Java的类装载步骤  
在Java中，类装载器把一个类装入Java虚拟机中，要经过三个步骤来完成：装载、链接和初始化，其中链接又可以分成校验、准备和解析三步，除了解析外，其它步骤是严格按照顺序完成的，各个步骤的主要工作如下：

装载：查找和导入类或接口的二进制数据；  
链接：执行下面的校验、准备和解析步骤，其中解析步骤是可以选择的；  
校验：检查导入类或接口的二进制数据的正确性；  
准备：给类的静态变量分配并初始化存储空间；  
解析：将符号引用转成直接引用；  
初始化：激活类的静态变量的初始化Java代码和静态Java代码块。  
初始化类中属性是静态代码块的常用用途，但只能使用一次。

## 静态代码块的初始化顺序

|  |
| --- |
| class Parent{  static String name = "hello";  {  System.out.println("parent block");  }  static {  System.out.println("parent static block");  }  public Parent(){  System.out.println("parent constructor");  }  }   class Child extends Parent{  static String childName = "hello";  {  System.out.println("child block");  }  static {  System.out.println("child static block");  }  public Child(){  System.out.println("child constructor");  }  }   public class StaticIniBlockOrderTest {   public static void main(String[] args) {  new Child();//语句(\*)  }  } |

问题：当执行完语句(\*)时，打印结果是什么顺序？为什么？  
解答：当执行完语句(\*)时，打印结果是这样一个顺序 :

|  |
| --- |
| parent static block child static block parent block parent constructor child block child constructor |

分析:当执行new Child()时，它首先去看父类里面有没有静态代码块，如果有，它先去执行父类里面静态代码块里面的内容，当父类的静态代码块里面的内容执行完毕之后，接着去执行子类(自己这个类)里面的静态代码块，当子类的静态代码块执行完毕之后，它接着又去看父类有没有非静态代码块，如果有就执行父类的非静态代码块，父类的非静态代码块执行完毕，接着执行父类的构造方法；父类的构造方法执行完毕之后，它接着去看子类有没有非静态代码块，如果有就执行子类的非静态代码块。子类的非静态代码块执行完毕再去执行子类的构造方法，这个就是一个对象的初始化顺序。

总结:  
对象的初始化顺序:首先执行父类静态的内容，父类静态的内容执行完毕后，接着去执行子类的静态的内容，当子类的静态内容执行完毕之后，再去看父类有没有非静态代码块，如果有就执行父类的非静态代码块，父类的非静态代码块执行完毕，接着执行父类的构造方法；父类的构造方法执行完毕之后，它接着去看子类有没有非静态代码块，如果有就执行子类的非静态代码块。子类的非静态代码块执行完毕再去执行子类的构造方法。总之一句话，静态代码块内容先执行，接着执行父类非静态代码块和构造方法，然后执行子类非静态代码块和构造方法。

注意:子类的构造方法，不管这个构造方法带不带参数，默认的它都会先去寻找父类的不带参数的构造方法。如果父类没有不带参数的构造方法，那么子类必须用supper关键子来调用父类带参数的构造方法，否则编译不能通过。