|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 接口 | 抽象类 | 类 | 对象 |
| 组成 | <http://hao.360.cn/>全局常量  公共的抽象方法  内部类（内部类、接口、内部枚举类） | 1成员变量 2成员方法【普通、抽象方法】 3构造方法  4初始化块 5内部类  6内部枚举类 | 1【宏观】成员变量 成员方法  2【微观上】类变量、方法  实例变量、方法  静态代码块+构造函数 | 成员变量 成员方法 |
| 特点 | 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，全是抽象方法  4类与接口之间是实现关系而且类可以继承一个类的同时实现多个接口  5接口之间可以继承 | 1抽象类和抽象方法必须用abstract修饰  2 含有抽象方法的类必须被声明为抽象类  3、抽象方法只有方法声明，没有方法体定义在抽象类中  4抽象类不可以被实例化，即不能用new创建对象  5 | 1可以直接实例化对象  2 可以继承其他类，普通方法只能单一继承  3 可以继承接口，可以多元继承  4可以通过类部类实现多继承 | 1是java的核心  2具有唯一性  3不予许直接访问对象，而是通过引用访问  4 GC自动回收内存，无需手动回收 |
| 定义格式 | **[Public] interface 接口{}；**  **[public]interface 接口extends**  **接口1，接口2{}** | **public abstract class name（）**  **{**  **}** | **（public）Class 类名{**  属性； //状态：成员变量==静态变量  方法 ： //行为：成员方法==动态变量  **}** | **类名 对象名 = new类名（）；**  **Dog dog = new Dog（）** |
| 使用 | 1接口继承其它接口，添加新的属性和抽象方法  2接口没有构造方法，不能实例化  3子类实现某个接口时，必须实现接口中所有的抽象方法否则这个类被定义为抽象类  4接口不能相互实现，但可以继承  5多个无关类可以实现同一个接口  6一个类可以实现多个无关接口 | 1成员方法可以只声明不定义  2抽象类中有构造函数且可被子类继承，由子类调用  3抽象类通过子类实例化，且子类必须覆盖父类所有抽象方法  4 abstract不能不static、final、private共存【1】 | 1【类】【类变量、类方法】  2实例化：实例化对象，对象【实例变量、实例方法】  3被单继承  4可以定义多个构造方法：使用new+构造方法创造新的对象 | 通过引用来访问对象  **对象引用。对象**  **对象引用。函数（）** |
| 拓展性 | 接口与实现类之间存在多态性 |  |  | 封装：继承：多态  **封装**：隐藏对象信息状态、外部不能直接访问，只能通过给定的方法访问  **继承：**软件复用的手段**关键词**extends  **多态：**相同类型的变量、调用同一个方法时呈现出洞中不同的行为特性 |
|  |  |  |  |  |

【1】static不依赖于对象可直接通过类名调用，而abstract没有构造方法，没有实例化的对象