

一、初识面向对象

1. 面向对象并不是一个技术，而是一种编程指导思想
2. 面向对象是指把现实世界的具体事物全部看成一个一个的对象来解决实际问题
3. 生活中我们是按照对象化的方式解决问题的。如果程序也能够按照生活的中的方式来解决，那么程序就更符合人类的思维习惯，代码看起来会更易理解、更简单、更易维护。

二、类、对象

对象：是对现实世界具体事物的抽象，是类的具体实例。
类(可以看成是设计图)：同类事物的共性，是对对象的抽象。
在Java中，必须先设计类，才能获得对象

1.0 类的定义

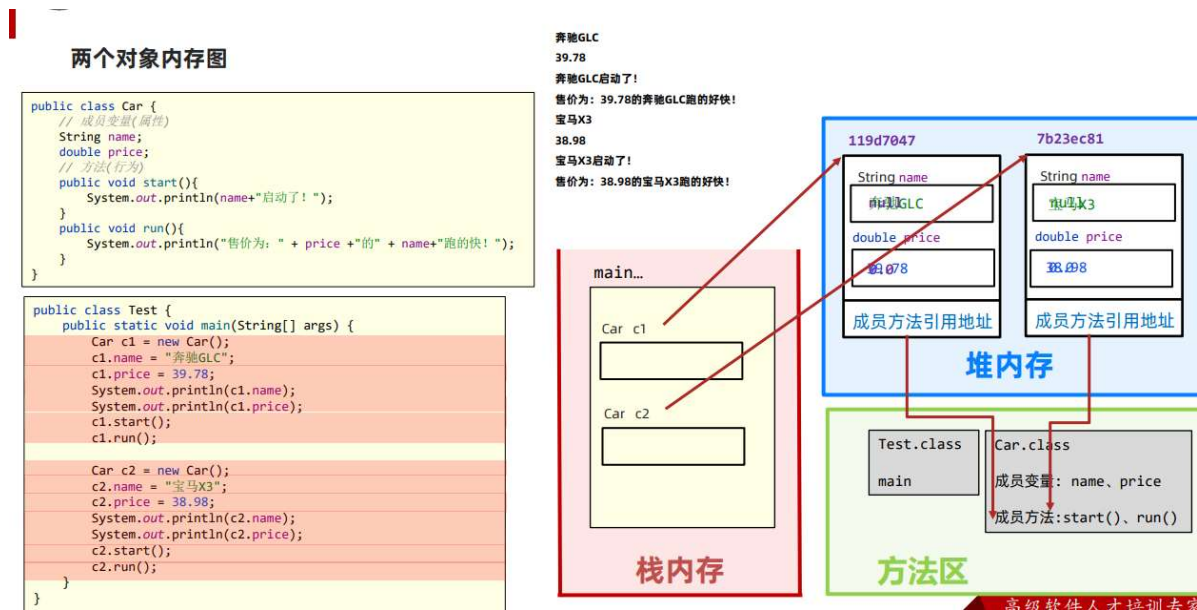
```
public class 类名{  
    //类中四大金刚  
    //1. 属性  
        成员变量:实例成员变量、静态成员变量  
        局部变量:方法中的变量、代码块中的变量、入参  
    //2. 构造器  
        有参构造器  
        无参构造器  
        私有构造器  
        公有构造器  
    //3. 方法  
        静态方法  
        实例方法  
    //4. 代码块  
        静态代码块  
        实例代码块  
    //5. 内部类  
        成员内部类  
        静态内部类  
        局部内部类  
        匿名内部类  
}
```

2.0 注意事项

1. 成员变量的完整定义格式是：修饰符 数据类型 变量名称 = 初始化值； 一般无需指定初始化值，存在默认值。
2. 类名大驼峰模式,方法名小驼峰
3. 一个Java文件中可以定义多个class类，且只能一个类是public修饰，而且public修饰的类名必须成为代码文件名。该java文件编译后会生成多个字节码文件
4. 实际开发中建议还是一个文件定义一个class类。
5. 成员变量 不同数据类型的默认初始值

byte	short	int	char	=	'\u0000'	0
float	double					0.0
boolean						false
类	接口	数组	String			null

三、Java对象内存图



• 内存图说明

1. 加载Test的字节码文件进入方法区，其中字节码文件包括成员方法和成员变量(这里没有)
2. 将Test字节码中的main方法提取到栈中，开始执行main方法里面的内容
3. 将Car的字节码文件加载进方法区，其中包括成员变量name、price和成员方法start()、run()，在堆内存中开辟一块内存空间代表汽车对象，其中会开辟一些区域用来存放属性信息（有默认值），还会汽车对象区域中开辟一些空间用于存储属于该对象的存放在方法区中的成员方法引用地址
4. 在（main方法中）栈内存中开辟一块空间作为main方法的局部变量，用于存放汽车对象的地址
5. 通过局部变量中存放的汽车对象地址找到汽车对象，然后对其name属性进行赋值操作
.....
6. 通过局部变量c1存储的对象地址找到汽车对象，再通过汽车对象中的成员方法引用地址找到方法区中的方法提取到栈中运行，因为该方法知道是哪个对象调用自己的，所以可以获取该对象的成员属性进行操作
.....

• 垃圾回收

1. 注意：当堆内存中的类对象或数组对象，没有被任何变量引用（指向）时，就会被判定为内存中的“垃圾”。
2. Java存在自动垃圾回收器，会定期进行清理。

四、构造器

• 构造器定义

构造器的作用用于初始化一个类的对象，并返回对象的地址

• 构造器定义格式：

修饰符 类名(形参列表){

}

不同修饰符的权限

private 本类有效

缺省 包权限

protected 包权限+子类有效

public 所有类皆有效

- 构造器分类

1. 无参数构造器（默认存在的）：初始化的对象时，成员变量的数据均采用默认值。
2. 有参数构造器：在初始化对象的时候，同时可以为对象进行赋值


- 注意

1. 任何类定义出来，默认就自带了无参数构造器，写不写都有。
2. 一旦定义了有参数构造器，默认无参数构造器就没有了，此时就需要自己写一个无参数构造器了

五、this关键字

1. **this**关键字可以出现在成员方法、构造器中，代表当前对象的地址。
2. 作用：访问当前对象的成员变量、成员方法。


```
public class Car {  
    String name;  
    double price;  
  
    public Car(String name, double price){  
        name = name;  
        price = price;  
    }  
}
```



完美

这里操作的是局部变量（就近原则）所以对象的成员变量是默认值

```
public class Car {  
    String name;  
    double price;  
  
    public void goWith(String name){  
        System.out.println(this.name + "正在和" + name + "一起比赛!!");  
    }  
}
```



完美

六、封装

- 概念

1. 将同一类事物共性（属性和方法）归到一个类中
2. 对象代表什么，就封装对应的数据，并提供相应的方法
3. 使用一些修饰符尽可能的隐藏内部细节，只暴露一些接口与外界发生联系

- 好处

1. 实现专业化分工（有事，找对象，调方法）
2. 隐藏信息，实现细节

- JavaBean

也可以称为实体类，其对象用于在程序中封装数据

- 格式

1. 成员变量使用 `private` 修饰。
2. 提供每一个成员变量对应的 `setXxx()` / `getXxx()`。
3. 必须提供一个无参构造器。

1.0 成员变量与局部变量

区别	成员变量	局部变量
类中位置不同	类中，方法外	常见于方法中
初始化值不同	有默认初始化值	没有，使用之前需要完成赋值
内存位置不同	堆内存	栈内存
生命周期不同	随着对象的创建而存在，随着对象的消失而消失	随着方法的调用而存在，随着方法的运行结束而消失
作用域		在所归属的大括号中