|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 对象 | 类 | 抽象类 | 接口 |
| 组成 | 简单的说就是类的实例 | 宏观上：  成员变量：  成员方法  微观上：  类变量、方法实例变量、方法  静态代码块+构造函数 | 1.用abstract修饰的类  2.可以有抽象方法也可以没有  3. 属性和行为【变量和方法】 | 1. 全局常量【public static final】  2. 公共的抽象方法【public abstract】  3. 内部类【内部类，内部接口，内部枚举类】 |
| 特点 | 1. 对象是Java中的核心   2.对象具有唯一性  3. 不允许直接访问对象，而是通过引用访问  4. 坐等GC回收内存 | 可以直接实例化对象 | 1.抽象类和抽象方法必须用abstract关键字来修饰  2.abstract和static不能同时使用  3.抽象类可以没有抽象方法  4.含有抽象方法的类必须声明为抽象类  5.抽象方法只有方法声明，没有方法体，定义在抽象类中  格式：修饰符 abstract 返回值类型 函数名(参数列表);  6.抽象类不可以被实例化，也不可以用new创建对象  7.抽象类通过其子类实例化，且子类需要覆盖掉抽象类中所有的抽象方法后才可以创建对象，否则该子类也是抽象类 | 1.没有构造方法，不能实例化  2.接口只能继承接口，不能继承类  3.接口里没有普通方法，方法全是抽象的 |
| 定义格式 | 1.类名 对象名 = new 类名();  2.类名 对象名 = new 类名(参数); | class 类名 {  属性；  方法；  } | [修饰符] abstract 类名{} | 1.[修饰符] interface 接口{}  2.[修饰符] interface 接口 extends 接口1{} |
| 使用 | 1.对象引用.变量 【new 类名().变量】  2.对象引用.函数()【new 类名().函数】 | 1.【类】.【类变量、类方法】  2.实例化对象.【实例变量、实例方法】 | 只有被继承才能使用。  除了不能被实例化，类的其他功能依然存在  父类本身是抽象类，所以其方法不能使用 | [修饰符] class 类名 implements 接口1，接口2...  1.接口的实现必须在extends之后  2.实现接口的方法必须是public类型 |