



# Java 核心技术

## 第六章 static、final和常量设计

### 第一节 static

华东师范大学 陈良育

# static(1)



- static 静态的, Java中特殊的关键字
- Java中的static关键字(可作用在)
  - 变量
  - 方法
  - 类
  - 匿名方法块

# static(2)



- 静态变量，类共有成员
  - 先查看例子 Potato.java。
  - static变量只依赖于类存在（通过类即可访问），不依赖于对象实例存在。即可以通过Potato.price即可访问。
  - 所有的对象实例，如例子中的obj1和obj2关于price变量的值都共享存储在一个共同的空间（栈）。

# static(3)



- static 方法
  - 静态方法也无需通过对象来引用，而通过类名可以直接引用。
  - 在静态方法中，只能使用静态变量，不能使用非静态变量。
  - 静态方法禁止引用非静态方法。
  - 查看例子StaticMethodTest.java。
- 关于static修饰类(内部类)，使用的机会较少，暂忽略不计。



# static(4)



- static块
  - 只在类第一次被加载时调用。
  - 换句话说，在程序运行期间，这段代码只运行一次。
  - 执行顺序：static块 > 匿名块 > 构造函数。
- 参看StaticBlockTest例子

# static(5)



- 总结

- static变量：不管有多少个对象，内存中只有一份
- static方法：可以用类名直接引用，无需new对象来引用
- static块：static块只执行一次，并且static块 > 匿名块 > 构造函数

# 代码(1) Potato.java



```
public class Potato {
    static int price = 5;
    String content = "";
    public Potato(int price, String content)
    {
        this.price = price;
        this.content = content;
    }
    public static void main(String[] a)
    {
        System.out.println(Potato.price); //Potato.content wrong
        System.out.println("-----");
        Potato obj1 = new Potato(10, "青椒土豆丝");
        System.out.println(Potato.price);
        System.out.println(obj1.price);

        System.out.println("-----");
        Potato obj2 = new Potato(20, "酸辣土豆丝");
        System.out.println(Potato.price);
        System.out.println(obj2.price);

    }
}
```

# 代码(2)StaticMethodTest.java



```
public class StaticMethodTest {
    int a = 111111;
    static int b = 222222;
    public static void hello()
    {
        System.out.println("000000");
        System.out.println(b);
        //System.out.println(a); //error, cannot call non-static variables
        //hi() //error, cannot call non-static method
    }
    public void hi()
    {
        System.out.println("333333");
        hello(); //ok, call static methods
        System.out.println(a); //ok, call non-static variables
        System.out.println(b); //ok, call static variables
    }
    public static void main(String[] a)
    {
        StaticMethodTest.hello();
        //StaticMethodTest.hi(); //error, 不能使用类名来引用非静态方法
        StaticMethodTest foo = new StaticMethodTest();
        foo.hello(); //warning, but it is ok
        foo.hi(); //right
    }
}
```



# 代码(3)StaticBlock.java



```
class StaticBlock
{
    //static block > anonymous block > constructor function
    static
    {
        System.out.println("222222222222222222222222");
    }
    {
        System.out.println("111111111111111111111111");
    }
    public StaticBlock()
    {
        System.out.println("333333333333333333333333");
    }
    {
        System.out.println("444444444444444444444444");
    }
}
```

# 代码(4)StaticBlockTest.java



```
public class StaticBlockTest {  
  
    public static void main(String[] args) {  
        System.out.println("0000000000000000");  
        // TODO Auto-generated method stub  
        StaticBlock obj1 = new StaticBlock();  
        StaticBlock obj2 = new StaticBlock();  
    }  
}
```



谢谢!