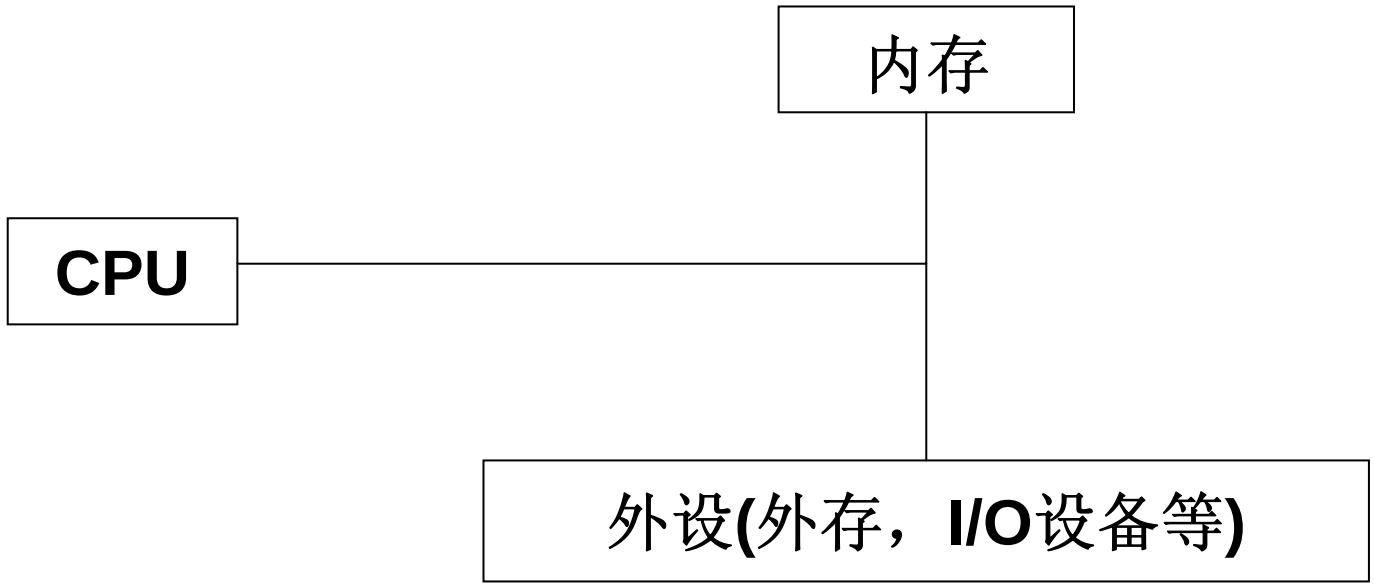
面向对象程序设计概述

主要内容

- 计算机的工作模型
- 程序设计范型
- C++语言（复习）
- 面向对象的方法

计算机的工作模型

- 冯•诺依曼（Von Neumann）体系结构

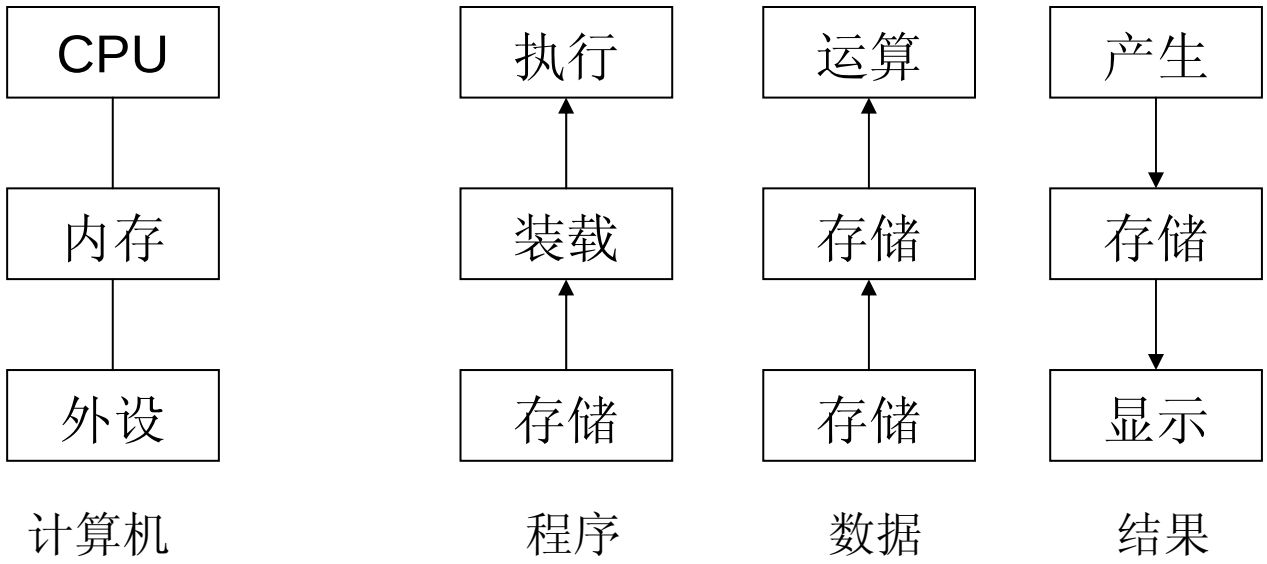


存储程序的思想

- 中央处理器（Central Processing Unit），简称CPU,用于执行计算机指令以完成计算任务。包含控制器、运算器以及寄存器等。
- 内部存储器或主存储器（Memory），简称内存,用于存储计算机程序（指令和数据）。由许多存储单元构成，每个存储单元都有一个地址。
- 外围设备（Peripheral Devices），简称外设,提供了计算机与外界的接口，包括：
  - 输入/输出设备
  - 外部存储器（外存）。

冯•诺依曼计算机的工作过程

待执行的程序从外存装入到内存中，CPU从内存中逐条地取程序中的指令执行；程序执行中所需要的数据从内存或从外设中获得，程序执行中产生的中间结果保存在内存中，程序的执行结果通过外设输出。



程序设计范型

- 程序设计本质上可归结为：
  - 数据描述
  - 数据操作（加工）描述
- 对数据和数据操作之间关系的不同处置就形成了不同的程序设计范型（Programming Paradigms）
- 典型的程序设计范型有：
  - 过程式
  - 对象式（面向对象）
  - 函数式
  - 逻辑式等

过程式程序设计

- 一种以功能为中心、基于功能分解的程序设计范型。
- 一个过程式程序由一些子程序构成，子程序描述了一系列的操作，每个子程序对应一个子功能，它实现了功能抽象。
- 过程式程序的执行过程体现为一系列的子程序调用。数据处于附属地位，它独立于子程序，在子程序调用时作为参数或全局变量传给子程序使用。

程序 = 算法 + 数据结构

对象式（面向对象） 程序设计

- 一种以数据为中心、基于数据抽象的程序设计范型，通常称为面向对象程序设计。
- 一个面向对象程序由一些对象构成，对象是由一些数据及可施于这些数据上的操作所组成的封装体。对数据的操作是通过向包含数据的对象发送消息（调用对象提供的操作）来实现。对象的特征由相应的类来描述，一个类可以从其它的类继承。
- 面向对象程序的执行过程体现为各个对象之间相互发送和处理消息。

程序 = 对象/类 + 对象/类 + …

对象/类 = 数据 + 操作

函数式与逻辑式

- 函数式程序设计是围绕函数及函数应用（Function Application）来进行，它基于了递归函数理论和lambda演算，其中，函数也被作为值来看待。
- 逻辑程序设计是把程序组织成一组事实和一组推理规则，它基于的是谓词演算（Predicate Calculus）。

  上述两种程序设计范型常用于人工智能领域的程序开发。

- 练习：将以下程序设计语言分类
- A 过程式，　B 对象式，　C 函数式或逻辑式

（1）C	A
（2）java	B
（3）Pascal	A
（4）C++	B
（5）Basic	A
（6）Fortran	A
（7）Prolog	C
（8）Lisp	C
（9）Ada	B
（10）Smalltalk	B

C++语言（复习）

- C++是一个高级语言，是贝尔实验室的Bjarne Stroustrup为支持面向对象程序设计而设计。
- C++保留了C语言的所有成分和特点，并在C语言的基础上增加了支持面向对象程序设计的语言成分。
- C++语言是C语言的超集，它既支持过程式程序设计，又支持面向对象程序设计，属于一种混合语言。
- C++语言的特点是灵活和高效。
- 国际标准化组织（ISO）已于1998年为C++制定了国际标准。

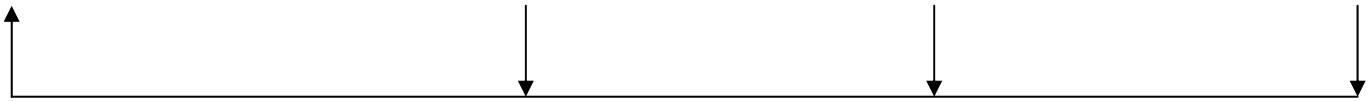
一个C++语言的例子：Hello World

C++程序的组成

- 逻辑上，一个C++程序由一些函数（子程序）、类、全局变量/对象的定义构成，其中必须有且仅有一个名字为main的函数。函数由函数名、参数和返回类型、局部变量/对象的定义以及语句序列构成；类由数据成员和成员函数构成。变量或对象的定义可以出现在函数的外部和内部，而语句只能出现在函数内部。C++程序从函数main开始执行。
- 物理上，一个C++程序可以放在一个或多个源文件（模块）中，每个源文件包含一些函数、类和外部变量/对象的定义，其中有且仅有一个文件中包含一个函数main。每个源文件可以分别编译。

C++程序的运行步骤

编辑(.cpp,.h)→编译(.obj)→联接(.exe)→运行→输出结果



- 编辑 利用某个编辑程序（如：Windows平台上的写字板、记事本、Word等）把C++源程序输入到计算机中，并作为文件（称为源文件）保存到外存（如硬盘等）中。C++源文件的文件名通常为：*.cpp和*.h。
- 编译 利用某个C++编译程序对保存在外存中的C++源程序进行编译，编译结果作为目标文件保存到外存。目标文件的文件名通常为：*.obj。
- 联接 通过一个联接程序把这些目标文件以及程序中用到的一些系统功能所在的目标文件（通常称为库文件）联接起来，作为一个可执行文件保存到外存。可执行文件的文件名通常为：*.exe。
- 运行 通过操作系统提供的应用程序运行机制，把某个可执行文件装入内存，运行其中的可执行程序。
- 在上述的编译、联接和运行过程中都有可能发现程序有错。 整个过程可能会重复多次，直到程序得出正确的运行结果。

C++集成开发环境

- 由于上述的C++程序的运行步骤非常麻烦，因此出现了很多集成的C++程序开发环境，如：Visual C++, devcpp, C++ Builder等。
- 集成环境提供了可视化的程序设计支持和功能强大的程序动态调试等工具。
- C++开发环境演示。

- 练习：写出以下程序的运行结果

```
#include <iostream>
  using namespace std;
  int count=0;
  int fib(int n)
   {  count++;
     if (n==1 || n==2) return 1;
     else return fib(n-1)+fib(n-2);
    }
  int main()
  { cout << fib(8);
    cout << ',' << count << endl;
    return 0;
  }
```

答案：21，41

面向对象的方法

面向对象方法的历史背景

- 20世纪50年代：
  - Fortran的矛盾：变量名在程序不同部分发生冲突
  - ALGOL60：首次提出封装，“begin-end”，被广泛用于Pascal，C中
- 20世纪60年代：
  - Simula67，最早提出对象的概念，鼻祖
- 20世纪70年代：
  - Ada，Smalltalk
- 20世纪80年代：广泛应用于程序设计
- 20世纪90年代：面向对象分析
  - 1997年UML（Unified Modeling Language）被OMG（Object Management Group）组织采纳为国际标准

面向对象的基本概念

- 对象：表示要研究的任何事物
  - 是由数据及其操作所构成的封装体
- 类：是对象的模板
  - 描述了一组具有相同属性和操作的对象
- 方法：指定义于某一特定类上的操作与法则。
- 消息：调用对象的操作，由对象名、消息名和实际变元构成
- 通信：指对象之间的消息传递，是引起面向对象程序进行计算的唯一方式

对象举例

- 学生：张三
  - 对象标识：对象名　student_1
  - 对象属性：
    - name= 张三
    - number=00123456
    - age　= 20
    - major=Physics
    - ……..
  - 对象操作：
    - SelectClass("class_math");
    - …….

面向对象主要特征

- 封装性
  - 把数据和实现操作的代码集中起来放在对象内部。对外部，表示对象状态的数据和实现操作的代码与局部数据都不可见，更不能从外面直接访问或修改这些数据和代码。
- 继承性
  - 继承是指能够直接获得已有的类的性质和特征，而不必重复定义它们。
- 多态性
  - 某一论域中的一个元素存在多种解释。
    - 一名多用：重载（函数、操作符）
    - 类属性：类属函数、类属类型
    - 动态绑定：虚函数

面向对象方法学

- 出发点：尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程

面向对象方法的四个要点

- 1．认为客观世界是由各种对象组成的，任何事物都是对象；复杂的对象可以由比较简单的对象以某种方式组合而成。
  - 因此，面向对象的软件系统是由各种对象组成的；
  - 任何元素都是对象，复杂的对象可以由比较简单的对象某种方式组合而成。

面向对象方法的四个要点

- 2. 把所有对象都划分成各种对象类（简称为类Class），每个对象类都定义了一组数据和一组方法。
  - 数据用于表示对象的静态属性，是对象的状态信息。
  - 每当建立该对象类的一个新实例时，就按照类中对数据的定义为这个新对象生成一组专用的数据，以便描述该对象独特的属性值。
  - 类中定义的方法，是允许施加于该类对象上的操作，是该类所有对象共享的，并不需要为每个对象都复制操作的代码。

面向对象方法的四个要点

- 3. 按照子类（或称为派生类）与父类（或称为基类）的关系，把若干个对象类组成一个层次结构的系统（也称为类等级）。
  - 派生类具有基类的特性（包括数据和方法），但是派生类又有某些基类没有的特性。
  - 这种现象称为继承（Inheritance）。

面向对象方法的四个要点

- 4. 对象彼此之间仅能通过传递消息互相联系。
  - 对象是进行处理的主体，必须发消息请求它执行它的某个操作，处理它的私有数据，而不能从外界直接对它的私有数据进行操作。
  - 也就是说，该对象的私有信息，都被封装在该对象类的定义中，这就是“封装性”。

面向对象方法学方程

- OO ＝ Objects + Classes + Inheritance + Communication with messages
- 也就是说，面向对象就是既使用对象又使用类和继承等机制，而且对象之间仅能通过传递消息实现彼此通信。

面向对象的软件工程

- 用面向对象方法学的思想进行软件开发
- 面向对象的分析（OOA－Object Oriented Analysis）
  - 其任务是了解问题域所涉及的对象、对象间的关系和作用（即操作），然后构造问题的对象模型，力争该模型能真实地反映出所要解决的“实质问题”。
  - 独立于语言
- 面向对象的设计（OOD－Object Oriented Design）
  - 即设计软件的对象模型
  - 根据所应用的面向对象软件开发环境的功能强弱不等，在对问题的对象模型的分析基础上，可能要对它进行一定的改造，但应以最少改变原问题域的对象模型为原则。然后就在软件系统内设设计各个对象、对象间的关系（如层次关系、继承关系等）、对象间的通信方式（如消息模式）等
  - 语言相关
  - 确定对象“应该做什么”
- 面向对象的实现（OOI－Object Oriented Implementation）
  - 即指软件功能的编码实现，
  - 包括：每个对象的内部功能的实现；确立对象哪一些处理能力应在哪些类中进行描述；确定并实现系统的界面、输出的形式及其它控制机理等
  - 是实现在OOD阶段所规定的各个对象所应完成的任务