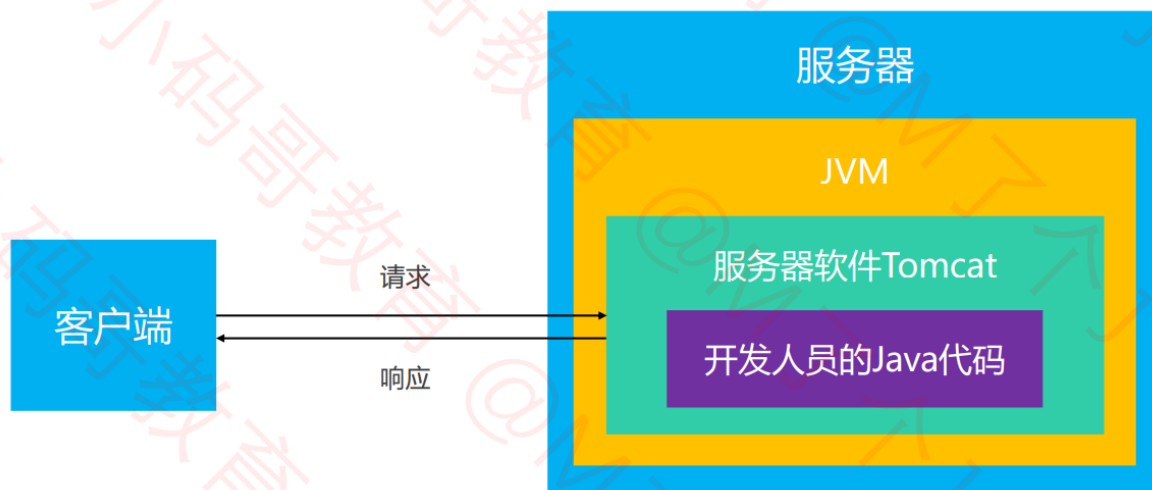


1. 基本概念

1.1 客户端/服务器

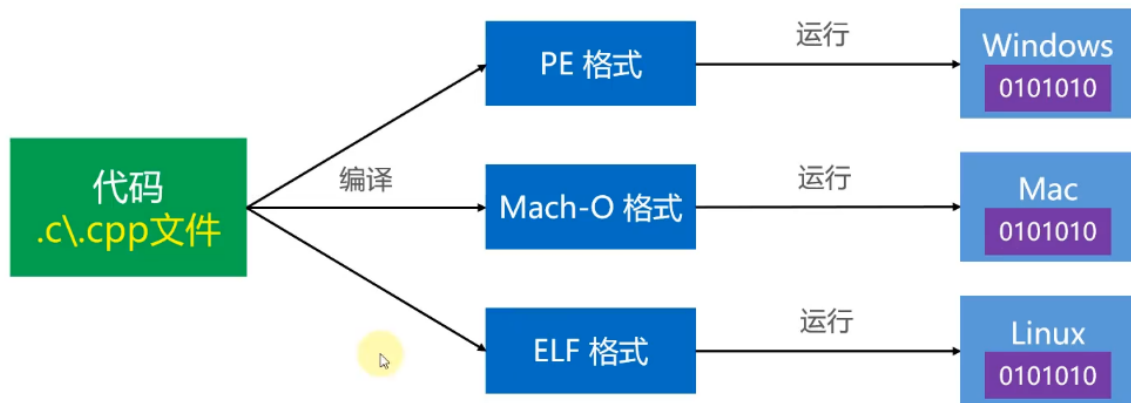


一般客户端会向服务器的一个端口发送数据, 这个端口就需要一个软件来监管该端口的数据



1.2 跨平台

C/C++跨平台



■ C/C++的跨平台：使用平台相关的编译器生成对应平台的可执行文件

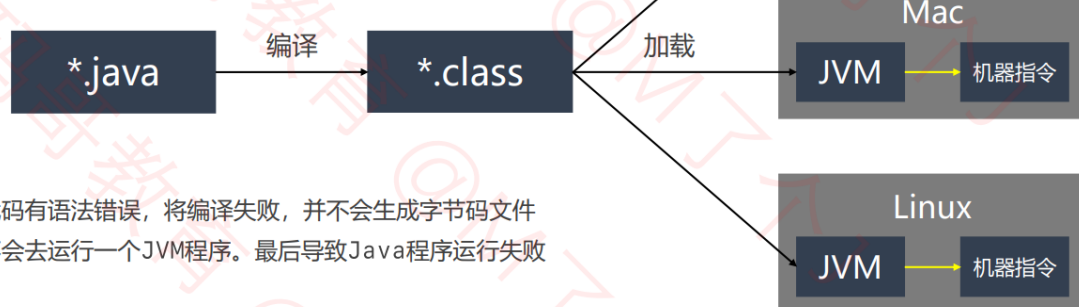
Java跨平台

■ JVM (Java Virtual Machine) : Java虚拟机

■ Java的跨平台：一次编译，到处运行

□ 编译生成跟平台无关的字节码文件 (class文件)

□ 由对应平台的JVM解析字节码为机器指令 (010101)



■ 如果代码有语法错误，将编译失败，并不会生成字节码文件

□ 那就不会去运行一个JVM程序。最后导致Java程序运行失败

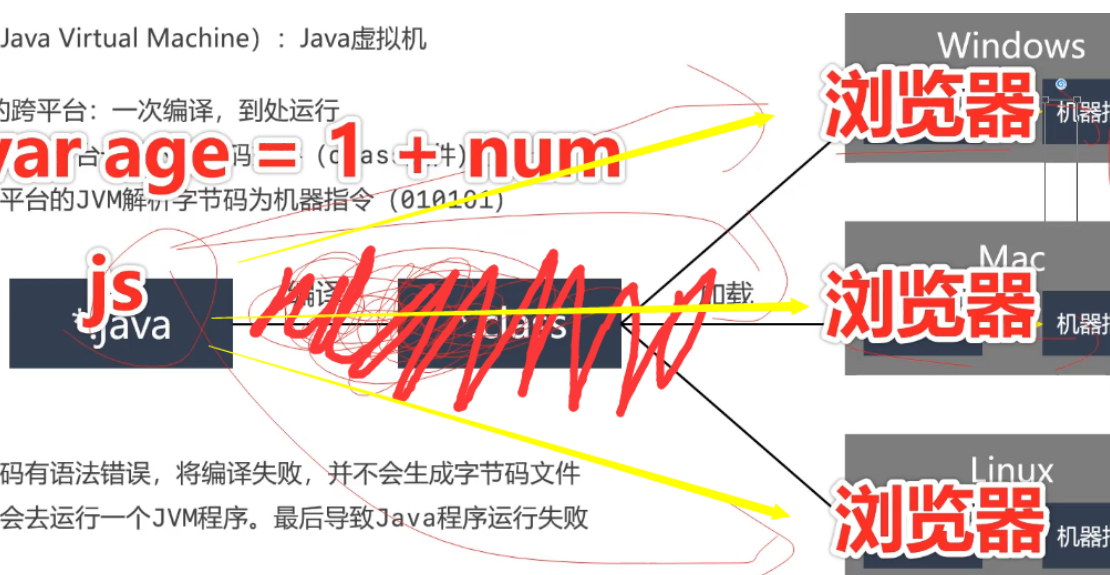
JavaScript跨平台

JVM (Java Virtual Machine) : Java虚拟机

Java的跨平台：一次编译，到处运行

编译 **var age = 1 + num** 代码 (class文件)

由对应平台的JVM解析字节码为机器指令 (010101)



如果代码有语法错误，将编译失败，并不会生成字节码文件

那就不会去运行一个JVM程序。最后导致Java程序运行失败

1.3 网络互联模型

- 为了更好地促进互联网络的研究和发展，国际标准化组织 ISO 在 1985 年制定了网络互连模型
- OSI 参考模型（Open System Interconnect Reference Model），具有 7 层结构

7	应用层（Application）	4	应用层（Application）	5	应用层（Application）
6	表示层（Presentation）	3	运输层（Transport）	4	运输层（Transport）
5	会话层（Session）	2	网际层（Internet）	3	网络层（Network）
4	运输层（Transport）	1	网络接口层（Network Access）	2	数据链路层（Data Link）
3	网络层（Network）			1	物理层（Physical）
2	数据链路层（Data Link）				
1	物理层（Physical）				

1.4 请求过程

