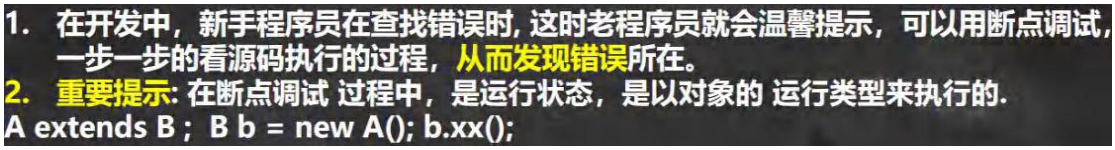
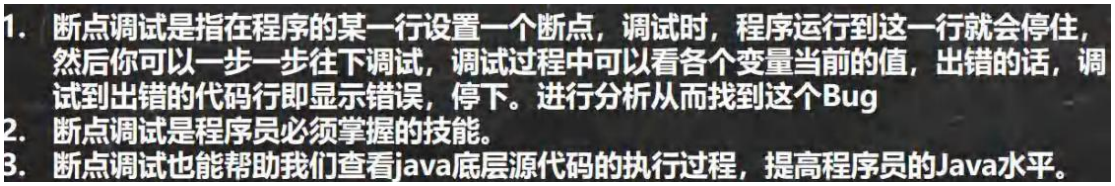
断点

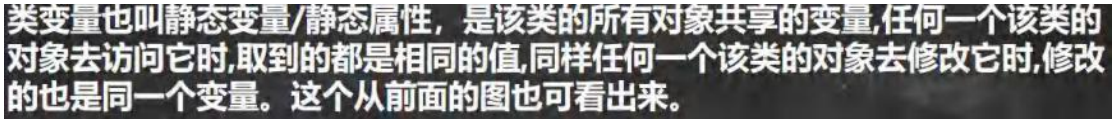




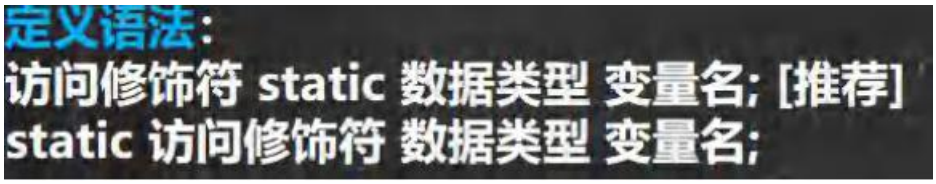
F7(跳入) F8(跳过)
shift+F8(跳出) F9(resume,执行到下一个断点)
F7：跳入方法内
F8: 逐行执行代码.
shift+F8:
跳出方法
//判断 对象是否属于类或者类的子类
对象名 instanceof 类名

零钱通能自己编写
房屋出租项目只能看懂还不能自己编写

第十章
类变量

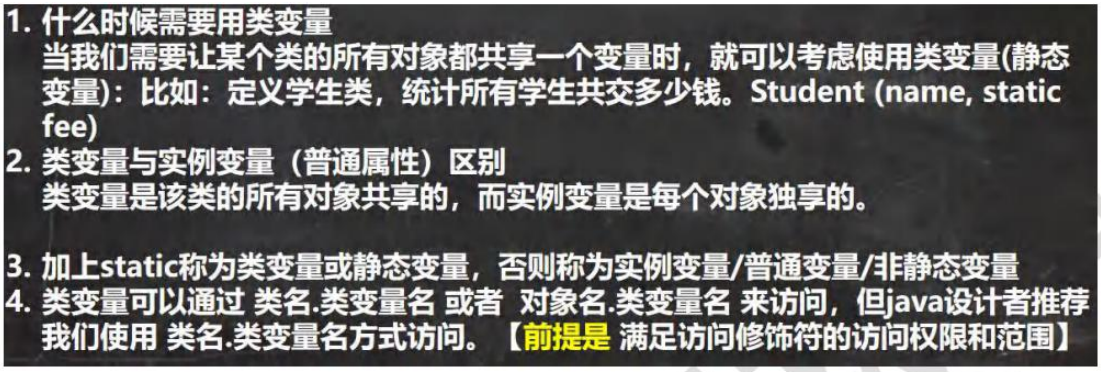


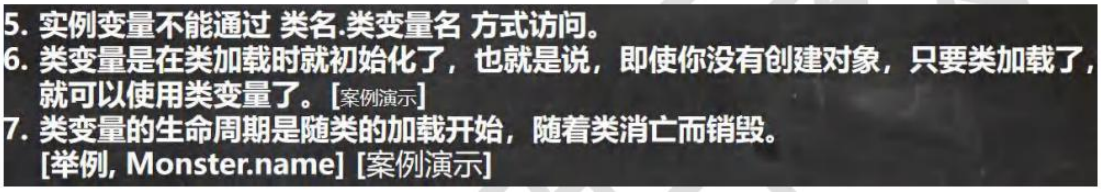
静态变量对对象共享
定义



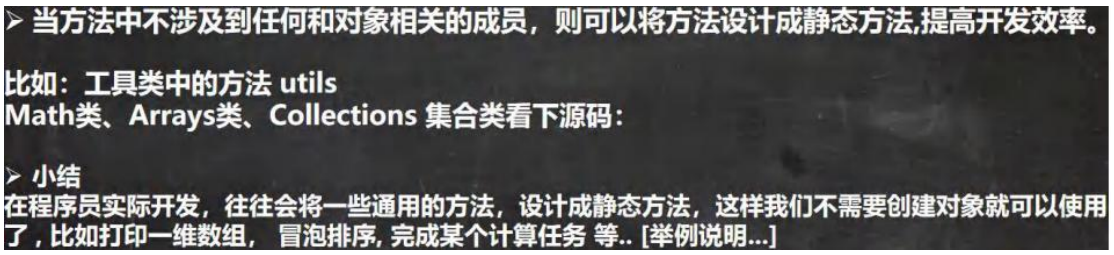
静态变量访问权限与普通的是一样的
可以通过对象访问也可以直接类调用属性
静态变量在类加载时就生成了（所以可以通过类调用）

类变量注意细节

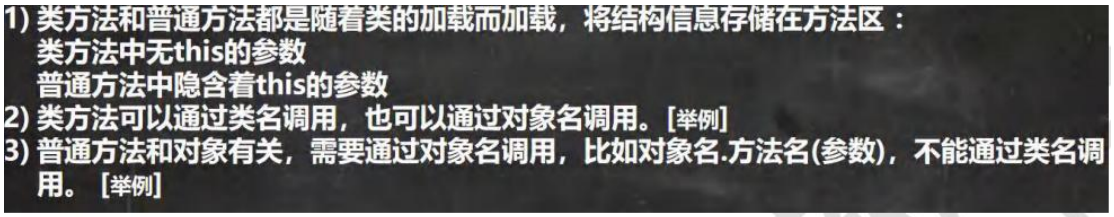


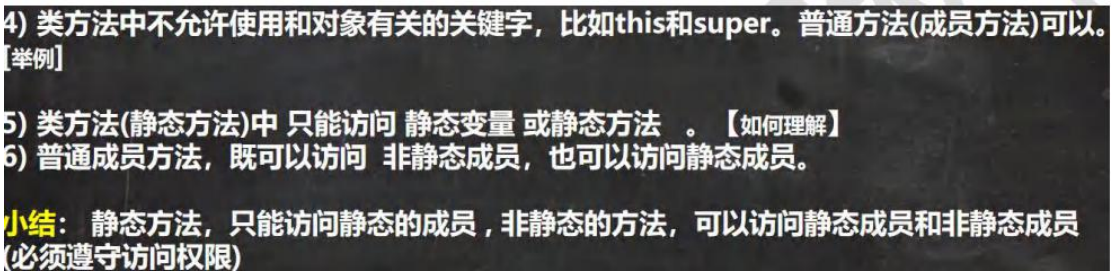


类方法

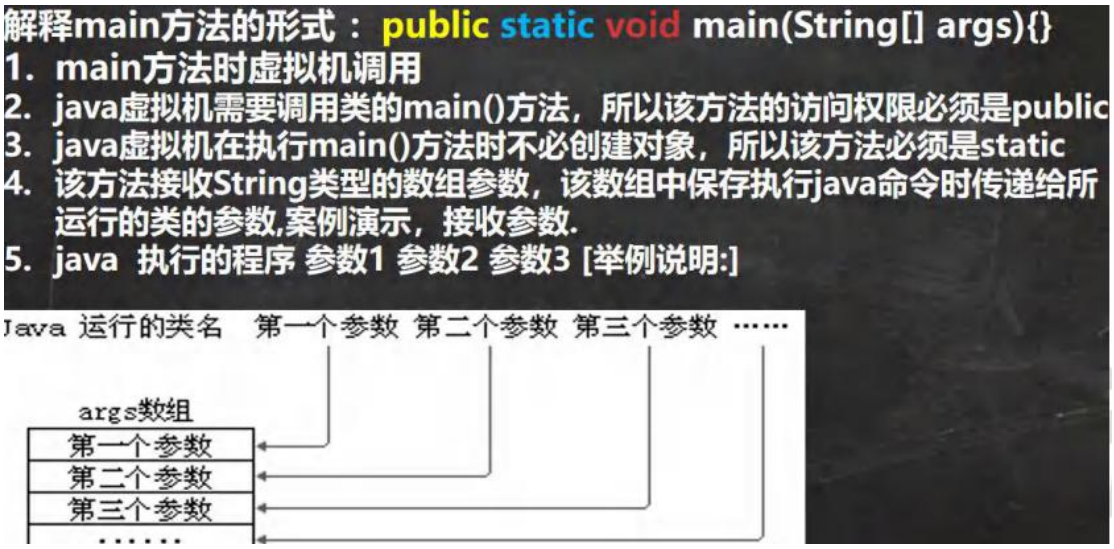


类方法使用细节





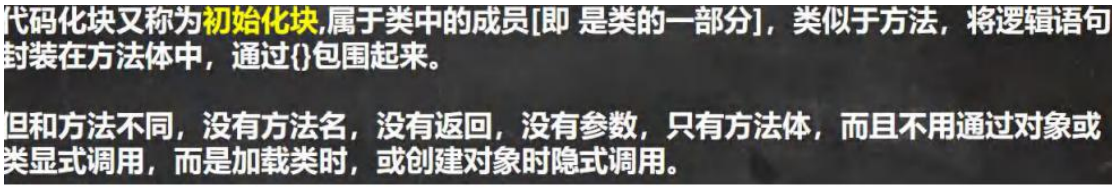
Main 方法



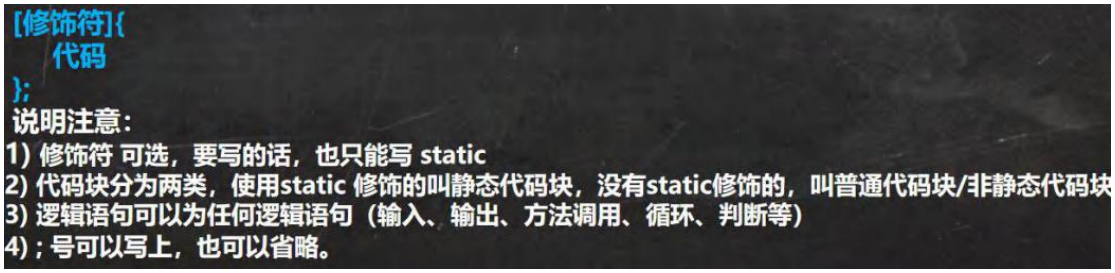
Jdk 中可直接给 args 设置参数

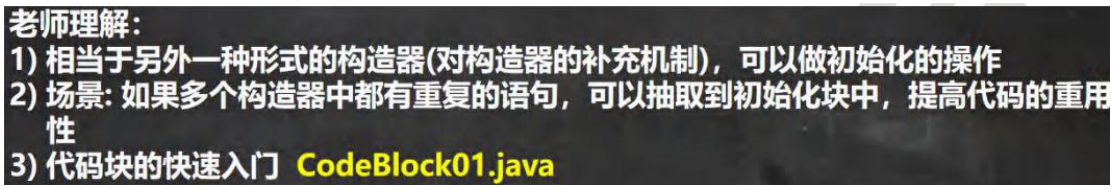
在 idea 中设置 configuration arguements

代码块



语法





代码块细节

1) static代码块也叫静态代码块，作用就是对类进行初始化，而且它随着类的加载而执行，并且只会执行一次。如果是普通代码块，每创建一个对象，就执行。
2) 类什么时候被加载[重要背!]
① 创建对象实例时(new)
② 创建子类对象实例，父类也会被加载
③ 使用类的静态成员时(静态属性，静态方法)
案例演示：A 类 extends B类 的静态块

3) 普通的代码块，在创建对象实例时，会被隐式的调用。
被创建一次，就会调用一次。
如果只是使用类的静态成员时，普通代码块并不会执行。
小结:1. static代码块是类加载时，执行，只会执行一次
2. 普通代码块是在创建对象时调用的，创建一次，调用一次
3. 类加载的3种情况，需要记住.

4) 创建一个对象时，在一个类 调用顺序是:(重点，难点)：
① 调用静态代码块和静态属性初始化(注意：静态代码块和静态属性初始化调用的优先级一样，如果有多个静态代码块和多个静态变量初始化，则按他们定义的顺序调用) [举例说明]

② 调用普通代码块和普通属性的初始化(注意：普通代码块和普通属性初始化调用的优先级一样，如果有多个普通代码块和多个普通属性初始化，则按定义顺序调用)
③ 调用构造方法。
新写一个类演示【CodeBlockDetail02.java】

5) 构造器 的最前面其实隐含了 super()和 调用普通代码块，新写一个类演示【截图+说明】,静态相关的代码块，属性初始化，在类加载时，就执行完毕，因此是优先于 构造器和普通代码块执行的 CodeBlockDetail03.java

```
class A {
  public  A() {//构造器
      //这里有隐藏的执行要求
      //(1) super();//这个知识点，在前面讲解继承的时候，老师说
      //(2) 调用普通代码块的
      System.out.println("ok");
  }
}
```

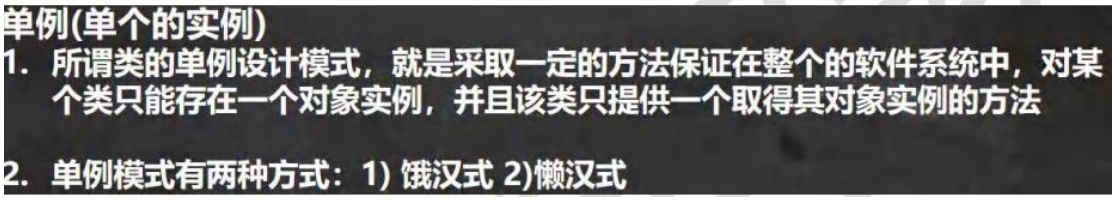
6) 我们看一下创建一个子类对象时(继承关系)，他们的静态代码块，静态属性初始化，普通代码块，普通属性初始化，构造方法的调用顺序如下:
① 父类的静态代码块和静态属性(优先级一样，按定义顺序执行)
② 子类的静态代码块和静态属性(优先级一样，按定义顺序执行)
③ 父类的普通代码块和普通属性初始化(优先级一样，按定义顺序执行)
④ 父类的构造方法
⑤ 子类的普通代码块和普通属性初始化(优先级一样，按定义顺序执行)
⑥ 子类的构造方法 // 面试题

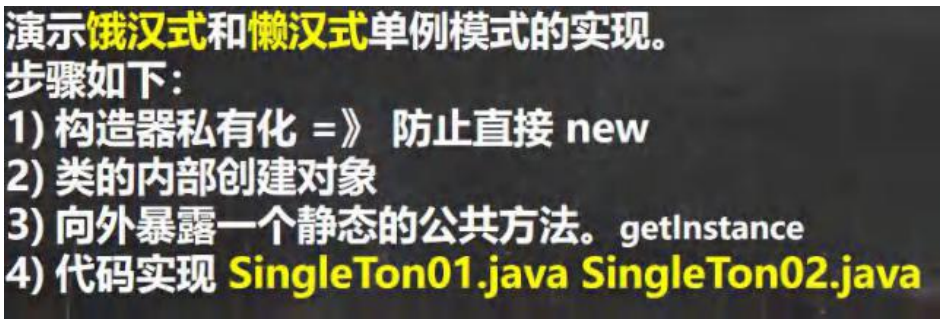
AAAAA extends BBBBB 类 演示 [10Min]55 CodeBlockDetail04.java

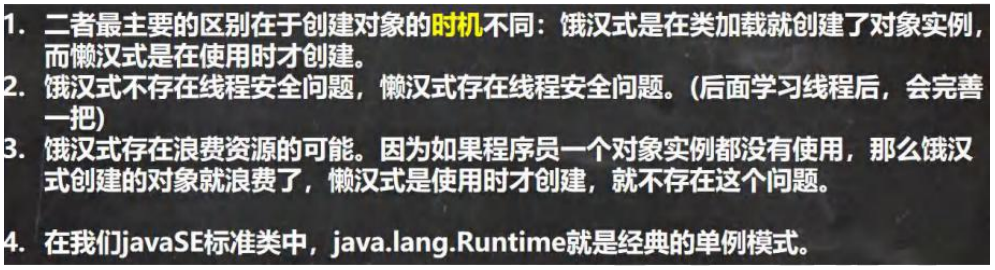
7) 静态代码块只能直接调用静态成员(静态属性和静态方法)，普通代码块可以调用任意成员。学习比较麻烦，工作轻松

直接调用静态变量时候不会加载普通代码块

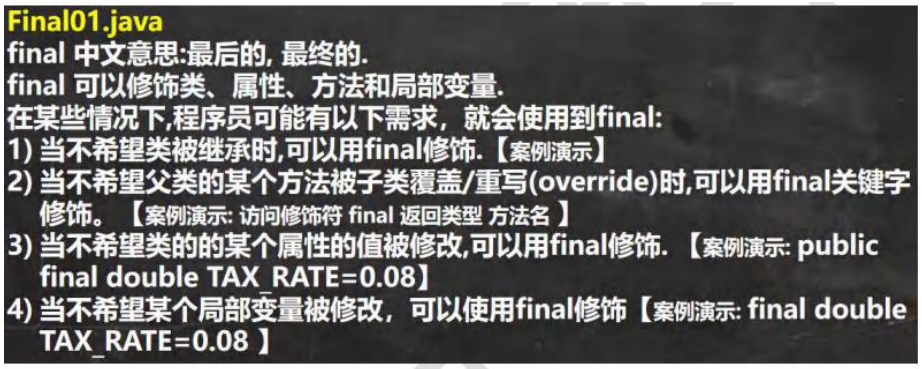
单例模式



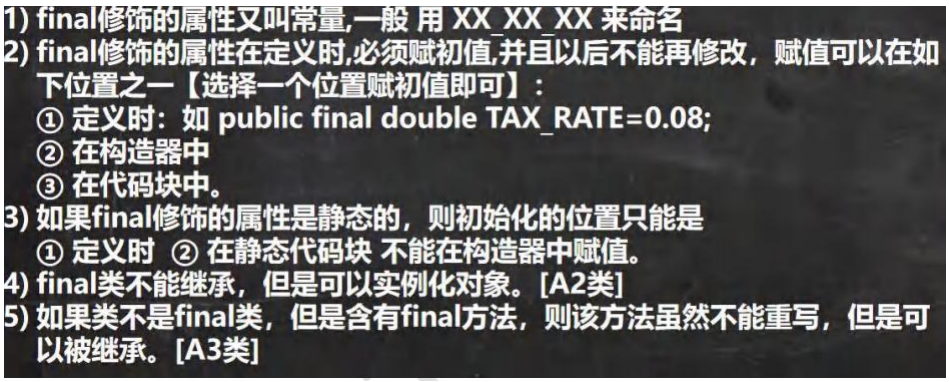




Final 关键字



细节



5) 一般来说，如果一个类已经是final类了，就没有必要再将方法修饰成final方法。
6) final不能修饰构造方法(即构造器)
7) final 和 static 往往搭配使用，效率更高，不会导致类加载.底层编译器做了优化处理。

```
class Demo{
        public static final int i=16; //
        static{
          System.out.println("韩顺平教育~");
        }
}
```

8) 包装类(Integer,Double,Float，Boolean等都是final),String也是final类。