

final 关键字，你想知道的都在这

Original 指北君 Java技术指北 5 days ago

收录于话题

#java 10 #JDK 19

PS:最近又赶上跳槽的高峰期，好多粉丝，都问我要有没有最新面试题，我连日加班好多天，终于整理好了，16000+ 道，295多份，多份面试题大全，我会持续更新中，马上就会整理更多！

【文末有领取方式】

回到正题，哈喽，大家好，我是指北君。

介绍完 native、static 关键字后，指北君马不停蹄，接着为大家介绍另一个常用的关键字——final。

对于Java中的 final 关键字，我们首先可以从字面意思上去理解，百度翻译显示如下：

final

英 ['faɪnəl] 美 ['faɪnəl]

adj. 最后的，最终的；决定性的；不可更改的

n. 决赛；结局；期末考试；〈口〉（报纸的）末版

复数：finals

记忆技巧：fin 结束 + al ...的 → 最后的；决定性的

高考

CET4

考研

也就是说 final 英文意思表示是最后的，不可更改的。那么对应在 Java 中也是表达这样的意思，可以用 final 关键字修饰变量、方法和类。不管是用来修饰什么，其本意都是指“它是无法更改的”，这是我们需要牢记的，为什么要无法更改？无非就是设计所需或者能提高效率，与前面介绍 static 关键字需要记住其与对象无关的理念一样，牢记 final 的不可变的设计理念后再来了解 final 关键字的用法，便会顺其自然了。

1、修饰变量

稍微有点Java基础的都知道用final关键字修饰的变量称为常量，常量的意思是不可更改。变量为基本数据类型，不可更改很容易理解，那么对于引用类型呢？不可能改的是其引用地址，还是对象的内容？

我们首先构造一个实体类：Person

```
package com.js.bean;

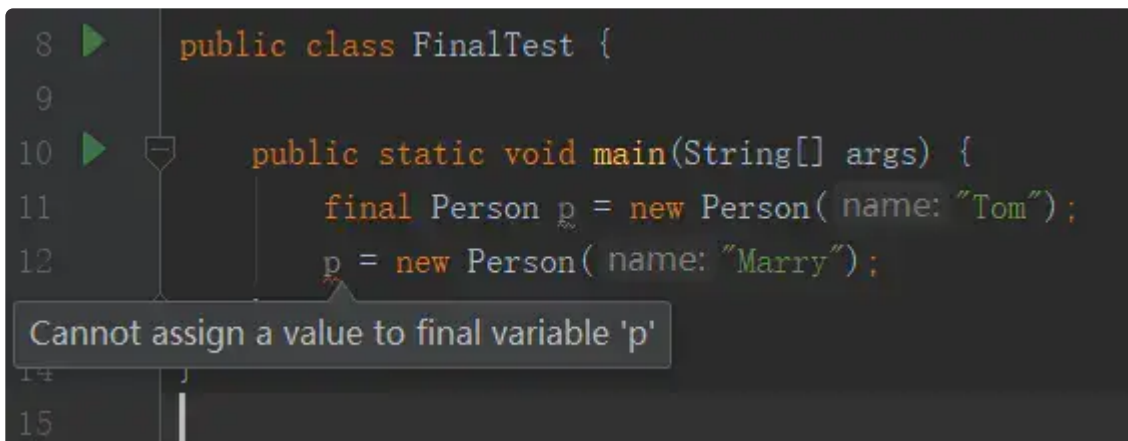
/**
 * Create by YSOcean
 */
public class Person {
    private String name;

    public Person(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

接着根据创建一个 Person 对象：



```
8  public class FinalTest {
9
10  public static void main(String[] args) {
11      final Person p = new Person( name: "Tom");
12      p = new Person( name: "Marry");
13  }
14
15
```

Cannot assign a value to final variable 'p'

可以看到，首先通过 `final` 关键字修饰一个对象 `p`，然后接着将 `p` 对象指向另一个新的对象，发现报错，也就是说`final`修饰的引用类型是不能改变其引用地址的。

接着我们改动 `p` 对象的 `name` 属性：

```
8  ▶ public class FinalTest {
9
10 ▶  public static void main(String[] args) {
11     final Person p = new Person( name: "Tom");
12     p.setName("Marrp");
13 }
14 }
```

发现程序没有报错。

结论：被 `final` 修饰的变量不可更改其引用地址，但是可以更改其内部属性。

2、修饰方法

`final` 关键字修饰的方法不可被覆盖。

在《Java编程思想》第 4 版 7.8.2 章节 `final` 方法p176 页这样描述：使用 `final` 方法原因有两个：

①、第一个原因是把方法锁定，以防止任何继承类修改它的含义，这是出于设计的考虑：想要确保在继承中使方法的行为保持不变，并且不会被覆盖。

②、第二个原因是效率，在 Java 的早期实现中，如果将一个方法声明为 `final`，就是同意编译器针对该方法的所有调用都转为内嵌调用，内嵌调用能够提高方法调用效率，但是如果方法很大，内嵌调用不会提高性能。而在目前的Java版本中（JDK1.5以后），虚拟机可以自动进行优化了，而不需要使用 `final` 方法。

所以`final` 关键字只有明确禁止覆盖方法时，才使用其修饰方法。

PS: 《Java编程思想》中指出类中所有的 private 方法都隐式指定为 final 的, 所以对于 private 方法, 我们显式的声明 final 并没有什么效果。但是我们创建一个父类, 并在父类中声明一个 private 方法, 其子类中是能够重写其父类的private 方法的, 这是为什么呢?

父类: Parent.class

```
package com.js.bean;
/**
 * Create by YSOcean
 */
public class Parent {
    private void say(){
        System.out.println("parent");
    }
}
```

子类: Son.class

```
package com.js.bean;
/**
 * Create by YSOcean
 */
public class Son extends Parent {

    private void say(){
        System.out.println("son");
    }

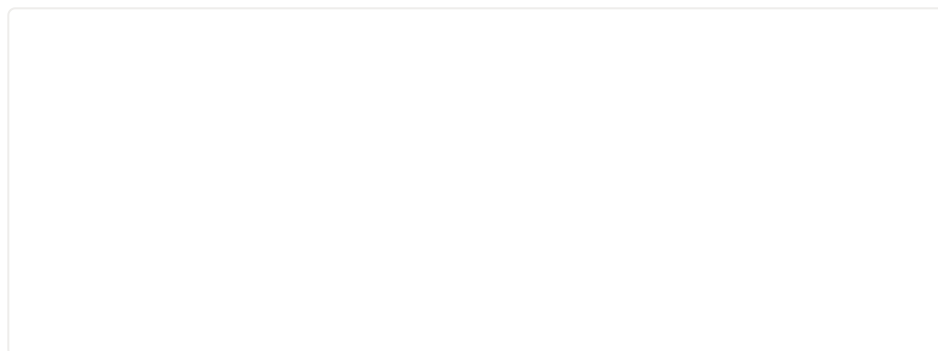
}
```

其实仔细看看, 这种写法是方法的覆盖吗? 我们通过多态的形式并不能调用到父类的 say() 方法:

```
11  
12 public static void main(String[] args) {  
13     Parent s = new Son();  
14     s.say();  
15 }
```

'say()' has private access in 'com.js.bean.Parent'

并且，如果我们在子类的 say() 方法中，添加 @Override 注解也是会报错的。



所以这种形式并不算方法的覆盖。

3、修饰类

final 修饰类表示该类不可被继承。

也就是不希望某个类有子类的时候，用 final 关键字来修饰。并且由于是用 final 修饰的类，其类中所有的方法也被隐式的指为 final 方法。

在 JDK 中有个最明显的类 String，就是用 final 修饰的，将 String 类用 final 修饰很重要的一个原因是常量池。关于 String 类的描述，可以参考我前面文章。

4、小结

好了，这就是Java中对 final 关键词的介绍，总之记住一句话，final 修饰的就是不可变的。指北君后续的文章会给大家介绍Java中的各种关键字。

我是指北君，操千曲而后晓声，观千剑而后识器。感谢各位人才的：点赞、收藏和评论，我们下期更精彩！

面试大全包括：包括 Java 集合、JVM、多线程、并发编程、设计模式、SpringBoot、SpringCloud、Java、MyBatis、ZooKeeper、Dubbo、Elasticsearch、Memcached、MongoDB、Redis、MySQL、RabbitMQ、Kafka、Linux、Netty、Tomcat、Python、HTML、CSS、Vue、React、JavaScript、Android 大数据、阿里巴巴等大厂面试题等、等技术栈！

领取方式：扫描下方二维码【Java技术指北】回复【**面试大全**】即可获取



Java技术指北

回复：java，获取精华资料。专注分享Java技术干货、Java 技术、Spring 全家桶、Java...
47篇原创内容

Official Account

People who liked this content also liked

学习VB是为了操作Excel?! 不然还能干啥!

兴趣使然的码农

Matlab与科学计算——数值计算（上）

50个Python学习资源，从初学者到高级玩家都有了！

GOGO数据