



链滴

# 类的实例化顺序

作者: [figo2young](#)

原文链接: <https://ld246.com/article/1567406399984>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

package core;

/**
 * @author 关注微信公众号：Java学习交流社区
 * (更多技术干货、原创文章、面试资料，群聊吹水一应俱全)
 */
public class InitOrderClass {

    public static void main(String[] args) {
        System.out.println("主函数");
        new Child();
    }
}

class Parent {
    static {
        System.out.println("Parent的static代码块...");
    }

    public String s1 = print("Parent的成员变量...");
    public static String s2 = print("Parent的static变量...");

    protected Parent() {
        System.out.println("Parent的构造函数...");
    }

    {
        System.out.println("Parent的代码块");
    }

    public static String print(String str) {
        System.out.println(str);
        return null;
    }
}

class Child extends Parent {

    public String s1 = print("Child的成员变量...");

    public static String s2 = print("Child的static变量...");

    public Child() {
        System.out.println("Child的构造函数...");
    }

    private static Parent parent = new Parent();

    static {
        System.out.println("Child的static代码块...");
    }
}

```

```
    {
        System.out.println("Child的代码块...");
    }
}
```

返回结果如下：

```
主函数
Parent的static代码块...
Parent的static变量...
Child的static变量...
Parent的成员变量...
Parent的代码块
Parent的构造函数...
Child的static代码块...
Parent的成员变量...
Parent的代码块
Parent的构造函数...
Child的成员变量...
Child的代码块...
Child的构造函数...
```

可以看出，类实例化的顺序是：当创建类对象时，，然后是非静态变量和非静态代码块，然后是构造。由于静态成员只会被初始化一次，所以如果静态成员已经被初始化过，将不会被再次初始化。

1. 第一梯队：父类的静态变量和静态块（与书写顺序有关，在前面的先实例化）
2. 第二梯队：子类的静态变量和静态块（与书写顺序有关，在前面的先实例化）
3. 第三梯队：父类的成员变量和块（与书写顺序有关，在前面的先实例化）
4. 第四梯队：子类的成员变量和块（与书写顺序有关，在前面的先实例化）

同时在例子可以看出，静态变量和静态块，在执行过一次之后就不再执行。

```
<hr>
<div style="text-align:center">
 </img>
</div>
<br>
<span>扫码关注微信公众号《Java学习交流社区》，即可获取原创《Java并发编程与高并发解决方
》+《Redis》思维导图</span>
<br>
作者：<a href="https://www.jvsc.cn" target="_blank">Java学习交流社区</a> <br>
地址：<a href="https://www.jvsc.cn/articles/2019/09/02/1567406392256.html" target="_bla
k">https://www.jvsc.cn/articles/2019/09/02/1567406392256.html</a> <br>
```