final——

final用于类表示该类不能派生子类。

final用于方法表示该方法不能被子类重写。

final用于变量时表示常量，类似于C/C++的const关键字。

final用于成员变量表示该成员变量为常量，不能被修改，必须在变量定义时赋值。

final用于局部变量表示该局部变量为常量，不能被修改，可以在变量定义时赋值，也可以先定义变量后赋值。

static——

static用于成员变量表示该变量只有一份，也就是说静态成员变量属于类而不属于某个具体的类实例对象，所有的类实例对象共享这个静态成员变量，访问静态成员变量时通过对应的类名完成。静态成员变量在定义时可初始化赋值，也可不赋值，不赋值时会被自动初始化，切记局部变量不可以是静态的。

static用于方法，使得不需要实例化类就可以通过类名来调用静态方法，需要注意的是在静态方法内不可以使用this关键字，不可以调用非静态方法，也不可以引用非静态成员变量。

static用于类，这里指的是内部类，那么在别的地方就可以通过外部类名来引用这个静态的内部类。

static还可以用于类的代码块，叫做静态代码块，是在类中独立于类成员的static语句块，可以有多个，位置可以随便放，它不在任何的方法体内，JVM加载类时会执行这些静态的代码块，如果static代码块有多个，JVM将按照它们在类中出现的先后顺序依次执行它们，每个代码块只会被执行一次。

类的访问权限（Y/N）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 关键字 | 类 | 包 | 子类 | 其它包 |
| public | Y | Y | Y | Y |
| protected | Y | Y | Y | N |
| default | Y | Y | N | N |
| private | Y | N | N | N |

public、protected和private三个关键字，可以用于类（内部类）、成员变量和成员函数，默认的访问权限是包内有效，当类访问权限与成员变量或成员函数的访问权限不同时，选择最低访问权限。

interface/implements/extends/class——

interface用于声明接口，接口中的方法只有声明，没有实现，访问权限可以是public或者默认权限，还可以指定为abstract。

implements用于实现接口，需要实现接口中的所有方法，可以同时实现多个接口。

extends用于继承父类或父接口，继承父类时只能是单继承，与C++的多继承不同，接口继承支持多继承。

class用于声明类，访问权限可以是public或者默认权限，还可以指定为abstract、final，访问权限对于顶层类和内部类的限制是不同的。

super——

super关键字可以用来调用父类的构造函数、成员变量和成员方法。

abstract——

abstract用于类表示这个类为抽象类，不能实例化。

abstract用于方法表示这个方法为抽象方法，只需声明，不用实现，由子类实现，抽象方法不能用private和static关键字。