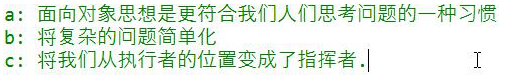
1 只是在控制台输出的话函数返回值为void。

2 面向对象思想的好处（面试题）：



3 对象是类的实例。类是属性和方法的集合。

4 属性是成员变量，方法是成员方法。

5 main函数中叫局部变量，类中方法外叫成员变量。

6 成员方法和普通的方法相比去掉了static。

7 局部变量在方法中或者方法的声明上（即形参）。

8 局部变量必须先赋值再使用，成员变量有初始化值，可以不赋值就使用。

9 不带main主方法的类是基本类，带main主方法的是测试类。

10 直接打印对象输出的是地址值。

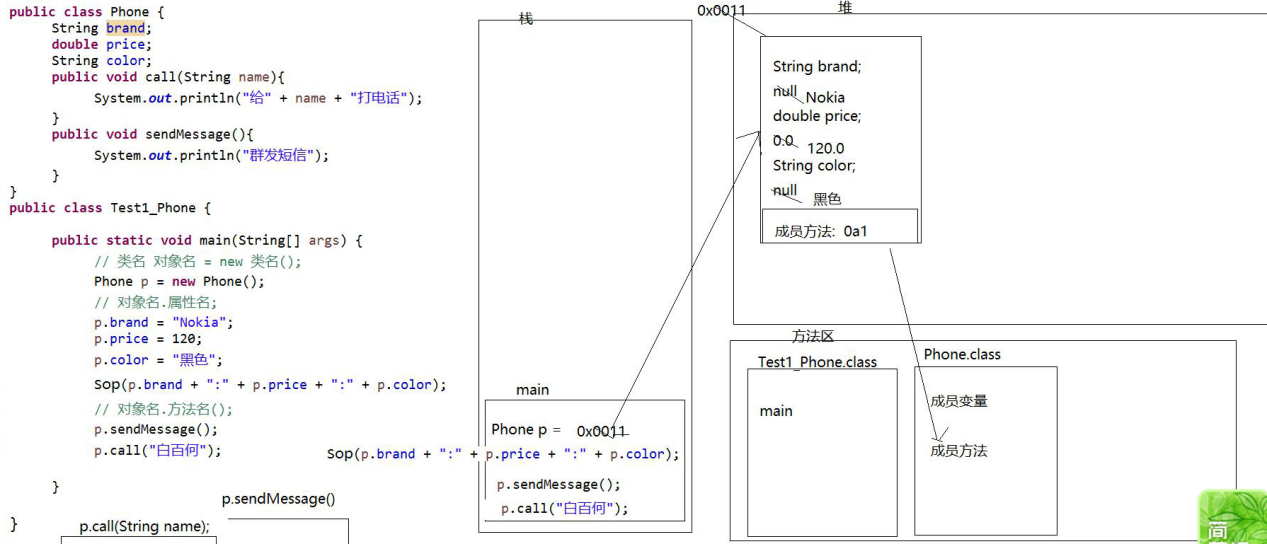
11 两个类不在同一个包里就要导入包。

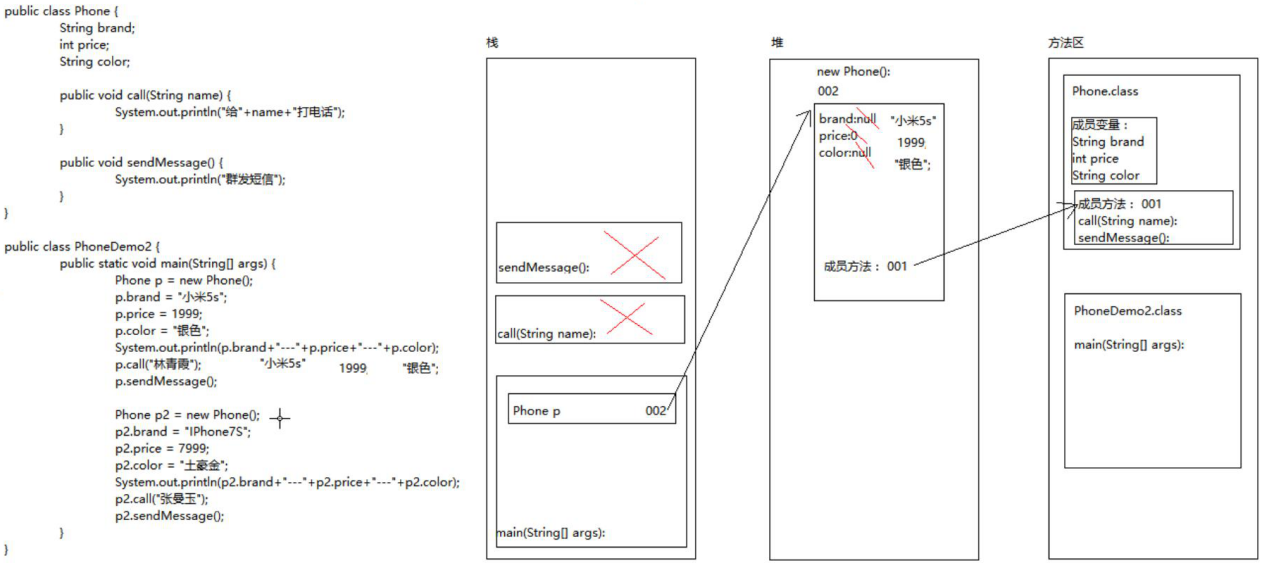
12 调用方法和属性都要在前面加上实例名。

13 字符串记得加双引号。

14 成员方法还可以传入参数。

15 对象的内存图：（要求会画）





注意，方法区里面是编译后的class文件，class文件里面有所有的属性和方法；栈内存中是main方法以及class里面的各种方法。当main方法new创建新对象时，堆就会复制一份方法区中对应的对象。当main方法调用对象中的方法或设置属性时，方法区中的方法会进入栈内存并且被执行方法或改变属性。

new的动作会在堆内存开辟空间创建对象。

栈是执行到哪个方法哪个方法就进栈。

**堆中的成员变量是从方法区复制的，而堆中的成员方法是引用自方法区，是指向方法区的。**

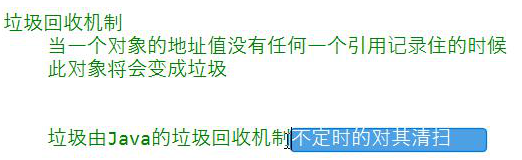
**堆中的成员变量会先进行默认初始化。**

15 在多个class文件时，只会进入方法区一次。一次性全部进入。

15 替换字符串用new string.replace(老字符串，新字符串)

16 栈内存执行完毕立即回收，堆内存在垃圾处理器空闲时回收。

17 垃圾回收机制：（当堆中的内容没有栈引用时，堆中的内容就成了垃圾）

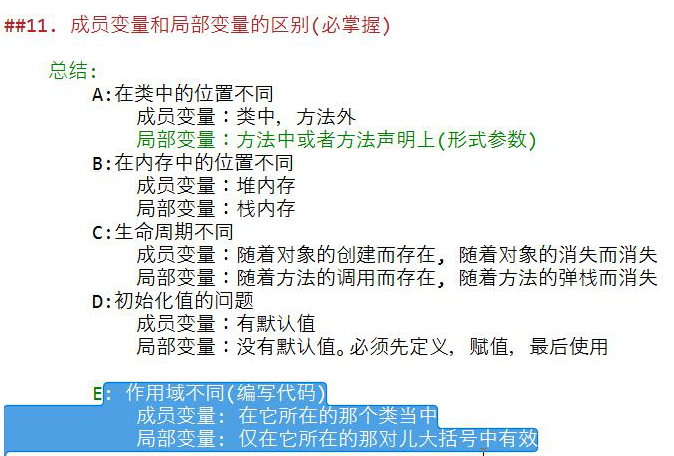


18 不同的变量可以指向相同的堆内存，因此他们可以互相覆盖。

19 局部变量没有默认初始化值，使用之前必须赋值。

20 成员变量在堆内存，局部变量在栈内存。前者随所在对象产生和消失，后者随所在方法产生和消失。

21 成员变量和局部变量的区别（注意对象里面有方法，方法里面有局部变量）：



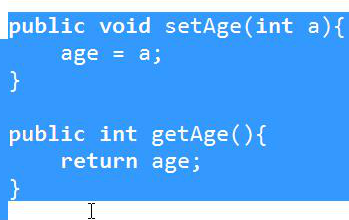
22 private用来修饰成员。他修饰的成员只能在本类使用。作用是不让外界的对象直接访问成员变量，防止出现安全问题。



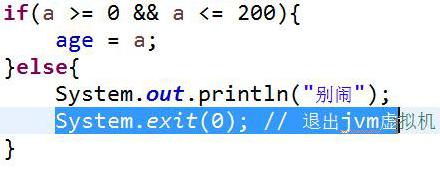
23 同一个包中的类可以互相引用而不用导包。

24 字符串默认是null。

25 注意两个方法的返回值：



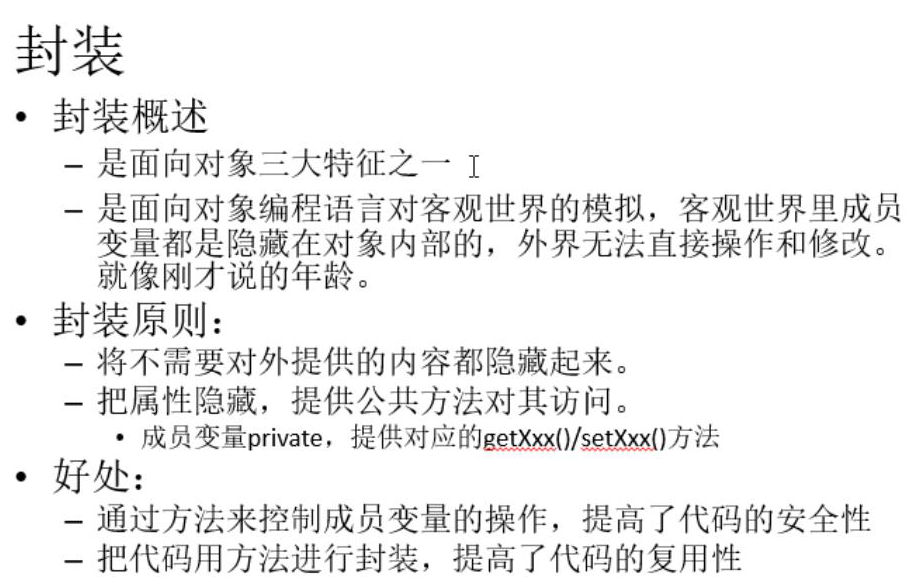
26 exit()方法停止虚拟机。



27 set和get方法不一定要成对出现。他们还有快捷键。

28 封装，继承，多态是面向对象的三大特征。

29 封装的特点：（复用性和安全性）

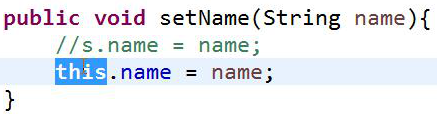


30 修改包名和类名：选中包，按f2

31 不仅成员变量可以被私有，成员函数也可以被私有。

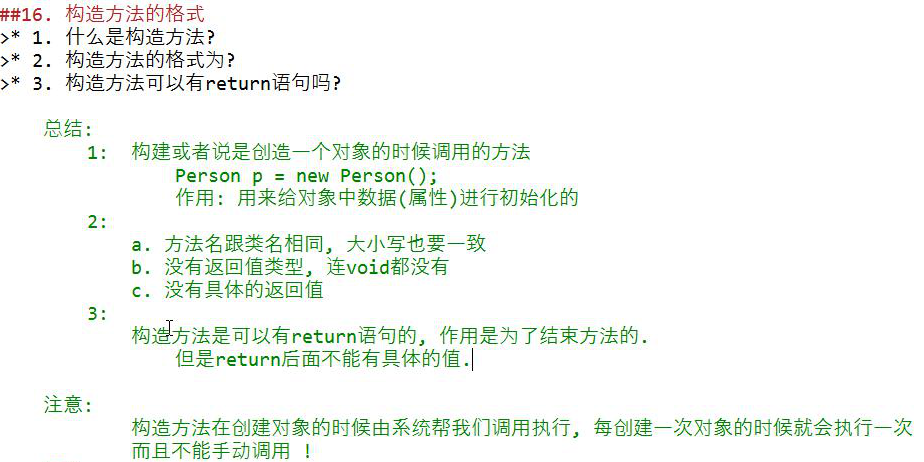
32 this：代表所在类的对象。用来区分局部变量和成员变量的重名问题。

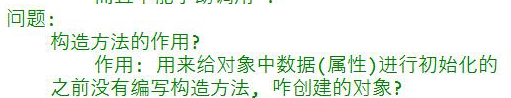
比如this相当于s：



33 所以局部变量就是给自己赋值，而不是给成员变量赋值。

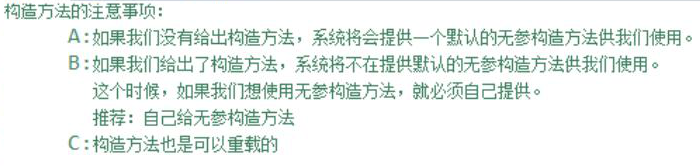
34 构造方法：



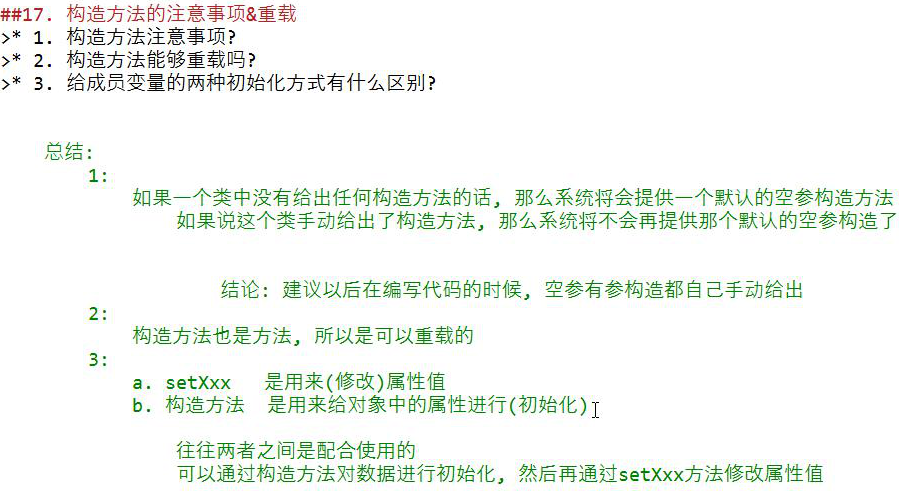


构造函数就是构造方法。

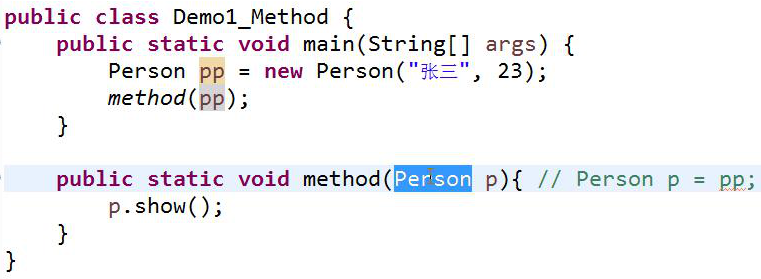
35 构造方法的重载：



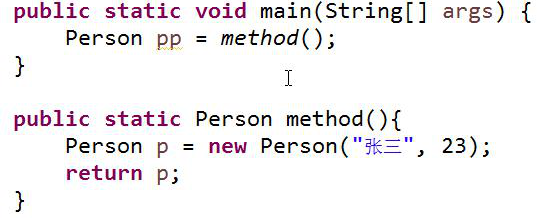
36



37 形参还可以是类名，应该用对象做参数，传递的是具体对象的地址值：



38 返回的类型还可以是类：



39 构造方法的return问题：



40 方法名大写也不会报错，但是不符合规范（业内习惯）。

41 **构造方法用于初始化，get和set用于初始化之后对同一对象进行修改，而不是创建一个新对象。**构造方法只在进入堆中时（也就是初始化时）执行一次。