

java 类 继承时 static 构造块 构造器执行顺序

作者: [dj2kk](#)

原文链接: <https://ld246.com/article/1576121450520>

来源网站: [链滴](#)

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

先上demo:

```
public class A extends B{
    {
        System.out.println("1");
    }
    static {
        System.out.println("2");
    }
    A(){
        System.out.println("3");
    }
    public static void main(String[] args) {
        new A();
    }
}
```

```
public class B {
    static A a = new A();

    {
        System.out.println("4");
    }
    static {
        System.out.println("5");
    }
    B(){
        System.out.println("6");
    }
}
```

求程序执行最终结果.

- 初试

乍看代码 A类继承 B类,然后又不知道继承类与'构造块'&'构造器'&'static修饰的成员变量'的执行顺序我们可以尝试在脑子里运行一边。我这边直接解释java这块运行规则。然后再对对答案吧。

- 规则

java类的初始化

- 类初始化时先加载父类，而类的加载的优先级最高的是static静态修饰符
- static 按顺序重上到下执行
- 构造块>构造器（加载顺序）

- answer

4613524613

你答对了吗+1

- 知识扩展

双亲委派模式及优势

双亲委派模式的工作原理是:如果一个类加载器收到了类加载请求,它并不会自己先去加载,而是把这个请求委托给父类的加载器去执行,如果父类加载器还存在其父类加载器,则进一步向上委托,依次归,请求最终将到达顶层的启动类加载器,如果父类加载器可以完成类加载任务,就成功返回,倘若类加载器无法完成此加载任务,子加载器才会尝试自己去加载,这就是双亲委派模式,即每个儿子都愿意干活,每次有活就丢给父亲去干,直到父亲说这件事我也干不了时,儿子自己想办法去完成,这就是传说中的双亲委派模式.那么这种模式有什么作用呢?

answer

- 双亲委派模式优势.

- 采用双亲委派模式的好处是Java类随着它的类加载器一起具备了一种带有优先级的层次关系,通过这种层级关可以避免类的重复加载,当父亲已经加载了该类时,就没有必要子ClassLoader再加一次。其次是考虑到安全因素,java核心api中定义类型不会被随意替换,假设通过网络传递一个名为`java.lang.Integer`的类,通过双亲委托模式传递到启动类加载器,而启动类加载器在核心Java API发现个名字类,发现该类已被加载,并不会重新加载网络传递的过来的`java.lang.Integer`,而直接返回加载过的`Integer.class`,这样便可以防止核心API库被随意篡改。