编译原理基础:语言处理器

1编译器和解释器的区别是什么？

：编译器就是将某种语言编写的程序代码编译成另一种可供所有机器执行的机器代码。解释器指的是将用户程序代码直接通过解释并得到输出，没有得到中间的目标程序。

例如java语言处理器，结合了编译器和处理器，这样java编译器先将程序编译成中间表示形式的字节码形式，然后再由解释器解释输出。好处是一处编译，随处解释执行。（通过网络可以进行机器迁移）

词法分析—>语法分析—>语义分析—>中间代码优化器

用户输入程序代码—>编译器—>目标程序—>运行输出

用户输入程序代码—>解释器—>运行的输出

2程序设计语言基础

静态和动态的区别：如果一个编译器对一个程序支持静态解决某个问题，那么就说这个语言使用了静态策略，或者说这个问题可以在编译时刻决定。一个只允许在运行时刻才做出的决定是动态策略。

声明的作用域，比如x的变量在作用域A，那么A中对x的引用都指向这个声明，即每个对象的实例，如果不是类变量，那么都会创建一个变量x，并且A中的引用都指向x,如果是类变量，类只会在初始化类的时候为变量x分配一次内存空间，所以static关键字的意思不是静态作用域，but编译器确定用于存放这个被声明变量的内存位置的能力。如果private static int x;那么每个这个类的实例都只存在一个存放这个x的内存位置的拷贝，而如果不是类变量，那么每一个实例都有一个内存位置来存放不同的x，编译器在程序运行之前无法确定这个内存位置。

环境和状态。