

0、前言：平台与软件

0.1 什么是平台

0.2 本课程的小目标

0.3 学习本课程的好处

0.4 本课程的内容大纲

0、前言：平台与软件

0.1 什么是平台

任何软件的开发和运行，都需要基于某种平台。

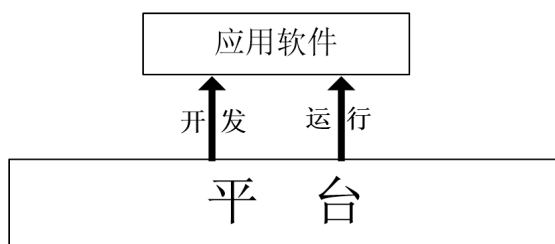


图1、平台与软件

编程平台包括很多种类，按层次从低到高包括：计算机硬件、操作系统、中间件、SDK、应用框架等。

0.2 本课程的小目标

•本课程的目标：

•学习某种计算机编程语言？

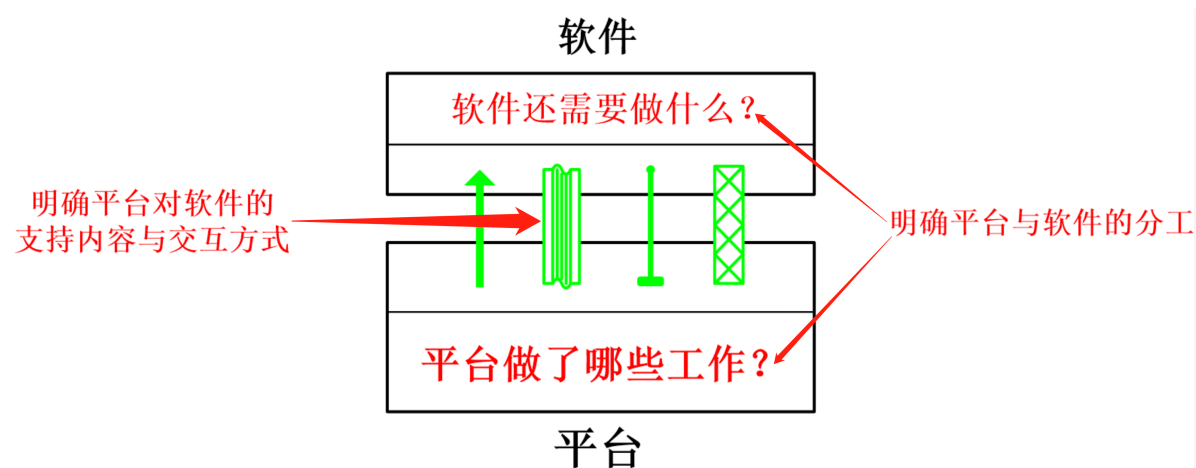
•学习某种编程平台？

•都不是！而是去研究“平台与软件的关系”后者“平台对软件开发的支持”。

•当我们面对一个新平台时，要了解两个东西：

•平台自己做了什么？ 还剩下什么需要我们程序员去做？

平台对应用程序开发的提供了哪些支持？ 内容和方式是什么？



0.3 学习本课程的好处

为达到本课程的目的，我们将——走过那些最典型最重要的平台，调查各个平台是如何支持编程的，从中得出共性的东西。这是带你掌握编程的“**底层逻辑**”！

掌握编程底层逻辑的好处是：当你需要在一个新的平台上软件开发时，可以迅速掌握它。

0.4 本课程的内容大纲

•囊括了六大类平台，以及平台对编程的支持特点。

基于平台的软件开发-课程大纲

- 1、计算机编程前传
 - 1.1 差分机-程序设计思想开始萌芽
 - 1.2 分析机-第一台真正意义上的 Computer
 - 1.3 制表机-打孔时编程代的到来
 - 1.4 从二进制到逻辑门
 - 1.5 图灵机
 - 2、基于CPU指令集的编程
 - 2.1 冯诺依曼构架
 - 2.2 机器语言编程
 - 3、基于操作系统的编程
 - 3.1 操作系统发展史
 - 3.2 操作系统的功能
 - 3.3 Windows API编程
 - 3.3.1 事件驱动编程模式
 - 3.3.2 MVC编程模式
 - 3.3.3 Windows动态链接库
 - 4、基于系统基础设施的编程
 - 4.1 图形用户界面编程
 - 4.2 Java网络编程
 - 5、基于面向对象应用框架的编程
 - 5.1 基于面向对象应用框架编程的特点
 - 5.2 基于面向对象应用框架编程举例-MFC
 - 6、基于组件应用框架的编程
 - 6.1 基于组件式框架编程的特点
 - 6.2 基于组件式框架编程举例1-Unity3D
 - 6.2 基于组件式框架编程举例2-Springboot
-