Static

关键字，代表静态

可以修饰变量，方法，代码块，内部类

如果四个同时出现，按代码从上往下的顺序执行，存在方法区内，可以修改，有共享性。

static修饰变量

静态变量 (存在于静态区)

Static 变量的定义；

静态变量属于类也称为类变量，

生命周期：类加载之后不在移除，直到整个程序运行结束；静态变量随着类加载被加载到方法区的静态区当中，直到类被移除才会释放。

调用格式：类名.静态变量名/对象.静态变量名；以上两种方式java都支持，但是为了提高程序的可读性，建议使用通过类名来调用静态变量。

应用场景：

1. 静态变量：如果每个对象有各自的特征点，不适合用static修饰，如果每个对象的特征值都一样则可以使用static修饰；
2. System.in System.out in out就是静态变量

注意：

1. 在构造代码块当中可以定义静态代码块吗？

不可以，类加载之后才能创建对象，构造代码块在创建对象时执行，静态代码块是在类加载时就要初始化。

1. 在构造方法中可以定义静态变量吗？

不可以，类加载之后才能创建对象，构造方法在创建对象时调用，静态代码块是在类加载时就要初始化

1. 能否在构造代码块或者构造方法中给静态变量赋值？

可以

静态方法：

Static修饰方法即可

生命周期：随着类加载而加载到方法区的静态区中，在被调用时才开始运行。

调用的格式： 类名.静态方法名() /对象名.静态方法名()

例如：Arrays.toString() Arrays.copyOf()

注意：

1. 静态方法中能否定义静态变量？

不可以，随着类加载而加载到方法区的静态区中，只是存储方法并未执行该方法，在被调用时才开始运行。静态变量是在类加载时加载到静态区并且初始化

1. 静态方法中能否直接调用非静态方法。

不可以，非静态方法和非静态变量必须得通过对象来调用，而当我们使用类名来调用静态方法时，可能根本没有创建对象。如果想调用可以在调用非静态方法必须创建对象调用，通过对象调用非静态方法。

1. 静态方法中是否可以用this、super这俩关键字

不可以，this指代当前对象super指代当前父类的对象，没有对象，他俩都不能使用

1. main方法中是否能用this super?

不能 main方法是静态方法

1. 静态方法能否重载？

可以

1. 子类能否继承父类的静态方法？

可以继承父类的静态方法

1. 子类能否重写父类的静态方法？

不能重写，静态方法和类绑定

1. 子类中能不能存在和父类方法签名等都一致的静态方法？

可以

静态代码块：

格式：

static {

}

执行：

在类真正被使用时会执行静态代码块；仅执行一次；

创建类的对象

调用类的静态方法。

调用类的静态属性

在类加载时被调用(辩证理解 Person.class时不会输出静态代码块)

注意：

1. 静态代码块中能否定义静态变量？

不可以，静态变量在类加载时初始化。代码块在类真正被使用时才执行静态代码块

1. 静态代码块中能不能给静态变量赋值？可以

3.静态代码块 构造代码块 构造方法的执行顺序？

静态代码块-->构造代码块 -->构造方法

4.基于父子类时静态代码块 构造代码块 构造方法的执行顺序？

父类的静态代码块--》子类的静态代码块--》父类构造代码块--》父类构造方法--》子类构造代码块--》子类构造方法

先父后子，先静态再动态，先构造代码块再构造方法

基于父子类并且子类中定义了（引用类型的属性）属性，创建类对象时，属性初始化和构造代码块执行的顺序取决于程序的顺序。

基于父子类然后父类中定义了属性，也按先后顺序执行

有静态变量和静态代码块时要取决于代码的顺序。

class A{

{

i = 5;

}

int i;

class B{

static{

i=7;//编译正确，但是不能使用，使用的话会报错

}

static int i;

}

}