static无需创建对象，通过类本身去调用方法或变量（只要类被加载了，就可以通过类名去访问）

一．静态的数据的生命周期：

1. 类文件先存在于硬盘
2. 加载主类文件——>在方法区中解剖类中的方法（静态方法和非静态方法同时存在方法区）（若存在静态变量，则在方法区中的静态数据共享区内初始化该静态变量）（静态代码块是在加载类文件时执行的）
3. 调用主类中的main方法——>在栈内存为main开辟一块内存空间——>执行main方法中的语句
4. 当main方法中第一次调用某个类时——>立即在方法区中加载该类文件——>解剖类中的方法和静态变量和执行静态代码块
5. 当在main方法中创建某个类的对象时——>在main方法的内存中开辟一个引用变量指向堆内存——>在堆内存中开辟一个内存——>初始化非静态的成员变量——>调用构造方法为成员变量赋值
6. 当在main方法中通过对象调用某个类的方法时——>立即开辟一个新的方法栈内存——>执行方法内的语句——>方法中语句结束，方法的栈内存消失
7. 当这个对象不再被引用变量引用，该对象等待垃圾回收器回收
8. 类文件属于懒惰型，加载后直到该程序执行结束才卸载
9. 静态的成员数据优先于对象的数据而存在的

因为：静态的成员数据是在加载类时初始化的，对象的数据是在创建对象时才存在于堆内存中 Person p = new Person();

二.static的三种修饰

1.static变量

注意：

静态变量被所有对象共享，在方法区内存中只有一个副本，仅在类初次加载时会被初始化，随类文件的消失而消失

非静态变量被对象所拥有，在堆内存内存中有多个副本，在创建对象时被初始化，随对象被垃圾回收器回收而消失

应用场景：

在类中，当有一个数据需要被对象共享，则该成员变量使用static修饰

2.static方法

static方法就是没有this的方法（因为没有对象）（不能出现super）

注意：

在静态方法中不能直接访问类的非静态成员变量和非静态成员方法（可以在静态方法中创建一个对象，间接访问非静态的成员变量）

在非静态成员方法中可以访问静态成员方法或变量

应用场景：

当不想通过创建对象去访问某个成员方法，并且该成员方法不会访问非静态成员变量。

即：创建工具类时，使用静态方法

3.静态代码块

一次类定义之后，它的一些信息已经写进代码区了，静态区域块就会自动在加载类时执行执行