**Java初始化代码**

Java初始化代码有三个区域，静态代码块，非静态代码块，构造函数。

* **静态代码块**

public class MyClass{

public static int a=0;

static{

a=1;

}

}

静态代码块只有在类第一次被加载的时候执行，而且在整个应用程序生命期间只会被执行一次。静态代码块是给类初始化的，而且只能初始化静态变量（不包括final修饰的常量）和调用静态方法。以下几种情况都会使类被加载并且执行静态代码块：

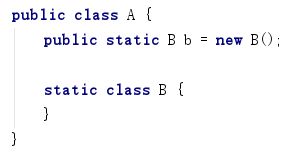
1、MyClass.a;

2、MyClass myClass;

3、MyClass 的子类被加载（子类被加载包括上面1、2两种形式）；

注意：

1、另外当类被加载的时候（包括上面的几种情况），类里面的静态变量如果有默认初始值，类里面的静态变量也会被初始化，而且有且只有一次初始化，如下当A类被加载的时候，A类里面的静态变量b也会被初始化，而且在应用程序里只会被初始化一次：



2、内部类中不允许存在静态代码块，也不能存在静态变量。

* **构造代码块**

public class MyClass{

public static int a=0;

public int b=0;

{

a=1;

b=1;

}

}

构造代码块只有创建对象的时候才会执行，创建对象时它会在调用构造方法之前被执行初始化变量，包括静态变量和非静态变量，也能调用静态方法和非静态方法。它与构造方法的区别是，一个类里面有多个构造代码块和多个构造方法，创建对象的时候只能执行其中的一个构造方法，但是所有的代码块都会先于构造方法执行。不同的构造方法构造出不同的对象，但是这些对象在创造之前都会执行相同的代码块来初始化自己的变量。

* **初始化顺序**
* **只有一个类**

静态变量**>**静态初始化块**>**变量**>**初始化块**>**构造器。

* **继承情况**

父类-静态变量**>**父类-静态初始化块**>**子类-静态变量**>**子类-静态初始化块**>**

父类-变量**>**父类-初始化块**>**父类-构造器**>**子类-变量**>**子类-初始化块**>**子类-构造器

不管是只有一个类还是继承情况，假如类里面有main()方法，即程序的入口，要想启动程序执行这个main()方法，那么一定是执行完了类里面的静态块或者父类和子类的静态块之后才会执行main()方法进入主线程。