Java类初始化

## 初始化

ClassLoader加载类到JVM后，并没有完成初始化。

一个类的初始化包括3个步骤：

* 加载（Loading），由类加载器执行，查找字节码，并创建一个Class对象（只是创建）；
* 链接（Linking），验证字节码，为静态域分配存储空间（只是分配，并不初始化该存储空间），解析该类创建所需要的对其它类的应用；
* 初始化（Initialization），首先执行静态初始化块static{}，初始化静态变量，执行静态方法（如构造方法）。

根据java虚拟机规范，所有java虚拟机实现必须在每个类或接口被java程序首次**主动使用**时才初始化。

**主动使用**有以下6种：  
1) 创建类的实例;  
2) 访问某个类或者接口的静态变量，或者对该静态变量赋值（如果访问静态编译时常量(即编译时可以确定值的常量)不会导致类的初始化）;  
3) 调用类的静态方法;  
4) 反射（Class.forName(xxx.xxx.xxx)）;  
5) 初始化一个类的子类（相当于对父类的主动使用），不过直接通过子类引用父类元素，不会引起子类的 初始化;

6) Java虚拟机被标明为启动类的类（包含main方法的）。

## 初始化顺序

1. 编译器初始化所有的已分配的空间为二进制0；
2. 执行父类静态代码，执行子类静态代码；
3. 初始化父类成员变量；
4. 初始化父类构造函数；
5. 初始化子类成员变量；
6. 初始化子类构造函数。

## Some Note

类与接口的初始化不同，如果一个类被初始化，则其父类或父接口也会被初始化，但如果一个接口初始化，则不会引起其父接口的初始化。

当访问一个Java类或接口的静态域时，只有真正声明这个域的类或接口才会被初始化，通过子类引用的父类的静态域，基类会执行初始化而子类不会不初始化。

static{}是在第一次初始化时执行，且只执行一次；

编译时常量必须满足3个条件：（static的，final的，常量），对它们的应用，不会引起类的初始化。