==========================三大修饰符===================================

#### **abstract(抽象的)**

1) 修饰方法 ----->抽象方法

特点：只有声明，没有实现。连{}都没有。

语法：访问修饰符 abstract 返回值类型 方法名(形参列表)；

注意：访问修饰符 abstract 没有先后顺序

注意：抽象方法必须定义在抽象类中

2) 修饰类 ------>抽象类

语法：abstract class 类名{ }

注意：a. 抽象类不能通过new关键字创建对象

b. 抽象类中有构造方法。在构建子类对象 先去构建父类对象时使 用。

c. 抽象类可以有普通的成员（方法和属性）

注意：若子类继承了抽象类，则必须实现抽象类中的所有抽象方法，要么子类也 必须为抽象类

作用：强制使用多态

**static(静态的)**

1. 修饰属性 -------->静态属性、类变量【重点】

语法：static 数据类 型 变量名；

特点：与创建多少对象无关，是全类共有的。

使用：对象名.静态属性名

类名.静态属性名 --------->建议使用 IMG_256

2) 修饰方法 --------->静态方法【重点】

语法：访问修饰符 static 返回值类型 方法名(形参列表){

//方法的实现

}

注意：访问修饰符 和 static没有先后顺序

使用：对象名.静态方法名(实参);

类名.静态方法名(实参); -------> 建议使用

IMG_257

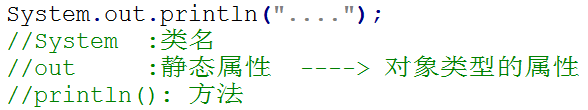
注意： a. 静态方法中只能访问静态成员（静态属性和静态方法）。

b. 静态方法中不能使用this/super关键字

c. 静态方法可以被继承，

静态可以被覆盖（只能被静态方法覆盖），但是没有多态。

**面试题：System.out.println(); ？？？？？？**



1. static 修饰的代码块----------->静态代码块。

a. 动态代码块 {} [了解]

定义位置：类以内，方法以外

执行时机：创建对象的第二步，初始化属性的时执行。跟声明属性 并且赋值，按照先后顺序，对属性进 行初始化

b. 静态代码块【开发重点】

i.概念：被static修饰的动态代码块

ii.定义位置：类以内，方法以外

语法：static {}

执行时机：在类加载的时候执行。跟声明静态属性并且赋值这句 代码，按照先后顺序，对静态属性进行 初始化。

**类加载：**

当JVM第一次使用一个类的时候，会根据 ClassPath（类路径）找到这个类的.class文件， 获取这个类的所有信息（包名、类名、属性、 父类、方法、 构造方法...）,并且进行保存。

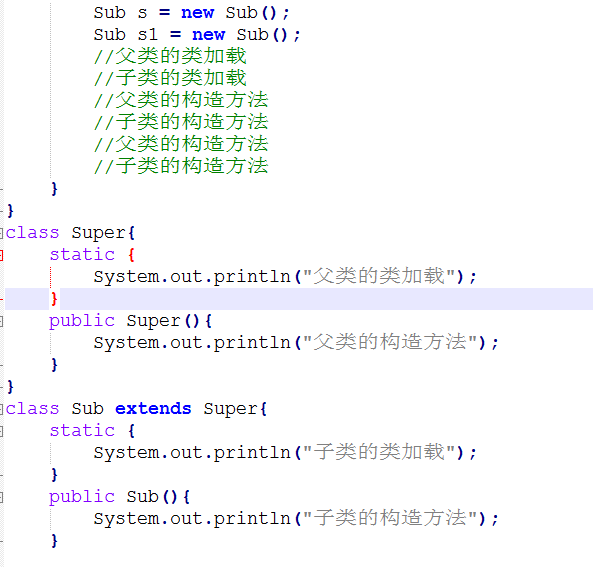
这个过程称之为类加载。并且只执行一次。

**类加载的执行时机**：

i. 第一次通过类名访问静态属性或静态方法时。

ii.第一次创建对象时，会进行类加载。

iii.当第一次创建子类对象时，会先去进行父类的 类加载，再进行子类的类加载。



**final(最终的)**

1.修饰变量

a.被final修饰，一旦被赋值不允许改变。只能赋值一次。

b.被final修饰的变量，称为作用范围内的常量。

c. i.修饰成员变量、属性

特点:被final修饰的属性，没有默认值

赋值时机：声明的同时并且赋值、构造方法、动态代码块[了解]

ii.修饰类变量、静态属性 ----》静态常量

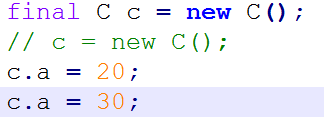
赋值时机：声明的同时并且赋值、静态代码块

iii.注意：

被final修饰的基本数据类型，数值不能改变

被final修饰的引用数据类型，指向的对象（地址）不能改变

但是 对象的属性的值可以改变



2. 修饰方法

被final修饰的方法 不能被覆盖

3. 修饰类

被final修饰的类 不能被继承（断子绝孙类）

**哪些修饰符不能连用？？？**

private 和 abstract 不能连用

final 和 abstract 不能连用

static 和 abstract 不能连用