机器学习程序设计基础
1. Python基础操作

陈湘萍

中山大學
SUN YAT-SEN UNIVERSITY

程序设计基础

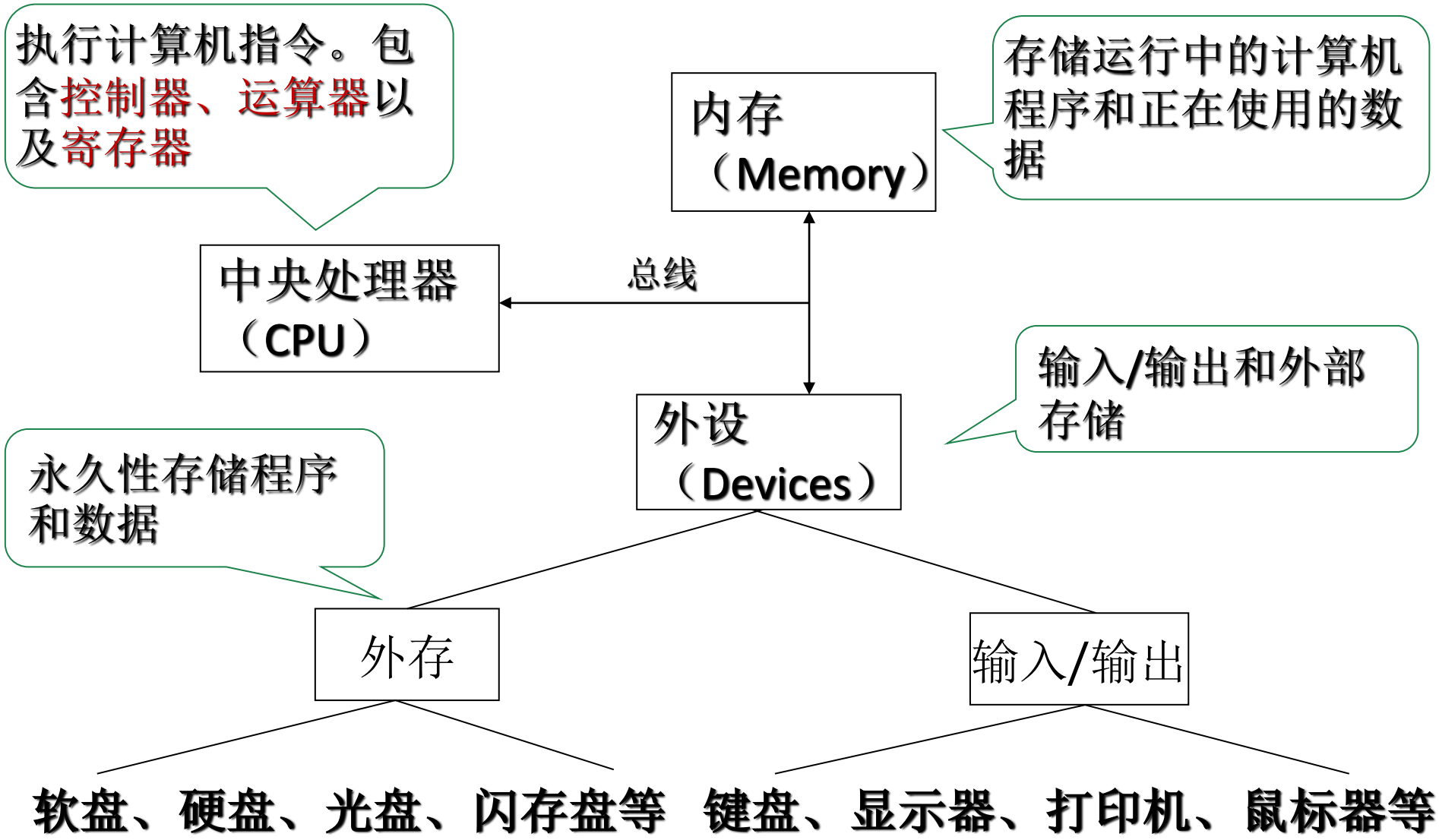
硬件与软件

- 计算机包含硬件和软件两方面
  - 硬件是指计算机的物理构成－－物质基础
  - 软件主要是指计算机程序（指令序列）－－灵魂
- 一台计算机的性能主要由硬件决定，而它的功能则主要是由软件来提供。

硬件概述

- 硬件是指构成计算机的元器件和设备
- 计算机元器件的发展经历了下面几个阶段
  - 电子管
  - 晶体管
  - 集成电路
  - 超大规模集成电路
- 计算机设备主要
  - 中央处理器
  - 内部存储器
  - 外部设备（外部存储器、输入/输出设备）

冯•诺依曼计算机的硬件设备组织



计算机能执行的指令

- 算术指令
  - 实现加、减、乘、除等基本运算。
- 比较指令
  - 比较两个操作数的大小等逻辑运算。
- 数据传输指令
  - 实现各单元之间的数据传输。
- 流程控制指令
  - 用于确定下一条指令的在存储单元中的地址。默认为顺序执行，可以是转移、循环以及子程序调用/返回等指令。
- 程序设计的任务是十分艰巨的，它要把各种应用问题落实到用一些简单的指令来解决!

软件概述

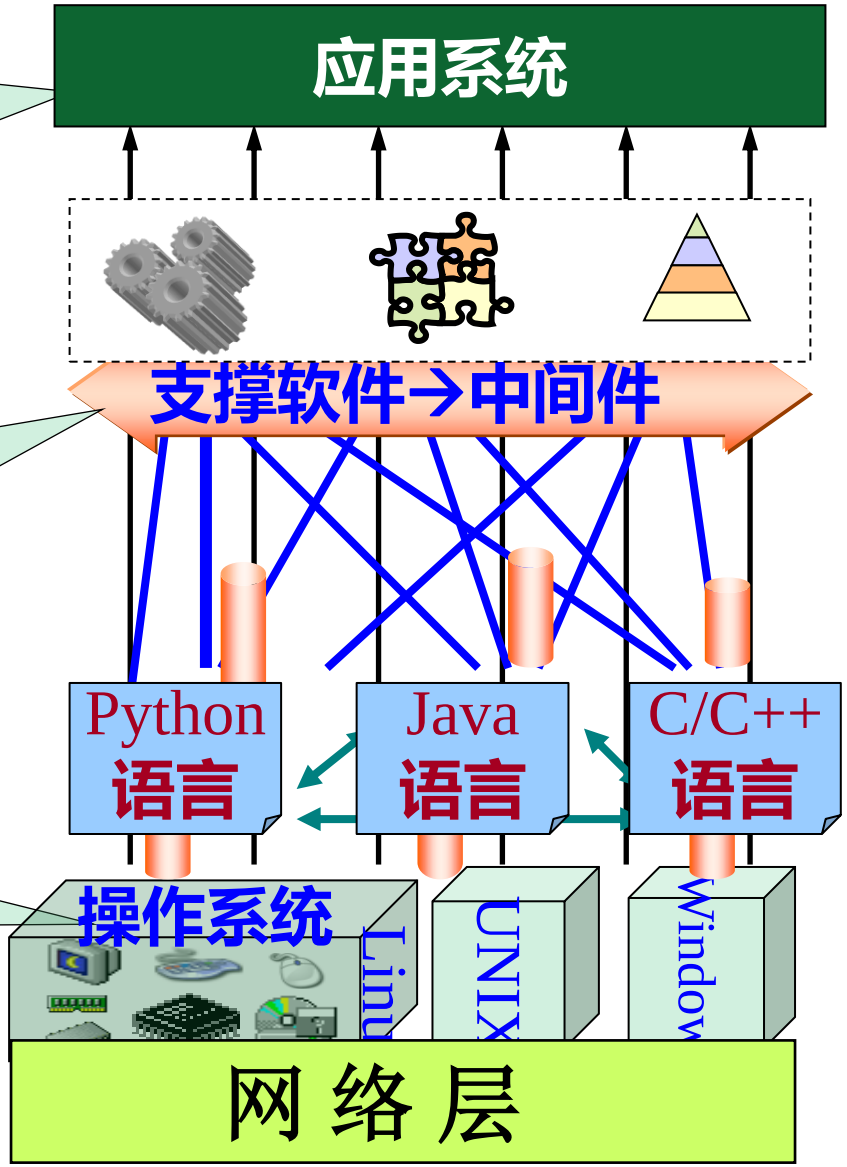
- 计算机硬件只是提供了执行存储在内存中指令的能力，而执行的指令（软件）是需要人来提供的
- 计算机软件是计算机系统中的程序以及相关的文档
  - 程序：计算任务的处理对象（数据）与处理规则（算法）的描述，处理规则体现为指令，由计算机执行
  - 文档：便于人理解程序所需的资料说明，供程序开发与维护使用

关于软件：软件技术的发展

为了弥补应用软件与现实计算环境之间的距离

为屏蔽操作系统和编程语言的异构性，发展了支撑软件和中间件

为屏蔽计算机硬件的异构性，发展了操作系统



程序设计（Programming）

- 要使得计算机能完成各种任务，就必须为它编写相应的程序
- 程序设计就是为计算机编写程序的过程
- 程序的本质

  程序 = 算法 + 数据结构

  - 算法（algorithm）是指对数据的加工步骤的描述
  - 数据结构（data structure）则是对反映待解问题本质的数据的描述

程序设计（Programming）

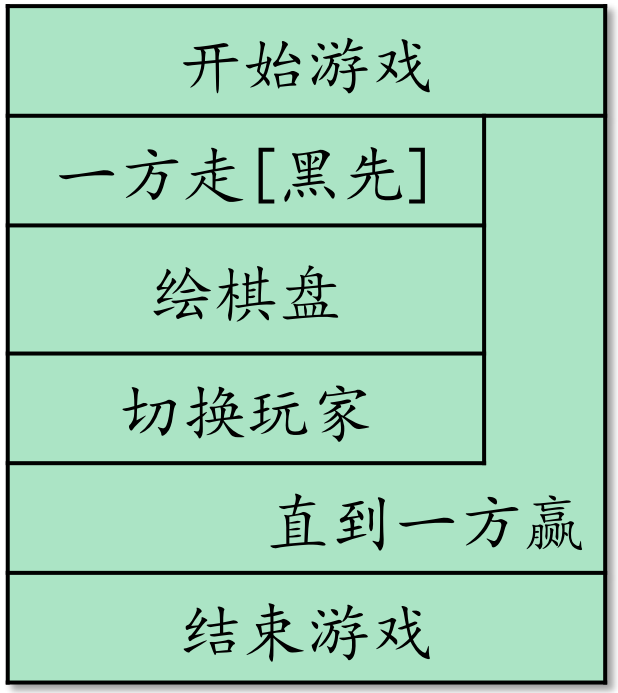
- 程序设计要涉及
  - 程序设计范式
  - 程序设计步骤
  - 程序设计语言

程序设计范式（programming paradigms）

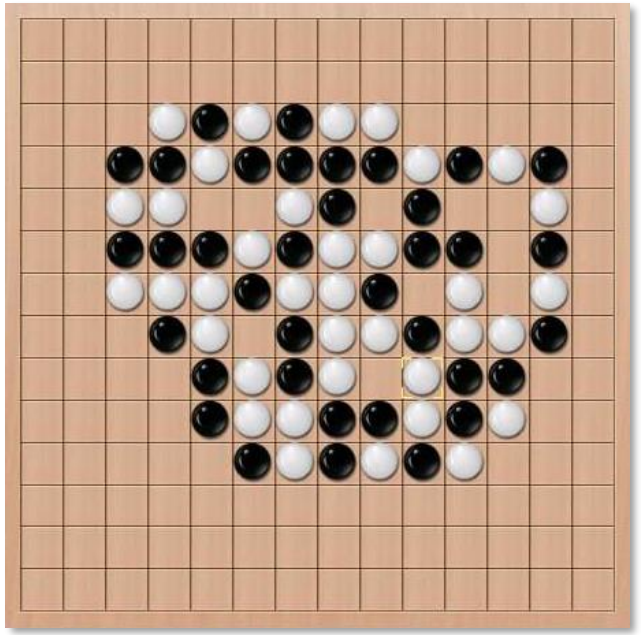
- 程序设计范式（programming paradigms）是设计、组织和编写程序的一种方式，反映了一种计算模型，它包含了一组理论、原则和概念。
- 不同的范式将采用不同的程序结构和程序元素来描述程序。
- 范式具有针对性，不同的范式往往适合于解决不同类型的问题。

面向过程设计方法

- 一种以过程为中心，自顶向下、逐步求精的结构化开发方法。
- 程序结构按照功能划分为若干个基本模块，形成一个树状（网状）结构
- 模块之间关系相对简单，在功能上相对独
- 模块内部一般都是由顺序、选择和循环3种基本结构组成



基于面向过程的设计思路

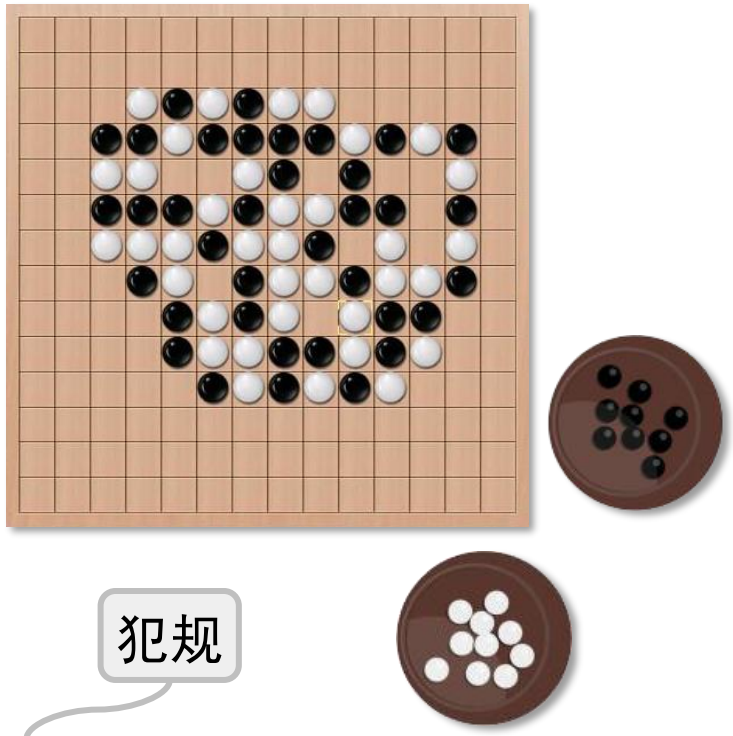


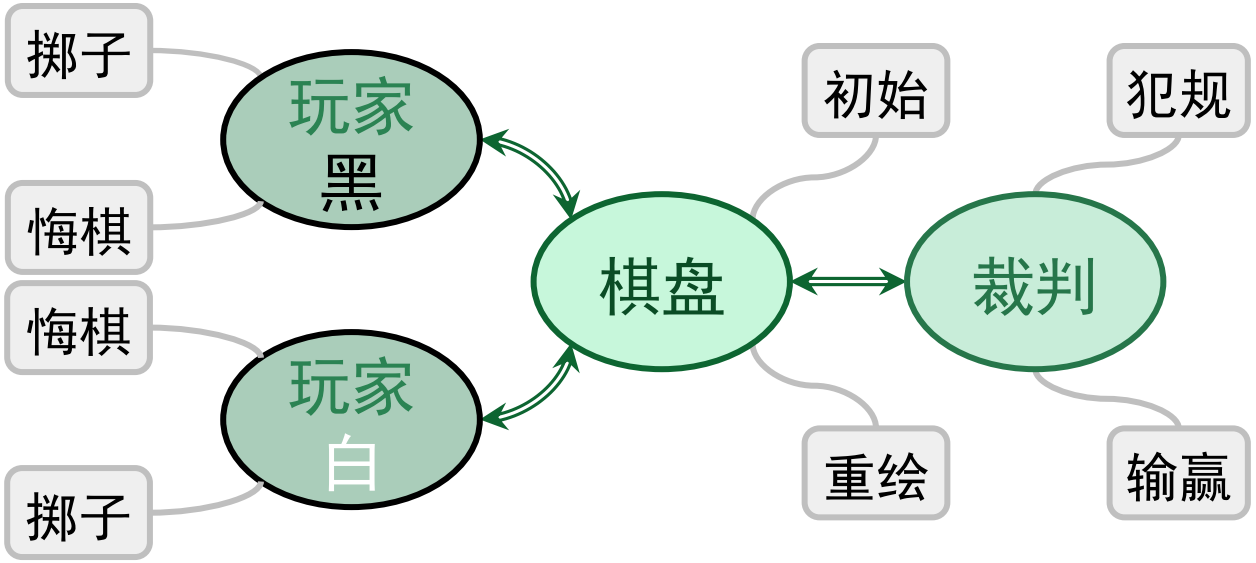
面向对象设计方法

- 面向对象设计方法把状态(数据)和行为(功能)捆绑在一起，形成对象
- 当遇到一个具体问题时，只需将一个系统分解成一个个对象，同时将状态和行为封装在对象中
  - 抽象：对一组有相同属性和相同方法对象的定义
  - 封装：隐藏类内部实现机制，仅暴露类外部接口
  - 继承：子类自动共享父类之间属性和方法的机制
  - 多态：一个对象有着多重属性和不同方法的机制

面向对象设计方法

- 玩家对象：黑白双方，两方行为相同
- 棋盘对象：负责绘制棋局
- 裁判对象：判定犯规、输赢等行为





程序设计（Programming）

- 程序设计要涉及
  - 程序设计范式
  - 程序设计步骤
  - 程序设计语言

程序设计步骤

- 明确问题
  - 搞清楚要解决的问题并给出问题的明确定义，即：做什么？
- 系统设计
  - 给出问题的解决方案，即：如何做？
  - 主要包括：
    - 数据结构的设计
    - 算法的设计
    - 如何组织上述两者，属于不同的程序设计范式。
- 实现
  - 选择用某种语言按系统设计进行编程。
  - 良好的编程风格可以通过学习和训练来获得。

程序设计步骤（续）

- 测试与调试
  - 程序可能含有错误。程序的逻辑错误和运行异常错误一般可以通过测试（test）来发现。
  - 对错误进行定位的过程称为调试（debug）。
- 运行维护
  - 所有的测试手段只能发现程序有错误，而不能证明程序没有错误!
  - 在程序使用中发现错误并改错称为维护，包括:
    - 正确性
    - 完善性
    - 适应性

程序设计（Programming）

- 程序设计要涉及
  - 程序设计范式
  - 程序设计步骤
  - 程序设计语言

程序设计语言

- 程序设计语言是一种用于书写计算机程序的语言。
  - 语言的基础是一组记号和一组规则。
  - 根据规则由记号构成的记号串的总体就是语言。
- 在程序设计语言中，这些记号串就是程序。

程序设计语言

- 根据与计算机指令系统和人们解决问题所采用的描述语言（如：数学语言）的接近程度，常常把程序语言分为：
  - 低级语言
  - 高级语言
- 通常所讲的程序设计语言往往指的是高级语言。

低级语言

- 低级语言是指特定计算机能够直接理解的语言（或与之直接对应的语言）
  - 机器语言：采用指令编码和数据的存储位置来表示操作以及操作数
  - 汇编语言：是用符号名来表示操作和操作数位置，以增加程序的易读性。需要翻译（汇编）成机器语言才能执行。

高级语言

- 高级语言是指人容易理解和有利于人对解题过程进行描述的程序语言。
  - 典型的高级语言有：FORTRAN、COBOL、Basic、Pascal、C、Ada、Modula-2、Lisp、Prolog、Simula、Smalltalk、C++、Java、Python等
  - 需要翻译成机器或汇编语言才能执行。

程序设计语言

机器语言

```
00100011
11101101
01100001
01110110
```

- 指令对应机器基本动作
- 占用内存少、执行效率高
- 通用性低、移植性差

汇编语言

```
MOV   AX, 300H
ADD   BX,  AX
MOV  [2100H], BX
HLT
```

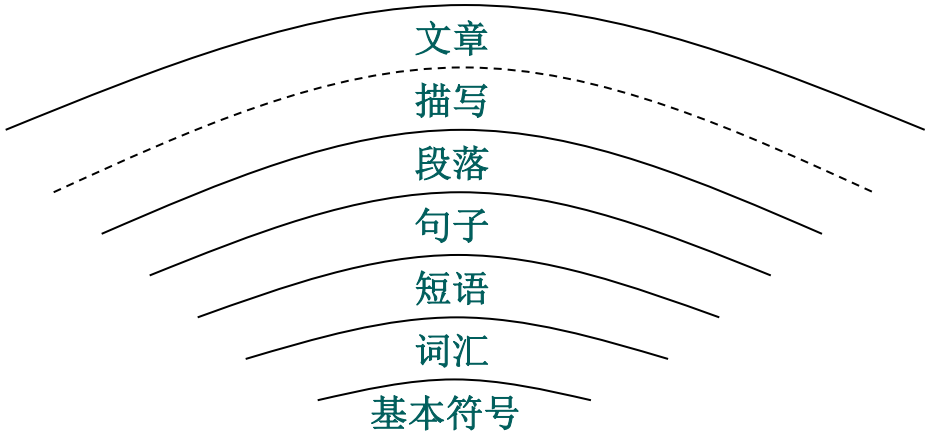
- 使用助记符号代替指令
- 代码简短、易于识记
- 开发效率低、周期长

高级语言

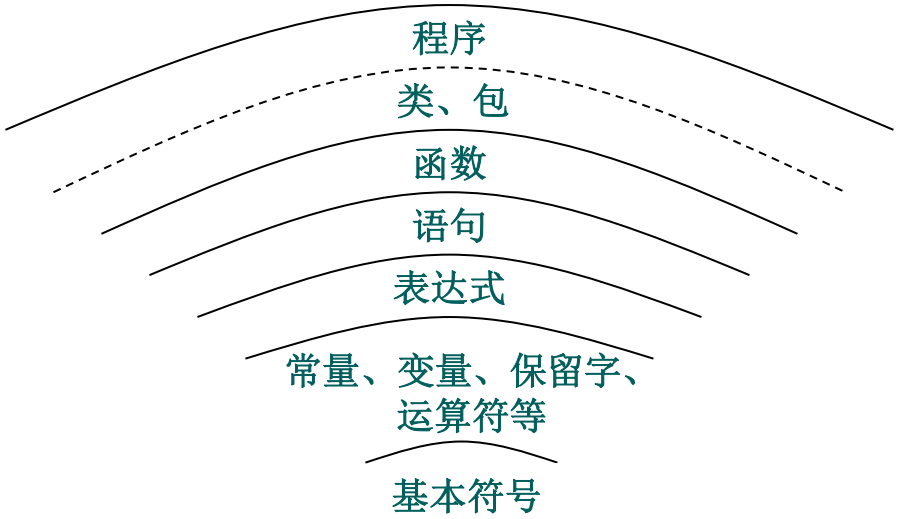
```
main( ) {
    int a, b, c;
    a=300;  b=18;
    c=a+b;
    printf("c= %d\n", c);}
```

- 接近于人类自然语言
- 简单直观、通用性强
- 交互性好，易于调试
- 执行效率低，目标代码大

程序设计语言



语言的基本体系结构



程序设计语言的基本体系结构

计算机语言的体系结构与语言的体系结构完全一致，只是每一个层次的内容要简单得多。

程序设计语言

- 语言的设计是指给出语言的定义，包括语言的语法、语义和语用等。
  - 语法：是指构作结构正确的语言成分所需遵循的规则集合
  - 语义：是指语言各个成分的含义
  - 语用：是指语言成分的使用场合及所产生的实际效果
- 语言的实现是指在某种计算机平台上写出语言的翻译程序
  - 针对某种语言可以有多种实现

源码执行方式

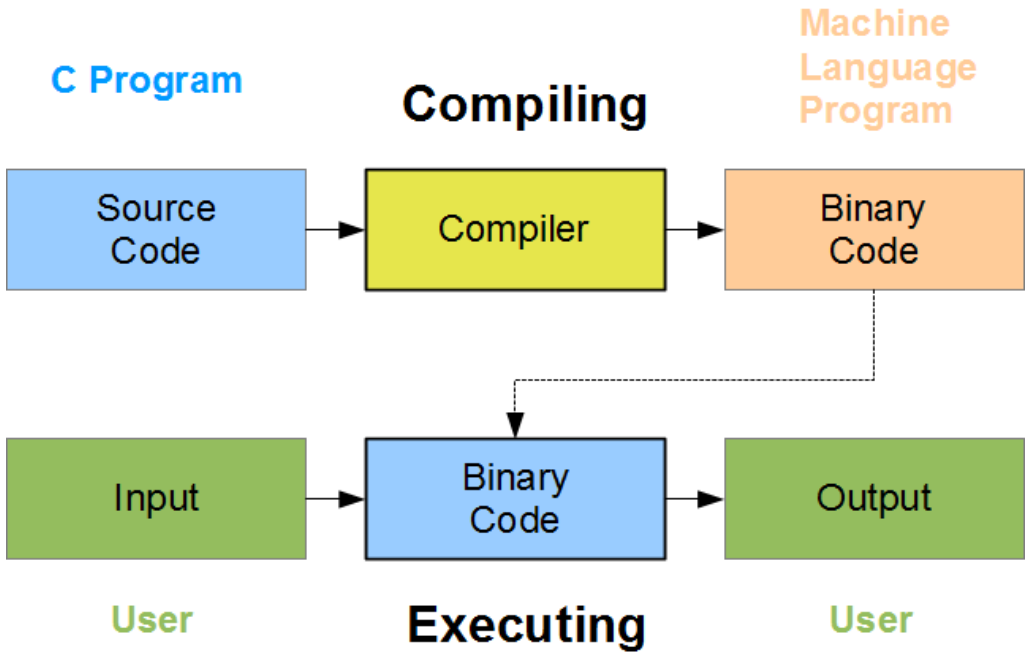
高级语言所编写的源代码最终都要以某种方式转换成能被计算机识别的0,1序列机器码，方可被计算机接受并执行。

- 编译执行
- 解释执行

源码执行方式

- 编译
  - 把高级语言程序（称为源程序）首先翻译成功能上等价的机器语言程序（称为目标代码程序）或汇编语言程序（再通过汇编程序把它翻译成目标代码程序）
  - 在目标代码程序的执行中不再需要源程序
  - 翻译工作由编译程序完成

◆ 编译执行时检查词法、语法代码优化、生成可执行文件等需要花费时间，编译后的机器码具有平台相关性，运行速度快



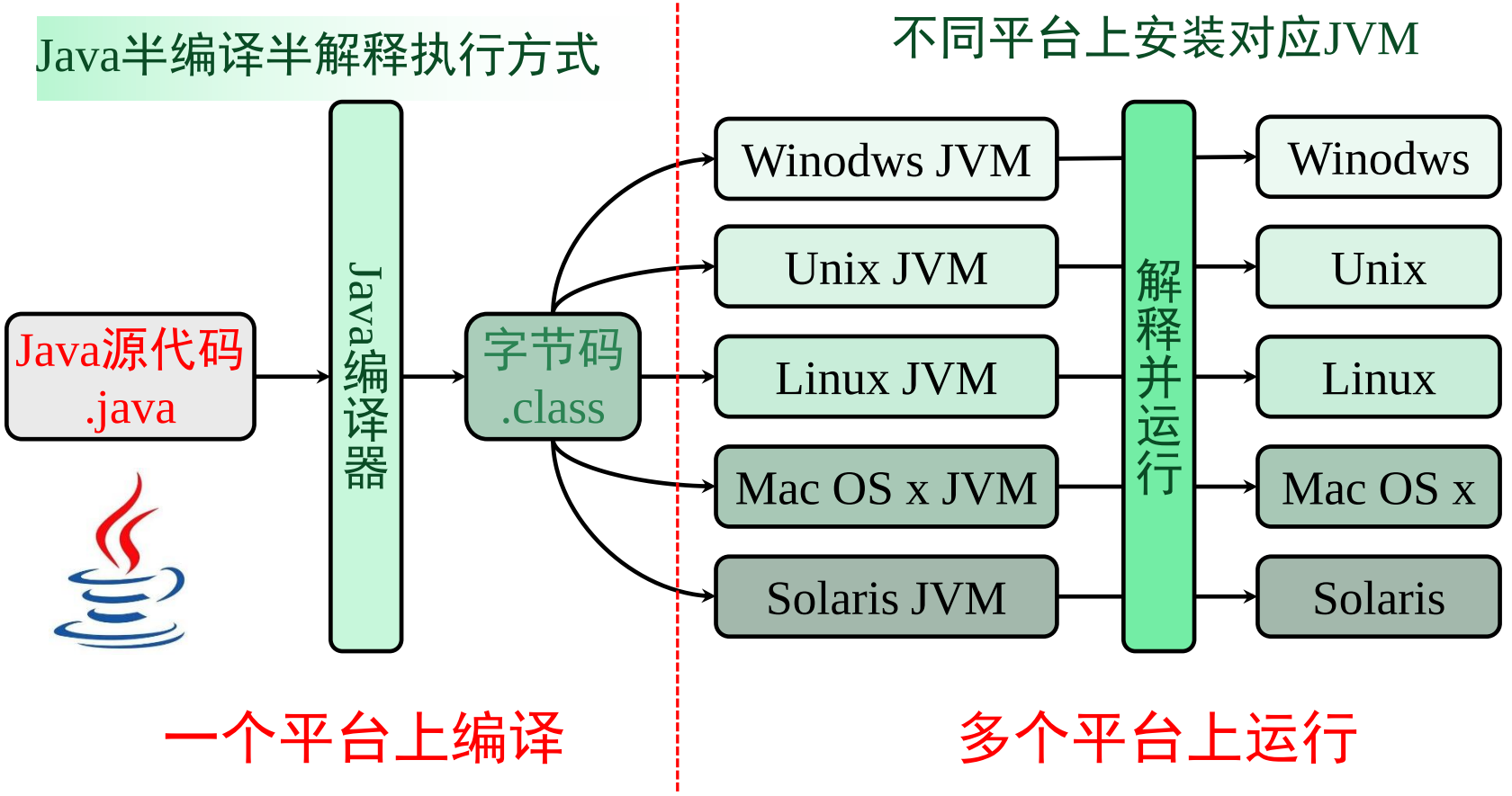
源码执行方式

- 解释
  - 对源程序中的语句进行逐条翻译成目标代码并执行，翻译完了程序也就执行完了
  - 这种翻译方式不产生目标代码程序，程序的每次执行都需要源程序
  - 翻译工作由解释程序完成

◆ 解释执行翻译一条执行一条，不生成目标程序，浪费计算机资源，效率低，但是不依赖于平台。

源码执行方式

- 编译型与解释型各有优缺点又相互对立，所以一些语言把两者折衷结合起来，形成了一种半编译、半解释的执行方式。



Python与Java执行方式类似。不同在于Python的编译通常发生在对某个模块的调用过程中。模块编译成字节码可以节省加载时间，以达到提高效率的目的

Python程序设计语言

- 简单
- 高级
- 面向对象
- 可扩展
- 免费和开源
- 可移植
- 丰富的库
- 丰富的接口

- Python的设计哲学是明确、简单、优雅。
- Python关键字少，结构简单，语法清晰，易读，易维护。
- 学习Python可以在短时间轻松上手。
- Python使用缩进格式。

- Python是高级语言，内置高级数据结构，程序员无需关心底层（如内存分配与回收），可以更高效地专著于问题本身。

- Python支持面向过程的编程，也支持面向对象的编程，还是一[illegible]程语言。

- Python提供丰富的API和工具，以便程序员能够轻松地使用C、C++语言来编写扩充模块，因此又被称之为“胶水语言”

Python程序设计语言

- 简单
- 高级
- 面向对象
- 可扩展
- 免费和开源
- 可移植
- 丰富的库
- 丰富的接口

- Python是自由/开放源码软件，允许自由地发布此软件的拷贝，阅读和修改其代码，或将其中一部分用于新的自由软件中。

- Python程序可以在Unix/Linux，Windows，Macintosh等不同主平台上运行。

- Python语言提供功能丰富的标准库，如网络、文件、数据库、图形界面、正则表达式、文档生成、单元测试等。
- 拥有大量的第三方库。如科学计算库NumPy，SciPy，Matplotlib等。

- Python提供面向其它系统和专用库的接口，如数据库管理系统、计算机视觉库OpenCV、三维可视化库VTK、医学图像处理库ITK等。还可以将Python嵌入C、C++程序中。

Q&A

谢谢大家!