链滴

# java 中类与对象的加载顺序

作者：Pleuvoir

原文链接：https://ld246.com/article/1522467678892

来源网站：链滴

## 1. 何时会初始化类?

---

- 主动new
- 调用类的静态方法
- 操作类或接口的静态属性或者为其赋值
- 反射操作
- 指定一个类作为Java虚拟机启动时的初始化类。
- 初始化一个类的子类 同样会初始化它的父类

以上均为初始化类的手段。

---

## 2. 加载顺序是什么?

创建两个类：父类Parent，子类Child继承父类Parent用于演示初始化子类时对父类的影响。

```
public class Parent {

    static String parenntName = "Parent";

    Integer age = 18;

    static {
        System.out.println("父类静态代码块。。。" + parenntName);
    }
```

原文链接：java 中类与对象的加载顺序

```
    {
        System.out.println("父类代码块。。。");
        System.out.println(age);
    }

    public static void print() {
        System.out.println("父类静态方法。。。" + parenntName);
    }

    public Parent(String content) {
        System.out.println(age);
        System.out.println("父类带参构造方法。。。");
    }

    public Parent() {
        System.out.println("父类构造方法。。。");
    }

}

public class Child extends Parent {

    static String childName = "Child";

    static {
        System.out.println("子类静态代码块。。。" + childName);
    }

    {
        System.out.println("子类代码块。。");
    }

    public Child() {
        System.out.println("子类构造方法。。。");
    }


    public Child(String name) {
        System.out.println("子类带参构造方法。。。");
    }
}
```

2.1 然后写一个main方法new Parent(),执行以后，输出的结果如下：

```
父类静态代码块。。。Parent
父类代码块。。。
18
父类构造方法。。。
```

这里可以看出来代码的执行顺序：先初始化静态变量 -> 静态代码块 -> 成员变量 -> 代码块 -> 构造法

2.2 new Child()试试：

```
父类静态代码块。。。Parent
子类静态代码块。。。Child
父类代码块。。。
18
父类构造方法。。。
子类代码块。。
子类构造方法。。。
```

执行顺序：先初始化父类静态变量 -> 父类静态代码块 -> 子类静态变量 -> 子类静态代码块 -> 父类员变量 -> 父类代码块 -> 父类构造方法 -> 子类成员变量 -> 子类代码块 -> 子类构造方法

2.3 试试调用类的静态方法Parent.print()，看看会怎么样：

```
父类静态代码块。。。Parent
父类静态方法。。。Parent
```

执行顺序：先执行静态变量 -> 静态代码块 -> 静态方法

2.4 操作类或接口的静态属性或者为其赋值Parent.parenntName = "jack"：

```
父类静态代码块。。。Parent
```

执行顺序：先执行静态变量 -> 静态代码块

PS:这里如果用Child去操作父类的静态方法或者静态属性效果是一样的。

2.5 使用有参构造new Child("pleuvoir"),结果如下：

```
父类静态代码块。。。Parent
子类静态代码块。。。Child
父类代码块。。。
18
父类构造方法。。。
子类代码块。。
子类带参构造方法。。。
```

显然这里和直接new Child()的结果类似，唯一不同的是最后执行的是子类带参构造方法，让我惊奇是父类也有类似的有参构造方法，实例化子类时JVM默认调用的是父类的无参构造方法。也就是说**子的构造方法，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。**

# 3. 结论

静态属性和静态代码块只会执行一次

静态属性 > 静态代码块 > 成员变量 > 代码块 > 构造方法

静态内容：静态属性和静态代码块（静态属性先于静态代码块）

普通内容：成员变量和代码块（成员变量先于代码块）

首先执行父类静态内容，然后执行子类静态内容，接着父类普通内容，父类构造函数，子类普通内容子类构造函数

如果调用类的静态属性或者静态方法会执行实际执行类的静态内容