JAVA 类 初始化过程

作者：WangRun
原文链接：https://ld246.com/article/1508416752371
来源网站：链滴

在Java的对象产生的时候虚拟机jvm会做一系列的行为，而理解这些行为将有助于我们更深入的理解Ja
a

当我们第一次调用一个类的某个静态方法或访问某个静态变量时将首先发生类加载，其过程如下

1. Java虚拟机JVM会先去方法区中查找是否已经加载java类名.class如果已经加载将执行下一步，如没有加载则将通过类的完全限定名找到相应的.class文件加载到方法区并创建一个Class对象，静态变只在Class对象首次加载时初始化。
2. 加载完成后，将首先将Class对象中的非静态内容加载到方法区的非静态区中
3. 然后再加载静态内容：

- 首先将静态内容加载到静态区中
- 静态内容加载完成之后，首先对静态变量进行默认初始化（基本数据类型赋值为默认值，引用类型null）
- 然后对静态变量进行显式初始化（即初始化为我们程序所给的值）
- 接着执行静态代码块中的代码（形如static{.....}）
- 类加载过程结束

当我们调用Object obj = new Object();创建对象时，首先需要明白即使没有显示的使用static关键，但是构造器实际也是静态方法，因此在创建对象时也将先执行类的加载，然后执行下列过程：

1. 首先new Object()在堆中为Object对象分配足够的存储空间
2. 将这段存储空间清零，然后将所有类的非静态内容加载到这段空间中
3. 对非静态成员变量进行默认初始化（基本数据类型赋值为默认值，引用类型为null）
4. 然后调用构造函数
5. 在构造函数函数体执行之前先隐式的执行如下几个步骤

1.如果构造函数没有显示的调动父类的构造函数时会默认调用super();调用父类的无参构造方法

2.对非静态成员进行显式初始化（即初始化为我们程序所给的值）

3，执行构造代码块（形如{.....}）

6. 执行构造函数中的代码
7. 将对象堆的地址赋值给栈中的obj，由此obj指向堆中的Object对象的实体

值得注意的是：

1. 当使用类字面常量来生成Class对象即Object.class时并不会自动执行Class对象的初始化。
2. 当访问一个类的编译期常量（static final 名字）时并不会触发对类的初始化
3. 当使用Class.forName(类的完全限定名)时一定会立刻触发类的初始化