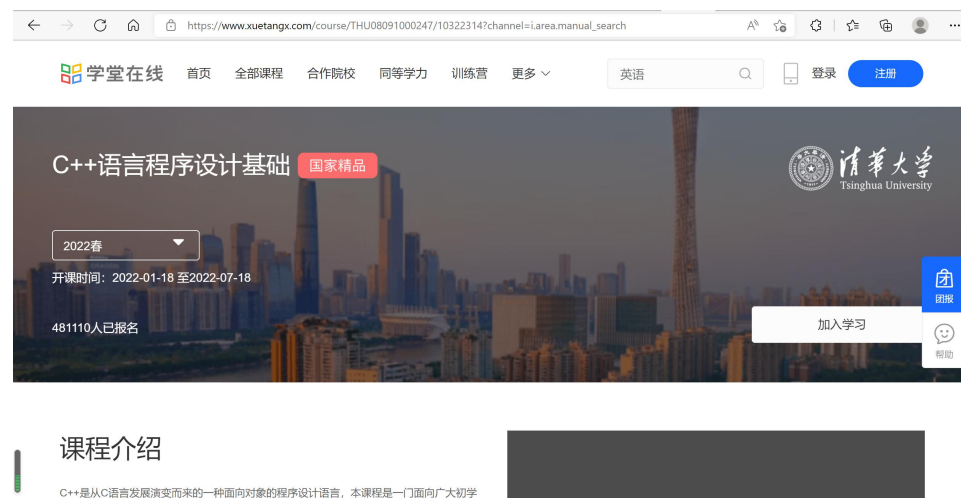
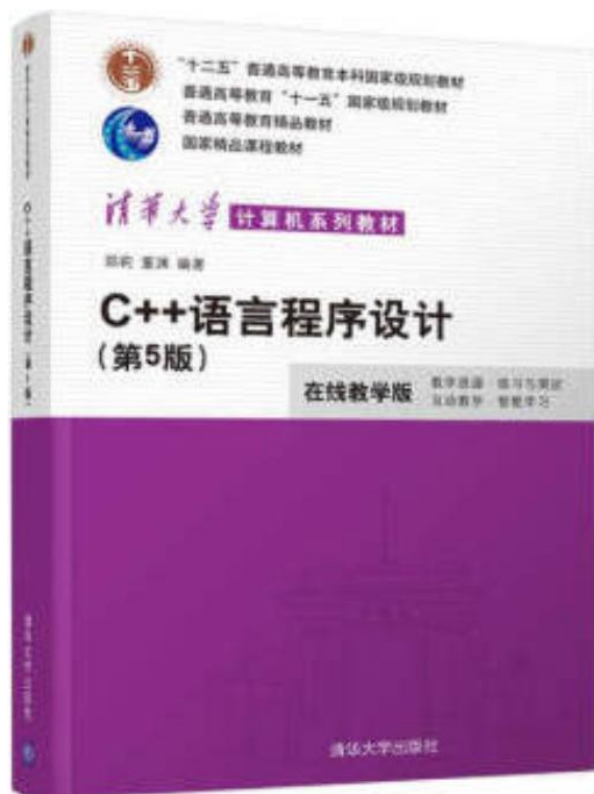
②

第二章 C++ 简单程序设计

C++ 语言概述

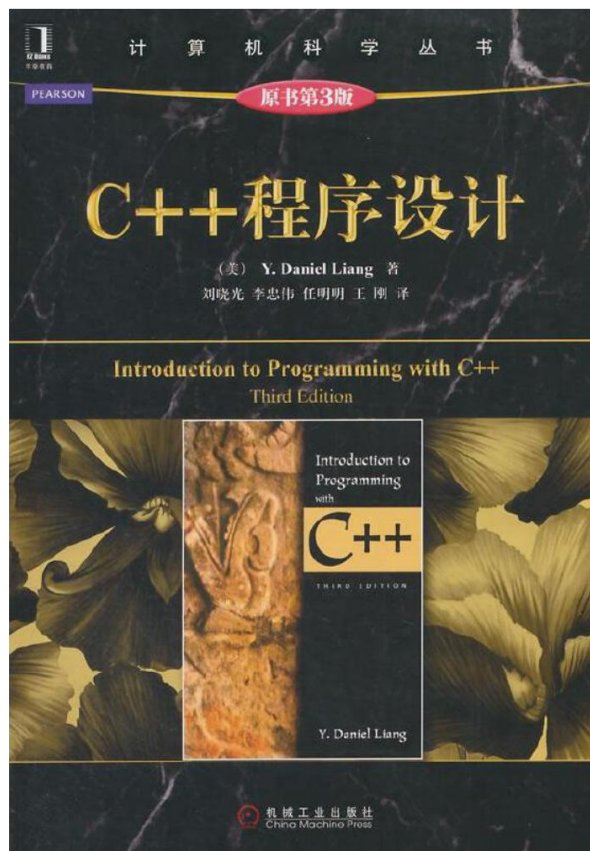
主讲书目

- 《C++语言程序设计》
清华大学出版社 郑莉等

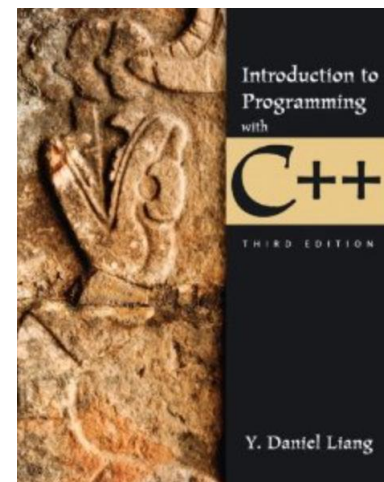


- 学堂在线: <https://www.xuetangx.com/>

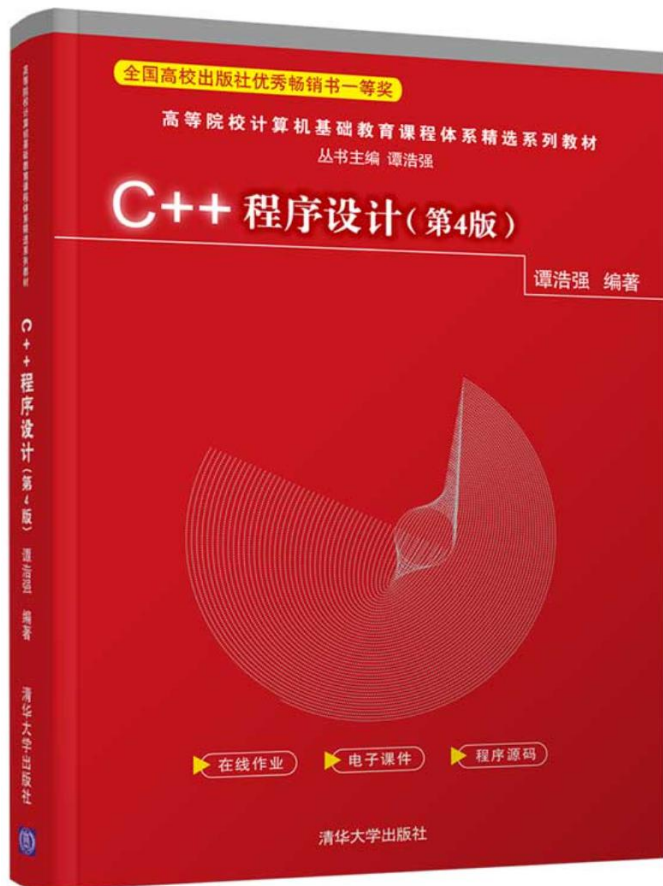
参考书目



- 《C++程序设计》
机械工业出版社 刘晓光等
- 《Introduction to Programming with C++》
出版社: Prentice Hall, Y. Daniel Liang (梁勇)
- 共16章
1-8章: 编程基础
9-16章: 面向对象



参考书目



■ 《C++程序设计》

清华大学出版社 谭浩强等

■ 共14章

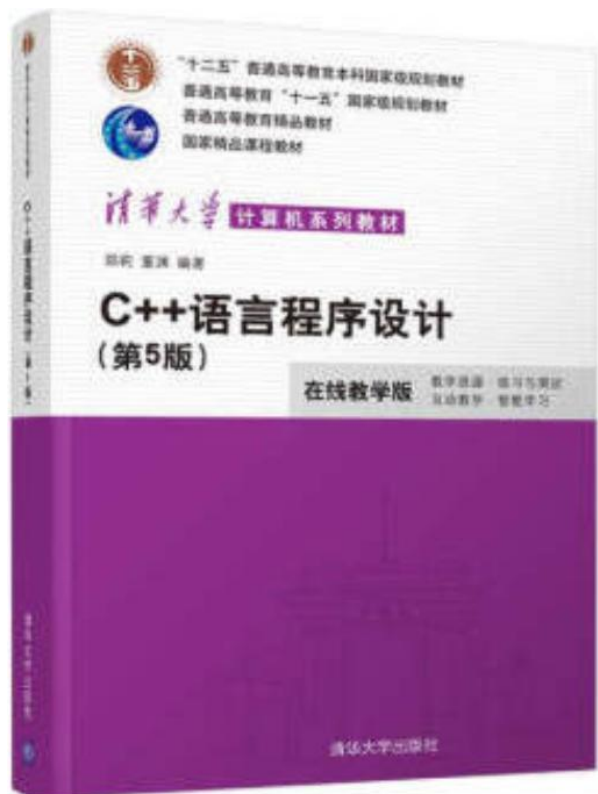
1-2章：基本知识

3-7章：关于过程的程序设计

8-10章：基于对象程序设计

11-14章：面向对象程序设计

课程安排



■ 共12章

1-3章：面向对象程序设计基础

主讲人：田原 副教授

4-6章：基于对象的程序设计

主讲人：杨溪 副教授

7-9章：面向对象程序设计

主讲人：马锐 副教授

10-12章：面向对象程序设计扩展

主讲人：王琪 助理教授

■ 48课时 讲授+32课时 实验

授课目标

- 4层：可以编写有效率程序
★ 课堂能学的
- 3层：可以编写程序
★ 课堂能学的
- 2层：可以改写程序
- 1层：可以看懂程序
- 0层：没有基础
☆ 你在这里



课程纲要

- ① 课程安排
- ② **第一章 绪论**
计算机程序设计语言的发展
面向对象的方法
面向对象的软件开发
信息表示与存储
- ③ 第二章 C++ 简单程序设计
C++ 语言概述

什么是计算机程序设计语言

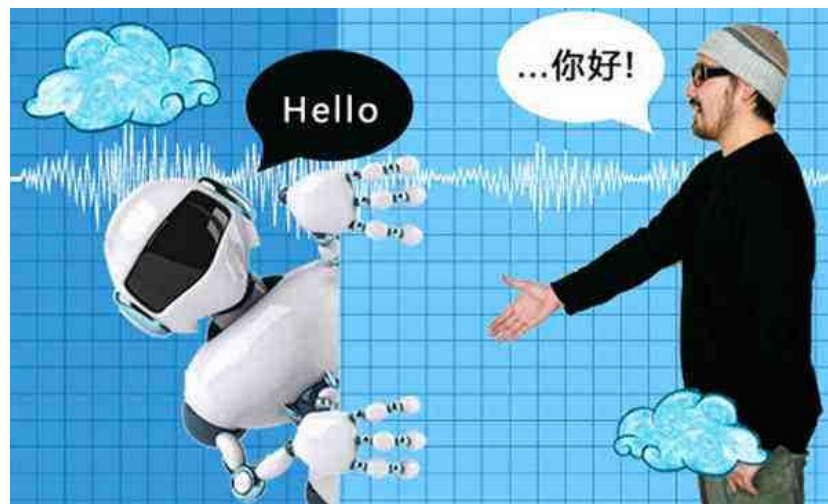
□ 语言：一套具有语法、词法的系统，用于交换信息。

□ 计算机程序设计语言：计算机可以识别的语言，用于人与计算机交换信息，指导计算机工作。

✓ 计算机的工作是用程序来控制的。

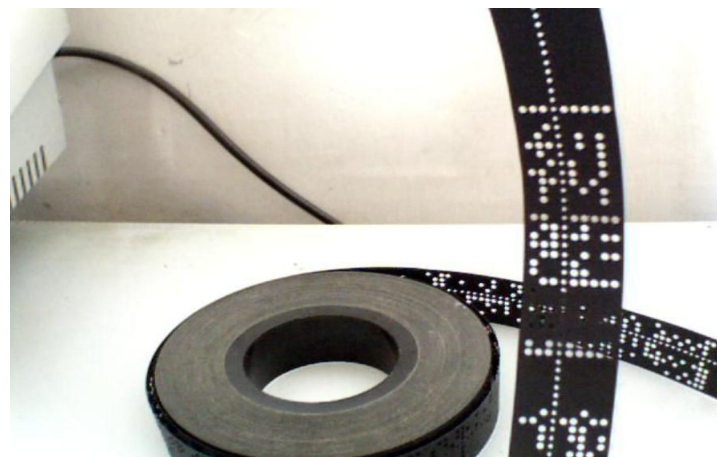
✓ 程序是指令的集合。

✓ 指令是计算机可以识别的命令。



第一代 机器语言

- 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。
- 特点：0、1组成的二进制代码构成指令，不同的CPU指令不一样，不能通用。
- 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。



第二代 汇编语言

- 汇编语言将机器指令映射为一些可以被人们读懂的助记符，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。

；实例：计算ffff:0~ffff:b 单元 中的数据之和，存储在dx中。

```
assume cs:TT
TT segment
    start :
        mov ax,0ffffh
        mov ds,ax

        mov dx,0

        mov bx,0

        mov cx,12
tag:
        mov al,[bx]
        mov ah,0
        add dx,ax
        inc bx
        loop tag

        mov ax,4c00h
        int 21h

TT ends
```

第三代 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涵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。

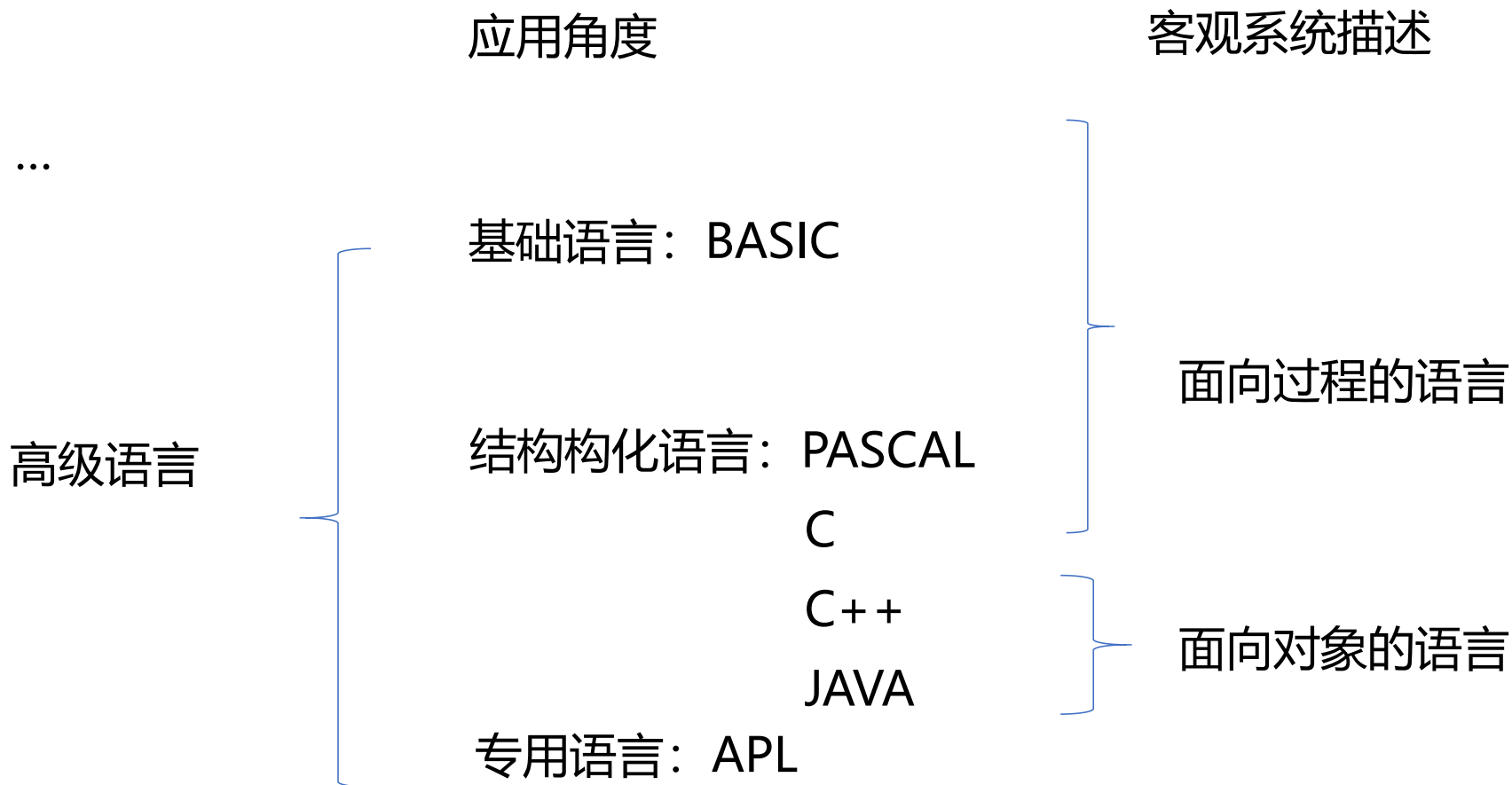
代码示例

以下代码为m与a两数相乘输出结果：

```
1 PROGRAM product
2 LET m = 2
3 LET a = 4
4 LET force = m*a
5 PRINT force
6 END
```



第三代 高级语言



第三代 高级语言

□ 面向过程 VS 面向对象

游记

暑假到了，一家人决定带我们去游乐场玩，我高兴极了！心想：这可是第一次去游乐场玩呀，我可要痛痛快快的玩耍，可不能再闹嘴了。到了第二天，我赶紧起床、刷牙、洗脸，很快把东西全部准备好了，就等着出发了，这是妈妈走过来说：“走，去游乐场，舅舅在车上等我们呢！在去的路上，坐在车子上真是舒服，既能感受到微风轻轻地拂过我的面颊，还能看大路旁的那些美丽的景色，有各色鲜艳的花，还有绿油油的小草，美丽极了！到了游乐场的大门口，我们走进去哇！游乐场里面人山人海。也有好多好玩的，我第一眼就看到了摩天轮，我跟妈妈说：“妈妈我想玩摩天轮好吗？妈妈爽快地答应了。我们坐上摩天轮，摩天轮就开动了，先是转的很慢，可以看见大美丽的风景，坐在摩天轮上，可以看到真个游乐场，有蹦蹦车、过山车、还可以划船、蹦蹦床等等。可是慢慢地摩天轮不知不觉的转快了，把妈妈、舅舅、舅妈都给转晕了，我心想我都好好的他没尽然都折磨虚弱唉，太没意思。下摩天轮时，妈妈已经呕吐的不成样了。我看见妈妈成了现在这样，都把我乐的哈哈大笑。妈妈说以后我再也不坐摩天轮啦！我们饿了吃汉堡渴了吃冰棒真是开心。时间过得可真快，转眼就快到了黑夜，我们不得不回家了，在回家的路上我还是恋恋不舍。这真是快乐的一天呀！

小说

VS



第三代 高级语言

□ 面向对象

- 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。

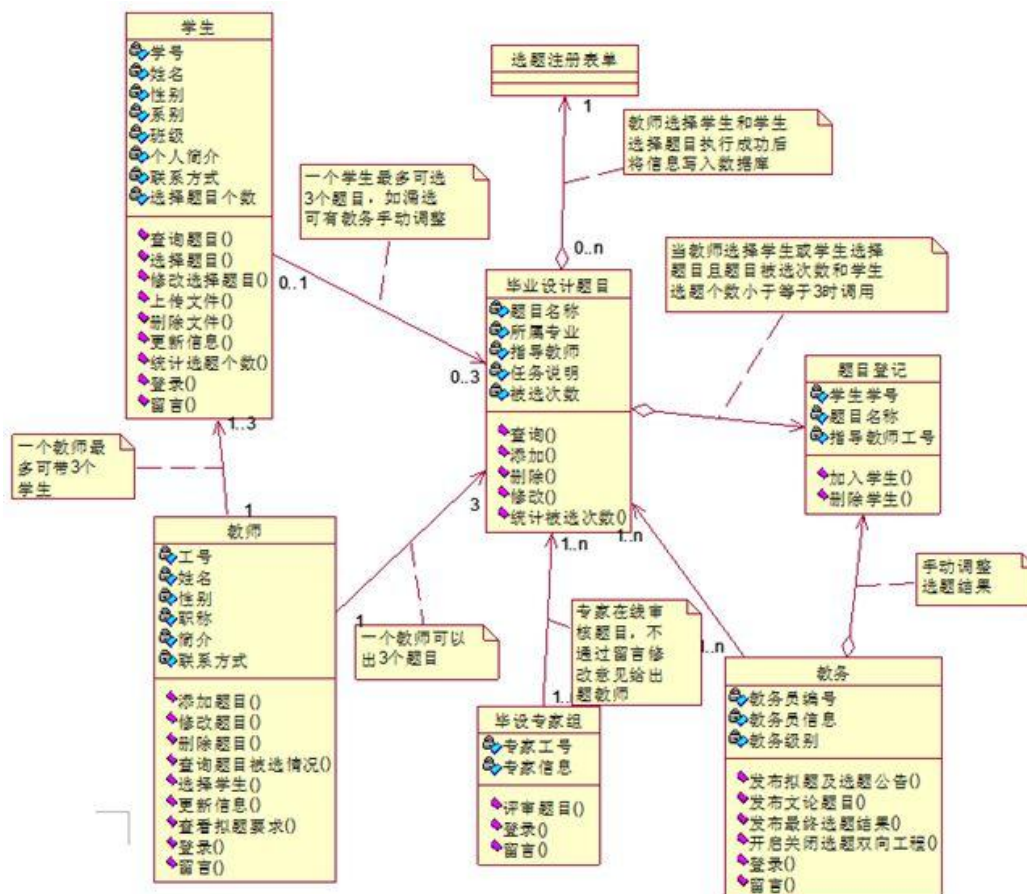


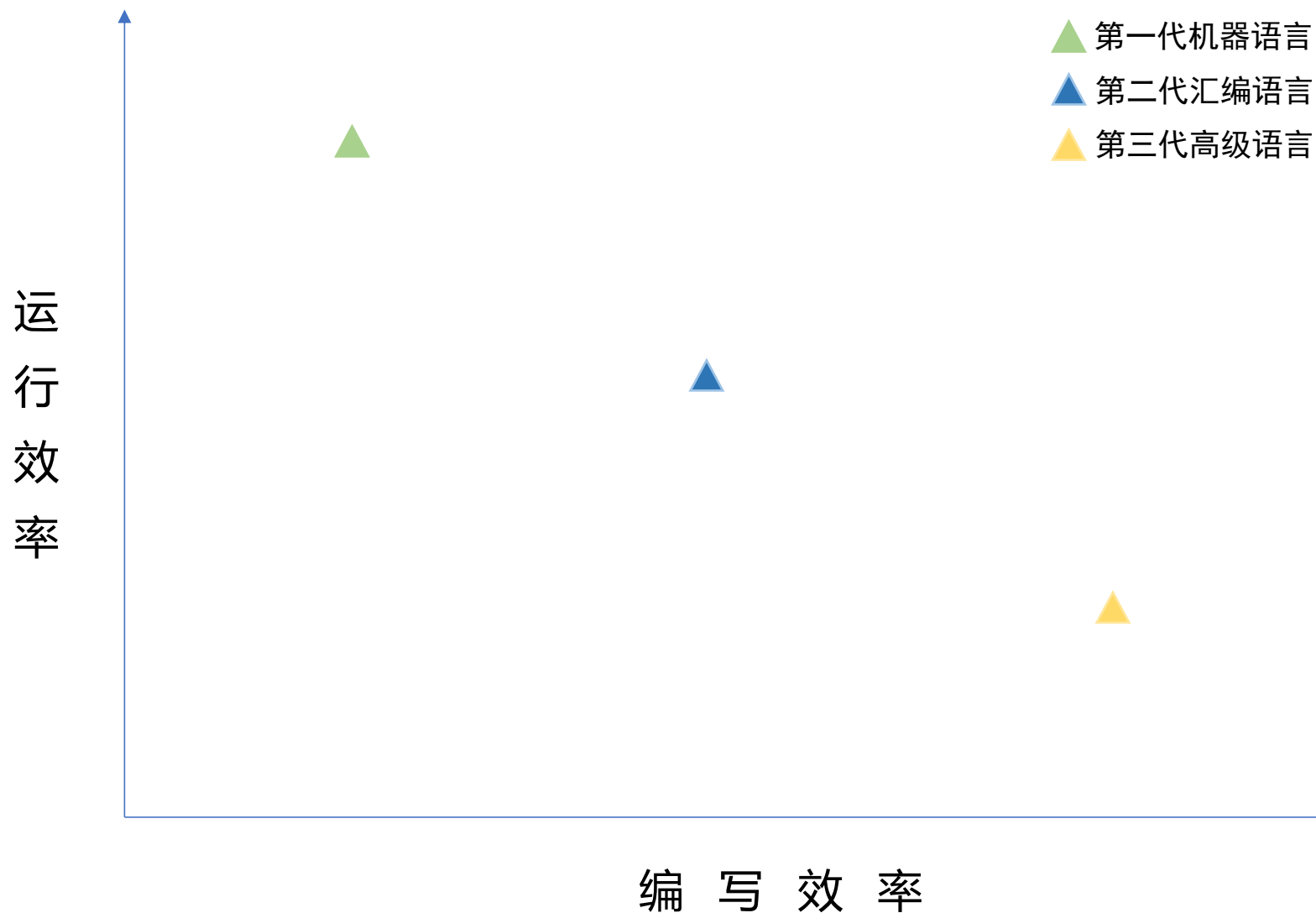
第三代 高级语言

□ 面向对象

● 优点:

使程序能够比较直接地反问题域的本来面目, 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进行软件开发。





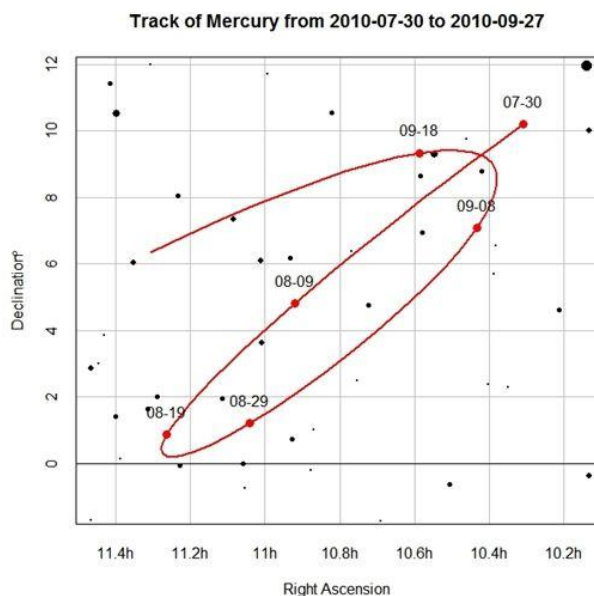
课程纲要

- ① 课程安排
- ② **第一章 绪论**
 - 计算机程序设计语言的发展
 - 面向对象的方法**
 - 面向对象的软件开发
 - 信息表示与存储
- ③ 第二章 C++ 简单程序设计
 - C++ 语言概述

程序设计方法的发展

(1) 面向过程的程序设计方法

- ❑ 程序的目的：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



程序设计方法的发展

(1) 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：

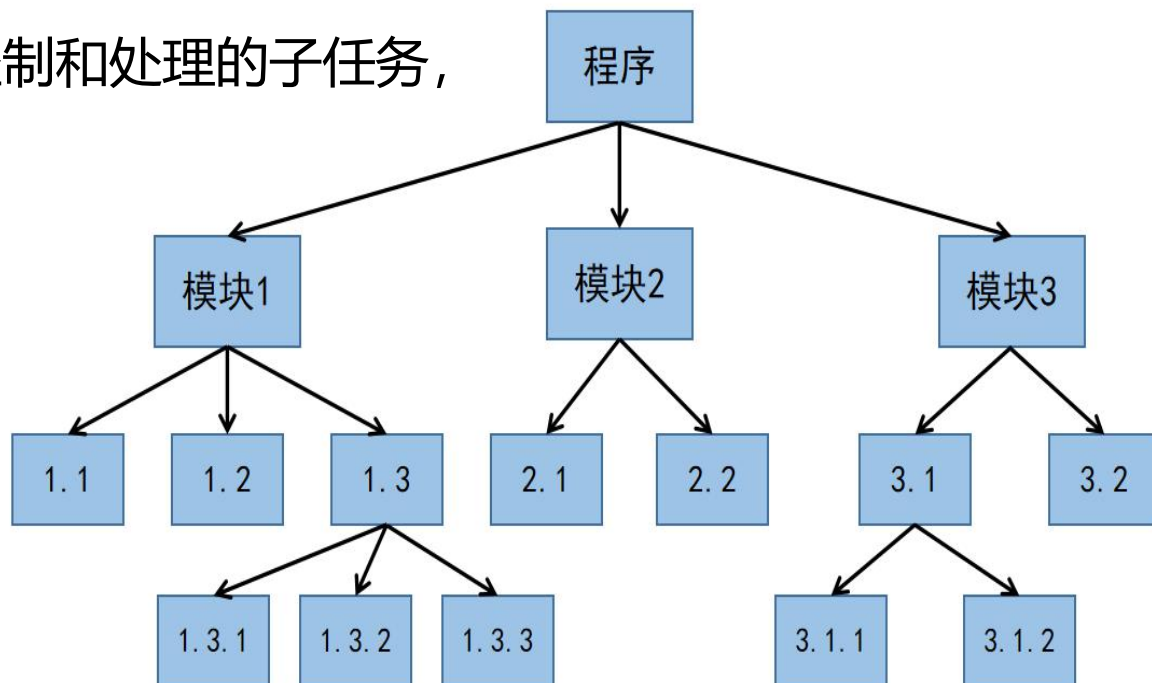
- 按功能划分为若干个基本模块，形成一个树状结构。
- 各模块间的关系尽可能简单，功能上相对独立；每一模块内部均是由顺序、选择和循环三种基本结构组成。
- 其模块化实现的具体方法是使用子程序。

程序设计方法的发展

(1) 面向过程的结构化程序设计方法

□ 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。



程序设计方法的发展

(1) 面向过程的结构化程序设计方法

- 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。

程序设计方法的发展

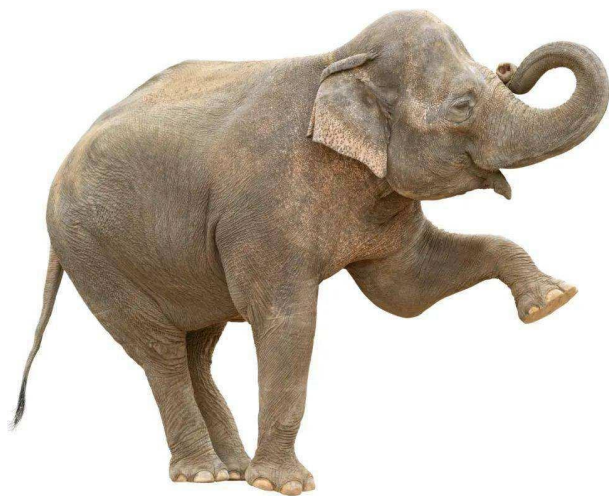
(2) 面向对象的程序设计方法：对象

□ 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

程序设计方法的发展

(2) 面向对象的程序设计方法：对象



静态特征： 生命值
动态特征： 新陈代谢
繁殖
跑



静态特征： 生命值
动态特征： 新陈代谢
繁殖
飞

程序设计方法的发展

(2) 面向对象的程序设计方法：对象

□ 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。

程序设计方法的发展

(2) 面向对象的程序设计方法：类

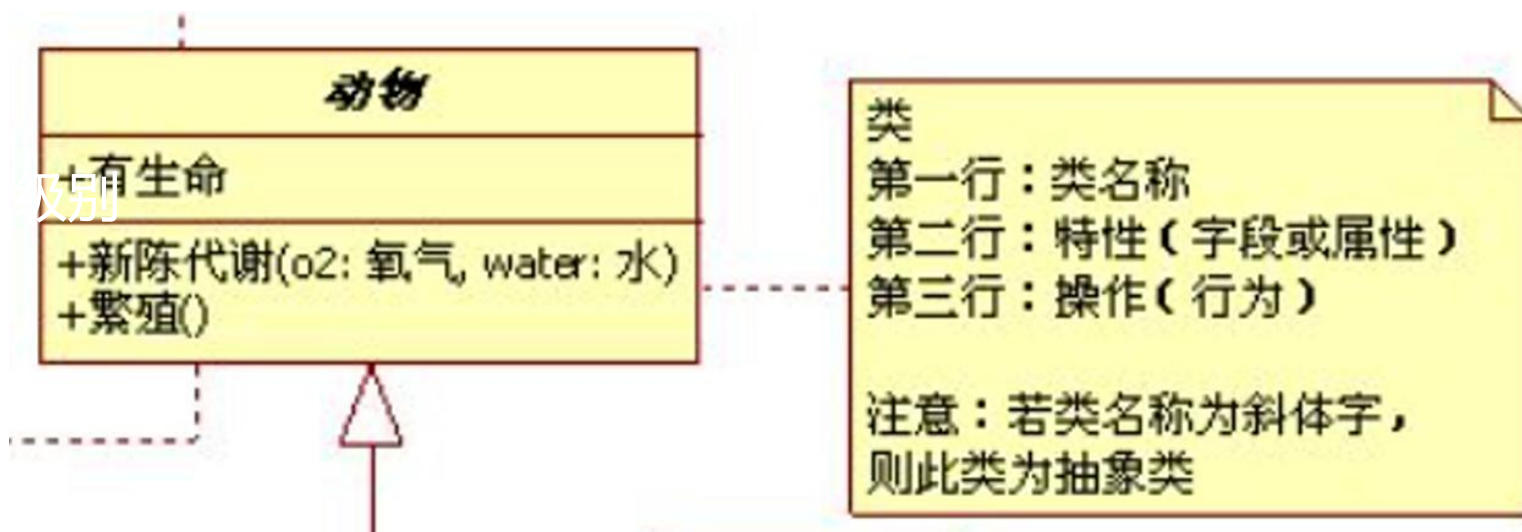
- 分类——人类通常的思维方法
- 分类所依据的原则——抽象
 - 忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽象的概念。
 - 例如，石头、树木、汽车、房屋等都是人们在长期的生产和生活实践中抽象出的概念。

程序设计方法的发展

(2) 面向对象的程序设计方法：类

□ 面向对象方法中的"类"

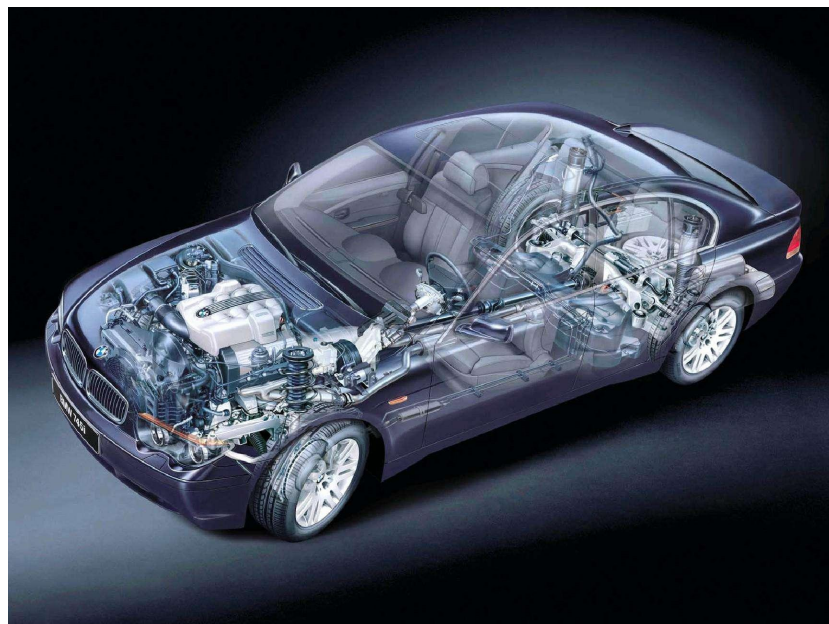
- 为属于该类的全部对象提供了抽象的描述，包括共有属性和行为。
- **类与对象的关系**：犹如模具与铸件之间的关系，一个属于某类的对象称为该类的一个实例。



程序设计方法的发展

(2) 面向对象的程序设计方法：封装

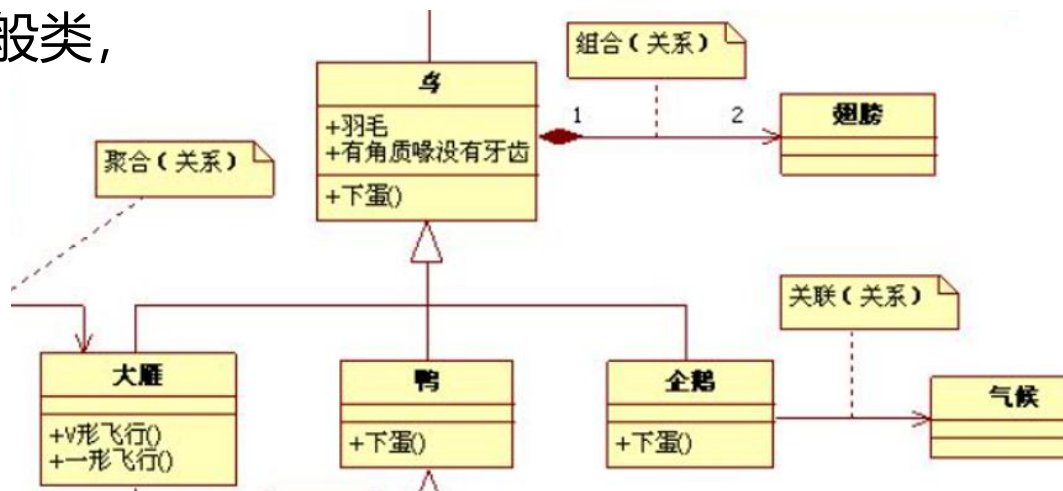
- 把对象的属性和行为结合成一个独立的系统单元。
- 尽可能隐蔽对象的内部细节。对外形成一个边界（或者说一道屏障），只保留有限的对外接口使之与外部发生联系。



程序设计方法的发展

(2) 面向对象的程序设计方法：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行为，称作特殊类对一般类的继承。
- ❑ 例如：将轮船作为一个一般类，客轮便是一个特殊类。



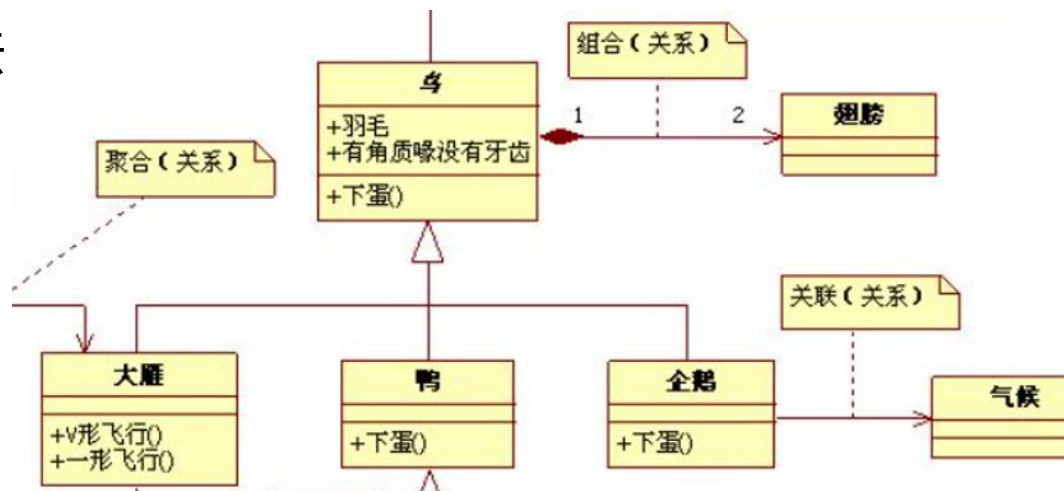
程序设计方法的发展

(2) 面向对象的程序设计方法：多态

- 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：

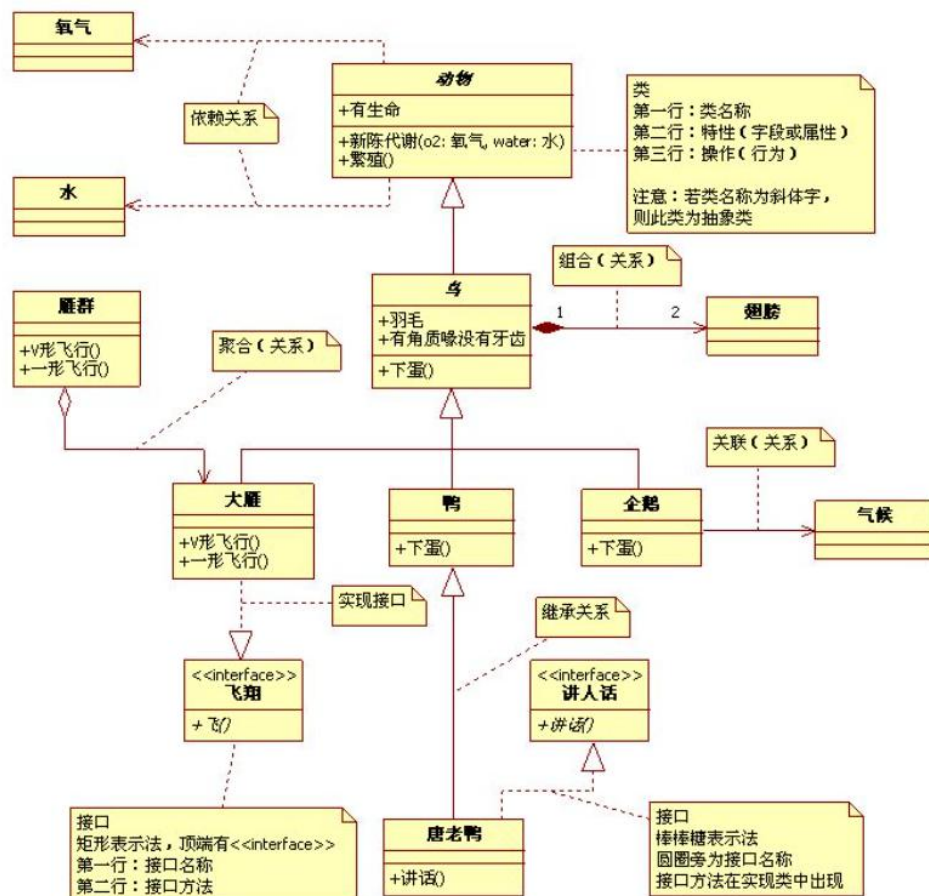
- 数的加法->实数的加法
->复数的加法



程序设计方法的发展

(2) 面向对象的程序设计方法

- ❑ 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讯。



程序设计方法的发展

(2) 面向对象的程序设计方法

□ 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。

程序设计方法的发展

(2) 面向对象的程序设计方法

□ 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。

课程纲要

①

课程安排

②

第一章 绪论

计算机程序设计语言的发展

面向对象的方法

面向对象的软件开发

信息表示与存储

③

第二章 C++ 简单程序设计

C++ 语言概述

面向对象的软件开发

□ 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。

它包括:

- 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)

(1) 面向对象的分析（建模）

- ❑ 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。



公式 (1) :

$$v_i = v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i)$$

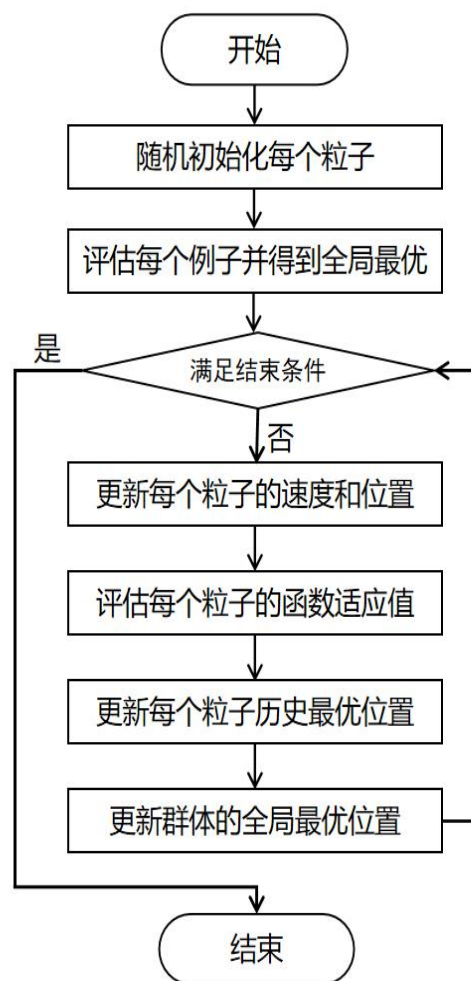
公式 (2) :

$$x_i = x_i + v_i$$

(2) 面向对象的设计（补全）

□ 针对系统的一个具体实现运用面向对象的方法。其中包括两方面的工作：

- 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。



(3) 面向对象的编程（实现）

- 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。

```
//功能：粒子群优化算法伪代码
//说明：本例以求问题最小值为目标
//参数：N为群体规模

procedure PSO
  for each particle  $i$ 
    Initialize velocity  $V_i$  and position  $X_i$  for particle  $i$ 
    Evaluate particle  $i$  and set  $pBest_i = X_i$ 
  end for
   $gBest = \min \{pBest_i\}$ 
  while not stop
    for  $i=1$  to  $N$ 
      Update the velocity and position of particle  $i$ 
      Evaluate particle  $i$ 
      if  $\text{fit}(X_i) < \text{fit}(pBest_i)$ 
         $pBest_i = X_i$ ;
      if  $\text{fit}(pBest_i) < \text{fit}(gBest)$ 
         $gBest = pBest_i$ ;
    end for
  end while
  print  $gBest$ 
end procedure http://blog.csdn.net/daaikuaichuan
```

(4) 面向对象的测试

- 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，可以更准确的发现程序错误并提高测试效率。

(5) 面向对象的软件维护

- 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。

课程纲要

①

课程安排

②

第一章 绪论

计算机程序设计语言的发展

面向对象的方法

面向对象的软件开发

信息表示与存储

③

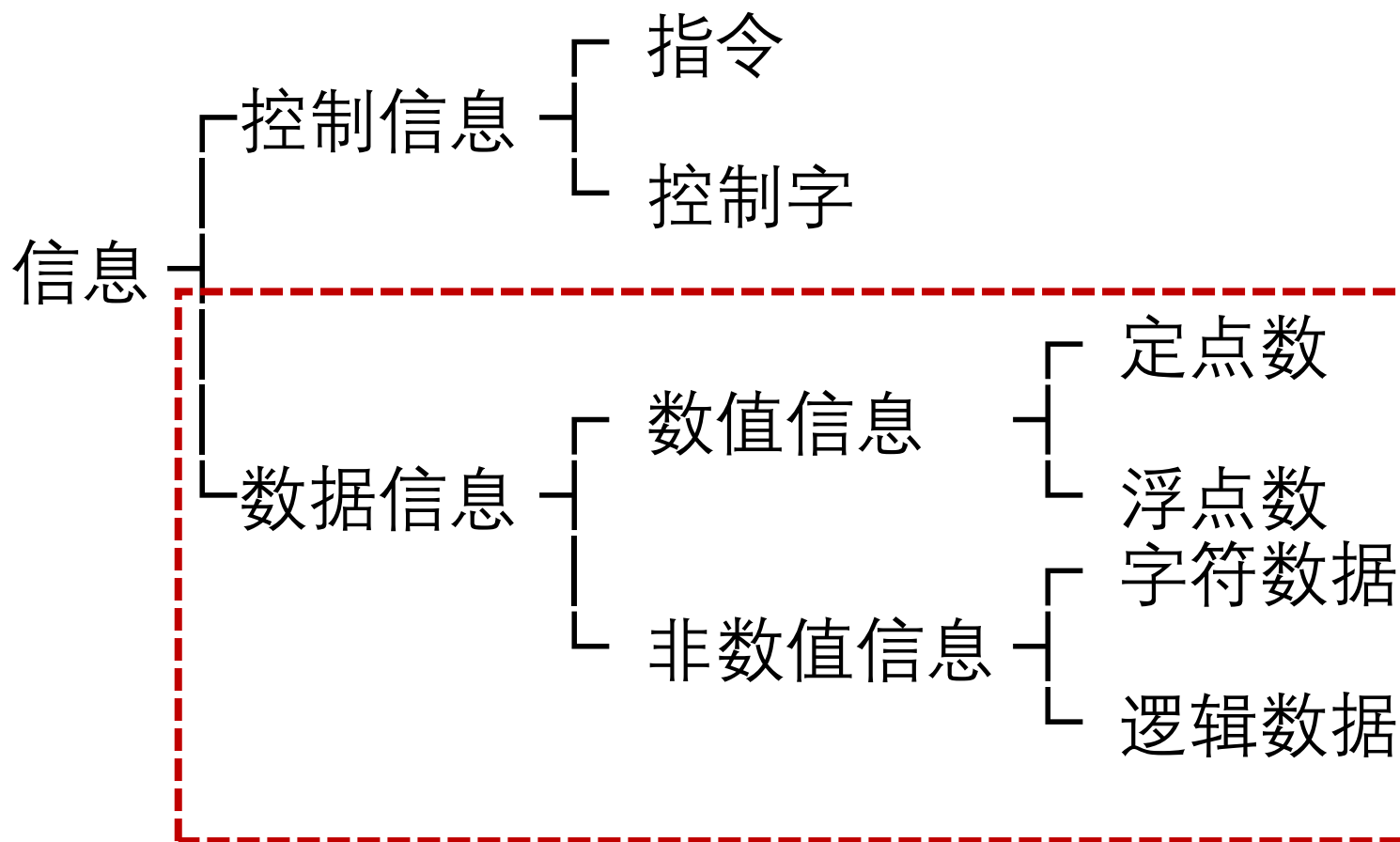
第二章 C++ 简单程序设计

C++ 语言概述

信息的表示和存储

- 信息的分类
- 计算机的数字系统
- 程序设计中常用的数制
- 不同进位计数制间的转换
- 信息的存储单位
- 二进制数的编码表示
- 小数的表示方法
- 非数值信息的表示

信息的分类



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,
16进制	16	逢16进1	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F

不同进位计数制间的转换

(1) R进制->十进制

□各位数字与它的权相乘，其积相加。

□例如：

- $(11111111.11)_2 = 1*2^7 + 1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} + 1*2^{-2} = (255.75)_{10}$
- $(3506.2)_8 = 3*8^3 + 5*8^2 + 0*8^1 + 6*8^0 + 2*8^{-1} = (1862.25)_{10}$
- $(0.2A)_{16} = 2*16^{-1} + 10*16^{-2} = (0.1640625)_{10}$

不同进位计数制间的转换

(1) R进制->十进制

□练习题:

(1) $(1010)_2$

(2) $(10001111)_2$

(3) $(7F)_{16}$

(4) $(2D3E)_{16}$

(5) $(F10E)_{16}$

answer:

(1) $(10)_{10}$

(2) $(143)_{10}$

(3) $(24515)_{10}$

(4) $(127)_{10}$

(5) $(11582)_{10}$

(6) $(61710)_{10}$

不同进位计数制间的转换

(2) 十进制→R进制：整数部分

□ “除R取余” 法

□原理：任何一个十进制数N，都可以用一个R进制数来表示：

$$\begin{aligned} N &= X_0 + X_1 R^1 + X_2 R^2 + \dots + X_{n-1} R^{n-1} \\ &= X_0 + (X_1 + X_2 R^1 + \dots + X_{n-1} R^{n-2}) R \\ &= X_0 + Q_1 R \end{aligned}$$

由此可知，若用N除以R，则商为Q，余数是 X_0 。同理： $Q_1 = X_1 + Q_2 R$

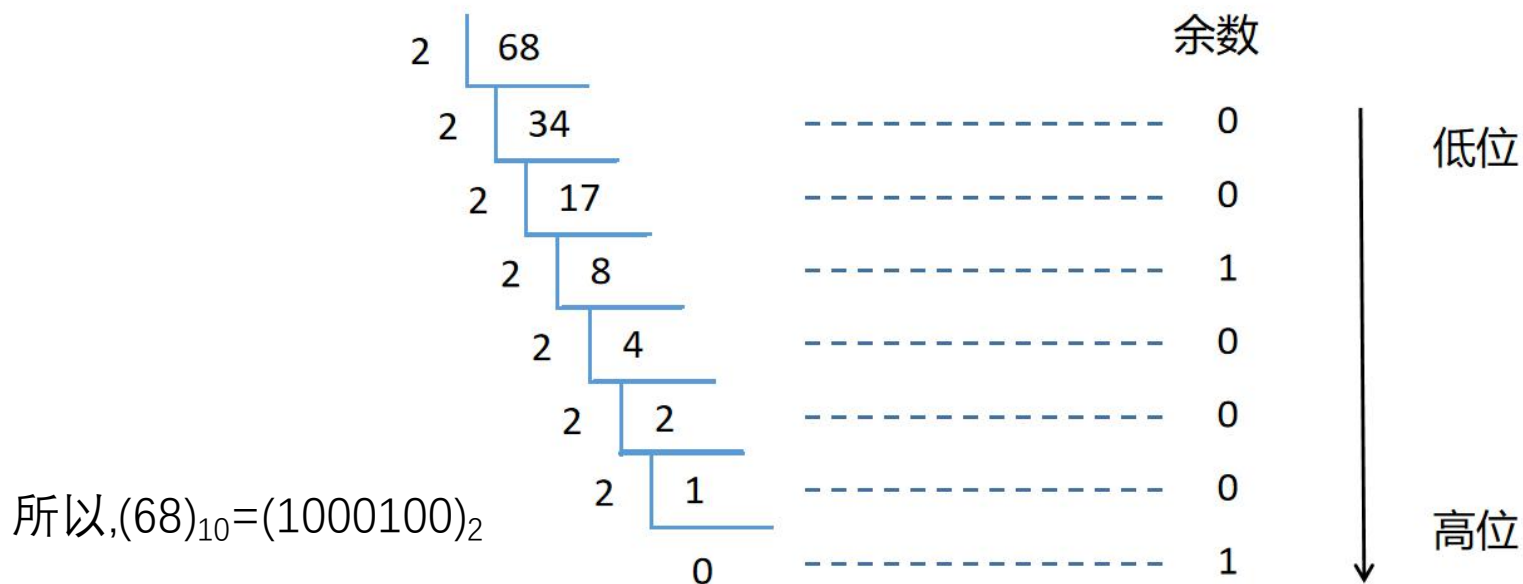
以此类推： $Q_i = X_i + (X_{i+1} + X_{i+2} R^1 + \dots + X_{n-1} R^{n-2-i}) R$

$= X_i + Q_{i+1} R$ 。这样除下去，直到商为0时为止

不同进位计数制间的转换

(2) 十进制->R进制：整数部分

□例题：将十进制数68转为二进制数



不同进位计数制间的转换

(2) 十进制- \rightarrow R进制：整数部分

□练习题：请将以下十进制数值转换为2进制数

(1)2

(2)9

(3)93

answer:

(1)(10)₂

(2)(1001)₂

(3)(1011101)₂

不同进位计数制间的转换

(2) 十进制->R进制：小数部分

□ “乘R取整” 法

□原理：可将某十进制小数用R进制数表示： $V = \frac{X_{-1}}{R^1} + \frac{X_{-2}}{R^2} + \frac{X_{-3}}{R^3} + \dots + \frac{X_{-m}}{R^m}$

等式两边乘以R得到 $V \times R = X_{-1} + (\frac{X_{-2}}{R^1} + \frac{X_{-3}}{R^2} + \dots + \frac{X_{-m}}{R^{m-1}}) = X_{-1} + F_1$

X_{-1} 是整数部分，即R进制小数点后第一位， F_1 是小数部分，小数部分再乘以R： $F_1 \times R = X_{-2} + (\frac{X_{-3}}{R^1} + \frac{X_{-4}}{R^2} + \dots + \frac{X_{-m}}{R^{m-2}}) = X_{-2} + F_2$

X_{-2} 是整数部分，即R进制小数点后第二位。依次乘下去，直到小数部分为0或达到所要求的精度为止（小数部分可能永不为0）。

不同进位计数制间的转换

(2) 十进制→R进制：小数部分

□例题：将 0.3125_{10} 转换成二进制数

高位

0.3125	$\times 2 =$	0.625
0.625	$\times 2 =$	1.25
0.25	$\times 2 =$	0.5
0.5	$\times 2 =$	1.0

所以, $(0.3125)_{10} = (0.0101)_2$

不同进位计数制间的转换

(2) 十进制->R进制：小数部分

□练习题：将 $(68.3125)_{10}$ 转为二进制数

提示：可分别进行整数部分和小数部分的转换，然后再拼一起

answer: $(1000100.0101)_2$

不同进位计数制间的转换

(3) 二、八、十六进制相互转换

□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

例如：

$$\bullet (1011010.10)_2 = (001\ 011\ 010\ .100)_2 = (132.4)_8$$

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

$$\bullet (F7)_{16} = (11110111)_2$$

不同进位计数制间的转换

(3) 二、八、十六进制相互转换

□练习题:

(1) $(1000100)_2$ 转为八进制

(2) $(1000100)_2$ 转为十六进制

(3) $(2D7F)_{16}$ 转为二进制

answer:

(1) $(104)_8$

(2) $(44)_{16}$

(3) $(0010\ 1101\ 0111\ 1111)_2$

信息的存储单位

□位(bit): 度量数据的最小单位, 表示一位二进制信息。

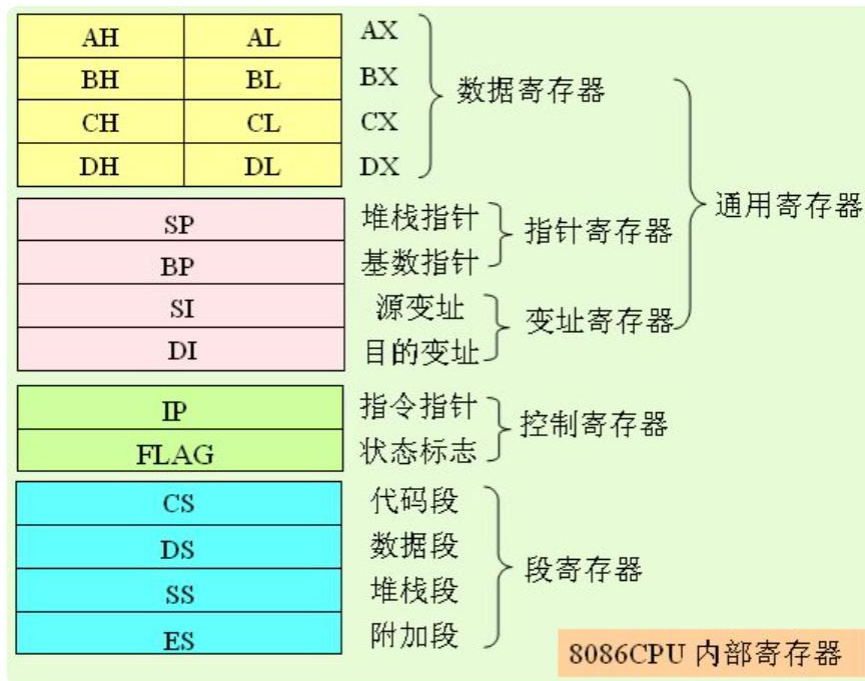
□字节(byte): 由八位二进制数字组成(1 byte = 8 bit), 简称 B。

- K 字节 1 KB = 1024 B
- M 字节 1 MB = 1024 KB
- G 字节 1 GB = 1024 MB

信息的存储单位

□字(word): 位的组合, 作为一个独立的信息处理, 又称计算机字。

□机器字长: 参加运算的寄存器所含有二进制数的位数。



二进制数的编码表示

(1) 原码

□ "符号——绝对值表示"的编码

• 例如：

$X = +0101011$ $[X]_{\text{原}} = 0\ 0101011$

$X = -0101011$ $[X]_{\text{原}} = 1\ 0101011$

符号位

□ 缺点：

• 零的表示不唯一：

$[+0]_{\text{原}} = 000\dots 0$ $[-0]_{\text{原}} = 100\dots 0$ 。

• 进行四则运算时，符号位需单独处理，且运算规则复杂。

二进制数的编码表示

(2) 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：

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

$$X = +0000000 \quad [X]_{\text{原}} = 00000000 \quad [X]_{\text{反}} = 00000000$$

□反码中零的表示也不唯一

- $X = -0000000 \quad [X]_{\text{原}} = 10000000 \quad [X]_{\text{反}} = 11111111$

□反码只是求补码的中间码

二进制数的编码表示

(2) 补码

□模数：从物理意义上，是某种计量容器的容量。在模数系统中有一种特殊的运算叫“取模（趋于）”运算。

如： $14\%12=2$

$$(8 + (-2)) \% 12 = 6$$

$$(8 + 10) \% 12 = 6$$

□补数：A减去B取模，或者说A加上-B取模，等于A加+C取模，则-B和C互为补数。例： $8 + (-2) = 8 + 10 (\%12)$

二进制数的编码表示

(2) 补码

□补码（负数）：

- 当A为n位2进制整数(包括一位符号位)，设其的模数为 2^n 时，其补数称作补码。
- 当A为n位2进制小数(包括一位符号位)，设其的模数为2时，其补数称作补码。
- 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

(2) 补码

例如: $X = -0110$ $[X]_{\text{补}} = 2^4 + (-0110) = 1010$

$X = -0.1011$ $[X]_{\text{补}} = 2 + (-0.1011) = 1.0101$

□ 一个更方便的算法: 对于一个负数, 其补码由该数字的反码的最末尾加1获得。

例如

1. 求 $X = -1010101$ 的补码

$[X]_{\text{原}} = 11010101$

$[X]_{\text{反}} = 10101010$

$[X]_{\text{补}} = 10101011$

2. 求 $X = -0.1011$ 的补码

$[X]_{\text{原}} = 1.1011$

$[X]_{\text{反}} = 1.0100$ (保留原码符号位)

$[X]_{\text{补}} = 1.0101$ (反码 + 0.0001)

二进制数的编码表示

(2) 补码

□优点：

- 补码、原码转化方便，补码的补码是原码。
- 零的表示唯一。
- 0000 0000
- 符号可以直接参与运算，加法与减法运算统一。

二进制数的编码表示

(2) 补码

例如：计算 $67-10=?$

计算机的计算过程如下：

$$[+67_{10}]_{\text{原}} = 01000011_2$$

$$[-10_{10}]_{\text{原}} = 10001010_2$$

$$[+67_{10}]_{\text{补}} = [+67_{10}]_{\text{原}}$$

$$[-10_{10}]_{\text{补}} = 11110110_2$$

$$\begin{array}{r}
 01000011_2 \quad [+67_{10}]_{\text{补}} \\
 + 11110110_2 \quad [-10_{10}]_{\text{补}} \\
 \hline
 1\ 00111001_2 = 57_{10}
 \end{array}$$

最高位的进位自然丢失

二进制数的编码表示

(2) 补码

例如：计算 $10-67=?$

$$\begin{aligned} [-67]_{\text{原}} &= 11000011_2 & [-67]_{\text{补}} &= 10111101_2 \\ [+10]_{\text{原}} &= 00001010_2 & & = [+10]_{\text{补}} \end{aligned}$$

$$\begin{array}{r} 00001010_2 \\ + 10111101_2 \\ \hline \end{array}$$

$$11000111_2$$

$$[\text{结果}]_{\text{补}} = 11000111_2 \qquad [\text{结果}]_{\text{原}} = 10111001_2$$

所以真值为 -0111001 ，十进制是 -57

二进制数的编码表示

(2) 补码

□练习题：将以下十进制数转换为十六进制补码 (2B)

(1)(-32)₁₀

answer:

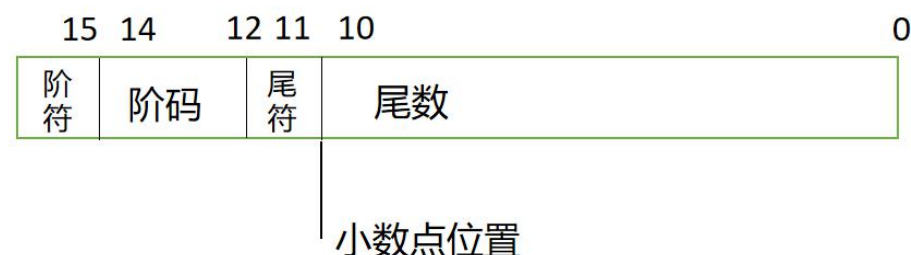
(1)(FFE0)₁₆

小数的表示方法

□定点数：



□浮点数：



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。

非数值信息的表示

□西文字符：

- ASCII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，包括95种打印字符和33控制字符。
- 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。

□汉字：

- 应用较为广泛的是"国家标准信息交换用汉字编码"(GB2312-80标准)，简称国标码。是二字节码，用二个七位二进制数编码表示一个汉字。

课程纲要

① 课程安排

①

第一章 绪论

计算机程序设计语言的发展

面向对象的方法

面向对象的软件开发

信息表示与存储

程序开发基本概念

②

第二章 C++ 简单程序设计

C++ 语言概述

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。

包括：汇编程序、编译程序、解释程序。

基本术语

□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。

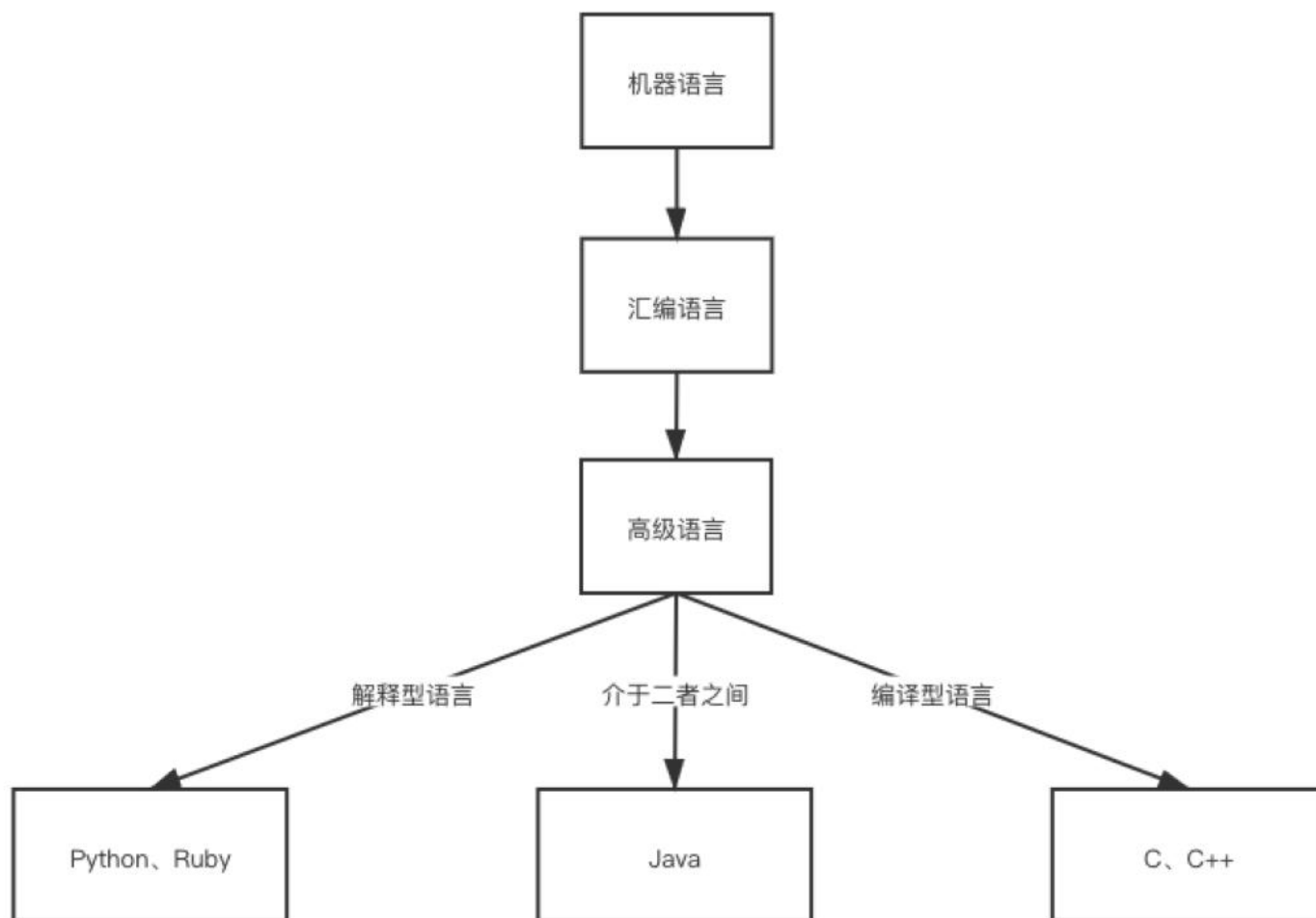
□编译程序：

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"。

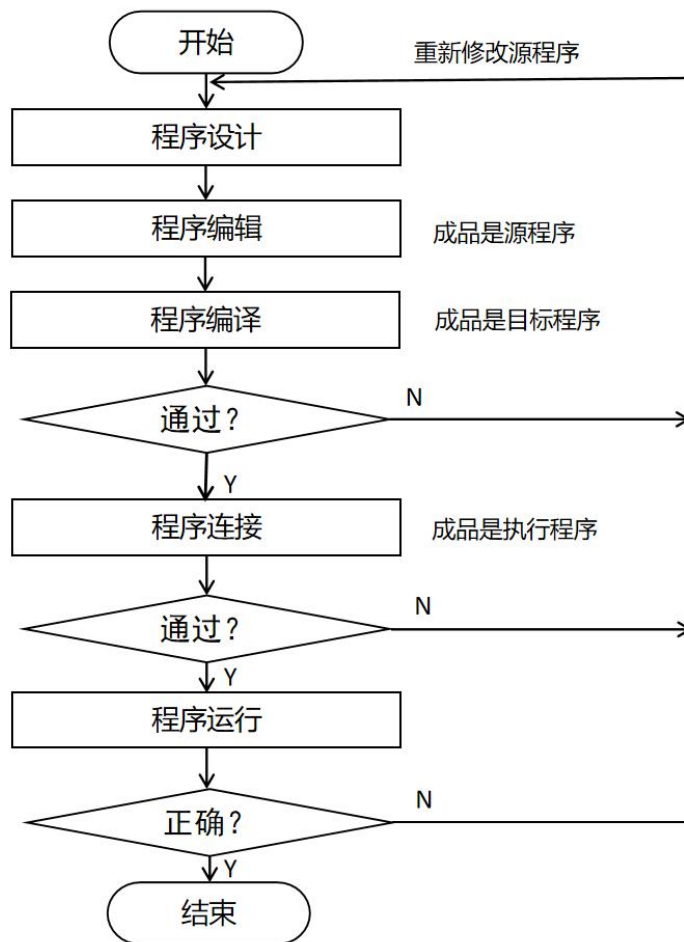
□解释程序：

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。

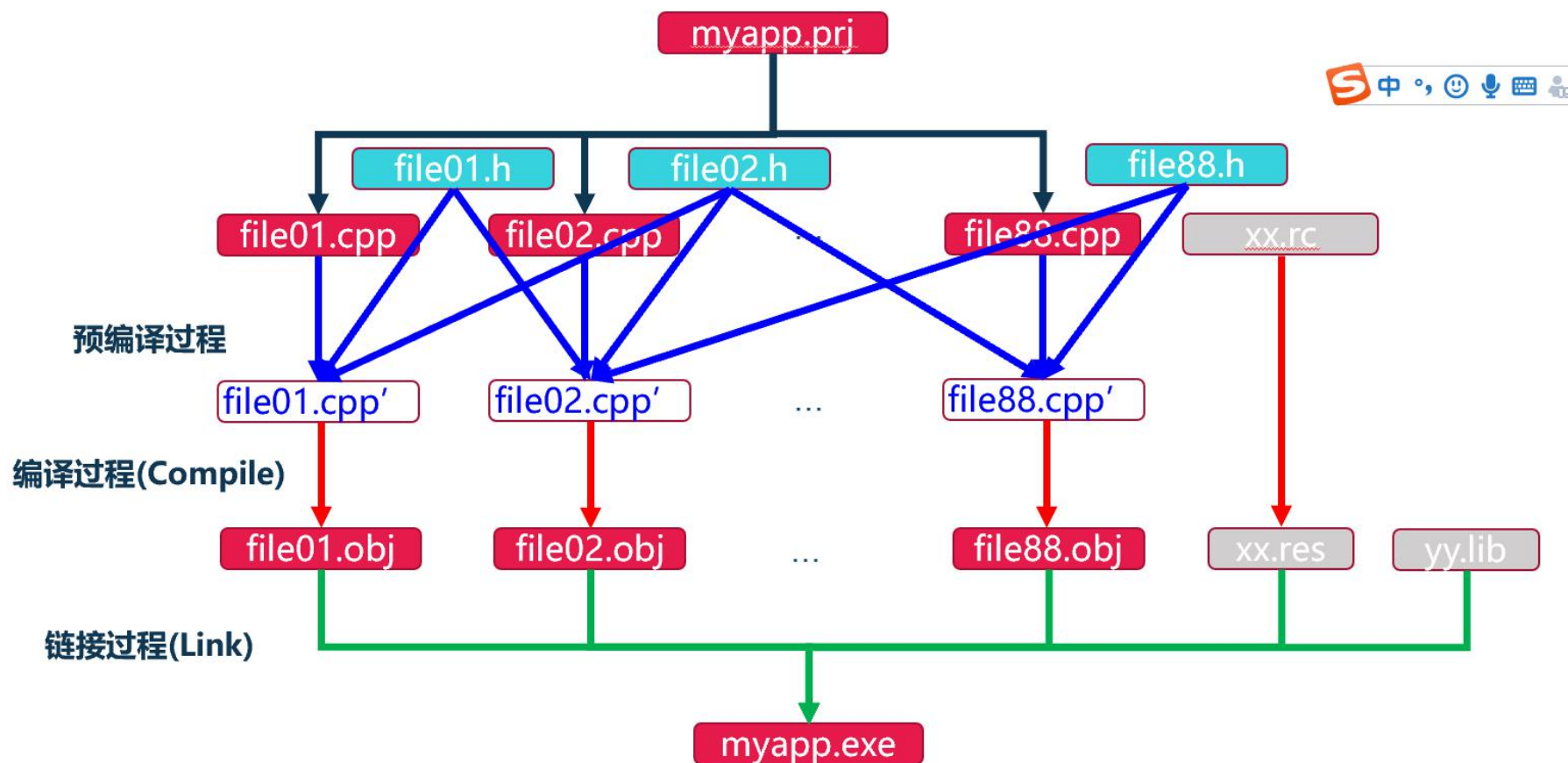
基本术语



C++程序的完整开发过程



C++程序的完整开发过程



课后作业

- 完成全部课后题
- 完成时间：下节课前



HOMEWORK