http://www.cnblogs.com/study-everyday/p/6752715.html

1.首先，需要明白类的加载顺序。

(1) 父类静态对象和静态代码块

(2) 子类静态对象和静态代码块

(3) 父类非静态对象和非静态代码块

(4) 父类构造函数

(5) 子类 非静态对象和非静态代码块

(6) 子类构造函数

**加载过程：**

**1、**JVM会先去方法区中找有没有相应类的.class存在。如果有，就直接使用；如果没有，则把相关类的.clss加载到方法区。

**2、**在.class加载到方法区时，先加载父类再加载子类；先加载静态内容，再加载非静态内容

**3、**加载静态内容：

* 把.class中的所有静态内容加载到方法区下的静态区域内
* 静态内容加载完成之后，对所有的静态变量进行默认初始化
* 所有的静态变量默认初始化完成之后，再进行显式初始化
* 当静态区域下的所有静态变量显式初始化完后，执行静态代码块

**4、**加载非静态内容：把.class中的所有非静态变量及非静态代码块加载到方法区下的非静态区域内。

**5、**执行完之后，整个类的加载就完成了。

对于静态方法和非静态方法都是被动调用，即系统不会自动调用执行，所以用户没有调用时都不执行，主要区别在于静态方法可以直接用类名直接调用（实例化对象也可以），而非静态方法只能先实例化对象后才能调用。

**二、对象的创建过程**

**1、**new一个对象时，在堆内存中开辟一块空间。

**2、**给开辟的空间分配一个地址。

**3、**把对象的所有非静态成员加载到所开辟的空间下。

**4、**所有的非静态成员加载完成之后，对所有非静态成员变量进行默认初始化。

**5、**所有非静态成员变量默认初始化完成之后，调用构造函数。

**6、**在构造函数入栈执行时，分为两部分：先执行构造函数中的隐式三步，再执行构造函数中书写的代码。

**隐式三步：**

①执行super()语句

②显示初始化(对开辟空间下的所有非静态成员变量进行)   
  
 ③执行构造代码块

**7、**在整个构造函数执行完并弹栈后，把空间分配的地址赋给引用对象。

**三、其他**

**super语句，可能出现以下三种情况:**

**1）**构造方法体的第一行是this()语句，则不会执行隐式三步，而是调用this()语句所对应的的构造方法，最终肯定会有第一行不是this语句的构造方法。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | package mytest.javaBase;    public class Student {        private String name;        private String age;        Student() {      };        Student(String name) {          this.name = name;        };        Student(String name, String age) {          // 不会执行隐式三步          this(name);          this.age = age;      };    } |

**2）**构造方法体的第一行是super()语句，则调用相应的父类的构造方法，   
**3）**构造方法体的第一行既不是this()语句也不是super()语句，则隐式调用super()，即其父类的默认构造方法，这也是为什么一个父类通常要提供默认构造方法的原因；