1. **面向对象**

面向对象：万事万物皆对象，将具体的属性和方法封装为一个对象，程序员关注设计对象以及处理对象关系，特点是封装、继承和多态。

类：是一个模板，是对象的类型，确定对象拥有的属性和方法，以Class为单元。

属性：对象具有的各种特征

方法：对象执行的操作

对象：类的具体实例即为对象

**成员变量和局部变量**

成员变量 也就是属性，不同方法内相互独立。

局部变量 仅出现和作用于方法内部，外部无法调用

初始值不同：java会给成员变量一个初始值，不会给局部变量赋初始值

变量同名时，局部变量具有更高的优先级

局部变量前不能放置访问修饰符 (private，public，和protected)

**构造方法**

构造方法用来创建对象，并且可以初始化对象的值

构造方法与类同名且没有返回值

构造方法的重载：方法名相同，参数不同，创建对象时需要传参来调用不同的构造方法

构造方法无法被继承

1.一个类没有添加构造方法时，系统会自动添加无参的构造方法

若写了有参构造，系统就不会自动添加无参构造

2.子类实例化会自动调用父类的构造方法

**This**

this.属性：调用当前对象的属性，用来区分成员变量和局部变量

this.方法：调用当前对象的方法

this（）：构造方法内调用其他构造方法，只能在第一行，只能调用一次；

**static静态变量/方法**

类名直接调用

静态变量用一次变一次，仅有一个固定值；

静态方法中不能直接调用非静态成员和方法，需要创建对象来调用；

static初始化块只在类加载时执行一次，只能给静态变量赋值

程序初始化执行顺序：static 静态初始化块——普通初始化块——构造方法

static常驻内存，程序加载时就会被分配空间，执行效率高，但释放内存时间长

**static final必须声明的同时初始化，通过类名访问**

1. **封装**

概念：将类的信息隐藏在类的内部，不允许外部程序直接访问，而是通过该类提供的特定的方法（setter.getter）来实现对隐藏信息的操作和访问

类：封装的是对象属性和行为

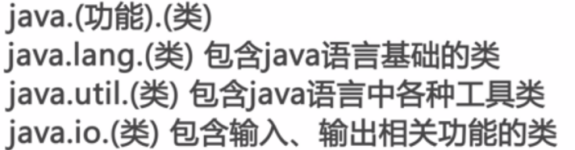
方法：封装的是功能具体实现

访问修饰符：封装的访问的权限

**使用包管理java中的类**

全小写，作用：管理java文件；解决同名文件冲突

必须放在java源程序第一行



包的使用：如何在其他包中使用我们自定义的类

Import com.imooc.\*; 导入包下所有文件

Import com.imooc.hello; 导入hello

**访问修饰符—修饰属性和方法的访问范围**

