

基础（1）类装载器？字节码校验器？解释器？java虚拟机？

1.java虚拟机：内存分布（堆/栈），垃圾收集 常量池？程序第一次使用某个字符串直接量（定义时赋值的数值）时，java会使用常量池来缓存该字符串直接量，后续需要用时，直接使用常量池。常量池是在编译器被确定并被保存在.class文件中的数据。2.泛型

基础（2）1.程序执行过程？代码加载到内存中后？2.浮点数的精度？

- 1.java浮点数有固定的表数范围和字段长度，与机器无关，遵循IEEE754原则；
 - 2.浮点型数据的取值范围和精度：浮点型数据存储在内存中时，分为：符号位S，指数位E、尾数位M；
- float型数据：

符号位(S)：最高位(31位)为符号位，表示整个浮点数的正负，0为正，1为负 指数位(E)：23-30位共8位为指数位，这里指数的底数规定为2。并且指数位是以补码的形式来划分的(最高位为指数位的符号位，0为正，1为负)。另外，标准中还规定了，当指数位8位全0或全1的时候，浮点数为非正规形式(0.M)，所以指数位真正范围为：-126~127。尾数位(M)：0-22位共23位为尾数位，表示小数部分的尾数，即形式为1.M或0.M，至于什么时候是1 什么时候是0，则由指数和尾数共同决定。小数部分最高有效位是1的数被称为正规(规格化)形式。小数部分最高有效位是0的数被称为非正规(非规格化)形式，其他情况是特殊值。说明：float型数据 = $(-1)^s * 2^{(e-127)} * (1.m)$ 或者 $(0.m)$ ，即非正规和正规，当指数位8位全0或全1的时候，浮点数为非正规形式(0.M)。

- float取值范围：由上述等式，尾数位正规式(1.m)中是大于1小于2的，非正规式(0.m)中是大于0小于1的，所以，float型数最大值一定小于 $(2^{127}) * 2$ ，即 2^{128} （8位指数位，最高位为0，其余都是1，这时适合规范式的标准，所以乘以规范式的最大值2）；同理，最小值大于 -2^{128} ；所以取值范围为： **$(-2^{128}, 2^{128})$** 。
- float的精度：23位的尾数位，所代表的数值为 $2^{23} = 8\ 388\ 608$ ，共7位，因为1.m或者0.m；所以6位小数必精确，7位部分精确，精确范围为6到7位。
- double是11位指数位，52位尾数位，所以精度为15-16位小数。
- java默认是double；如果希望使用float类型就要在浮点数后加f或者F；与long类型同理 3.基本类型转换？自动类型转换和强制类型转换。数值类型（整型、字符、浮点）和布尔类型之间不能转换
- 自动类型转换

- 1.系统会自动把表数范围小的直接赋值给表数范围大的（小->大）；
- 2.[char、(byte->short)] -> int -> long -> float -> double（整型->浮点）；
- 3.char、(byte->short)之间不会相互转换，因为计算时会先自动转换成int；
- 4.如果想把基本数据类型转换成字符串类型，就用+“空字符串”。

- 强制类型转换

书写格式：（数据类型）value；如果大变小，则是从低位开始截，而且是要补码转原码才能知道取值；
小知识：随机生成的验证字符串，设计原理：先随机生成一个指定范围的int数字（ASCII码字母大小写的取值范围），然后强制转换成char类型，再将多次生成的字符串起来。包装类：每个包装类都提供一个parseXXX(String str)的静态方法将字符串转换成基本数据类型。boolean -> Boolean / byte -> Byte / short -> Short / **int -> Integer** / long -> Long / **char -> Character** / float -> Float / double -> Double
表达式运算中，会把所有数据类型转换成表达式中最高那个在进行运算，其计算值也是表数范围最高数据的类型；如：

```
Short S = 1;
S = S + 2;
```

这样是错误的，表达式(s+2),因为java默认整型数据是int类型，所以运算值为int类型，转换成short类型会出错。注意整型运算和浮点运算中的默认类型这个设定

4.逻辑与或，短路与或，需要多做习题 5.增强for循环与迭代器

基础（3）面向对象与面向过程的区别？

1. 面向过程：

(1).又名结构化设计，按功能分析系统需求；遵循自顶向下、模块化设计等原则；(2).把系统按功能细分，每个功能要完成一定的数据处理，其设计的每个功能都负责对数据进行一次处理，形成数据流；(3).最小程序单元为函数；(4).局限性：1.设计不够直观；和人的习惯思维不一样。2.适应性差，可扩展性不强；当需要改变时，需要自顶向下的修改模块结构。

2. 面向对象：

(1).基本思想：使用类、对象、继承、封装、消息等基本概念进行设计；以现实世界中的事物（即对象）为中心进行思考；(2).系统由类组成，类抽象了客观事物的本质特点；事物的状态为类的属性（property，成员变量），事物的行为为类的方法；(3).最小的程序单元为类；(4).基本特征有：封装、继承和多态；

- 封装：隐藏对象的实现细节，通过公用方法暴露其功能；
- 继承：子类直接获得父类的属性和方法；提高了内聚性，降低了对象之间的耦合性。
- 多态：父类引用指向子类对象，运行时表现子类的行为。(5).对象拥有：标识唯一性、分类型、多态性、封装性、模块独立性好的特点；(6).消息是指对象之间相互合作，实例于实例之间通信的机制；判断一门语言是否面向对象，通常看是否使用继承和多态加以判断；

• 总结答案：

1. 出发点不同；面向过程是根据功能对系统需求进行结构化分析与设计，强调把解决问题的过程抽象化为模块，以过程为中心处理问题；而面向对象是以现实世界的事物（即对象）为中心，把客观事物抽象为类的处理问题；
2. 面向过程的最小程序单元为函数，系统由函数组成，自顶向下的进行模块化设计，通过函数对数据进行处理的过程来实现问题解决的过程；面向对象的最小程序单元为类，根据业务对事物的状态和行为的要求，定义类的属性和方法；创建对象，描述事物在解决问题步骤中的行为去实现解决问题的过程。
3. 面向过程的优点在于性能比面向对象高，因为不需要实例化，缺点是扩展性差、设计不直观与人的习惯思维不相同；面向对象的优点在于，由于具有继承、多态和封装的特性，使系统扩展性高、容易维护和低耦合，缺点在于性能较低。