

47从零开始学Java47之final修饰符、常量、常量方法与常量类

前言

一. final修饰符

1. 简介
2. 特性
3. 配套视频

二. 常量

1. 概念
2. 特性
3. 语法
4. 案例
 - 4.1 修饰局部变量
 - 4.2 修饰成员变量
 - 4.3 修饰基本类型的变量
 - 4.4 修饰引用类型的变量

三. 常量方法

1. 概念
2. 语法
3. 案例
 - 3.1 子类不能重写父类的final方法
 - 3.2 父类中私有的final方法

4. 配套视频

四. 常量类

1. 概念
2. 语法
3. 实现案例
4. 配套视频

五. 结语

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

前言

壹哥之前跟大家说过，在面向对象中，有abstract、static和final 这3个核心修饰符。截止到现在，我们已经把abstract与static修饰符学习完毕，接下来就让我们再来学习final修饰符的用法与特性吧。

-----前戏已做完，精彩即开始-----

全文大约【**3500**】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富的案例及配图视频，让你更好地理解 and 运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

一. final修饰符

1. 简介

在Java中，final表示“**最终的、不可改变的、完结的**”，它也是一种修饰符，可以**修饰变量、方法和类**。final修饰变量、方法和类时的意义是不同的，但本质是一样的，都表示不可改变，类似C#里的sealed关键字。final修饰的变量叫做**最终变量，也就是常量**，修饰的方法叫做**最终方法**，修饰的类叫做**最终类**。

2. 特性

3. 配套视频

与本节内容配套的视频链接如下：

二. 常量

1. 概念

被final修饰的变量一旦被赋值初始化后，就不能再被重新赋值。即变量值只能被赋值一次，不可被反复修改，所以叫做最终变量，也叫做常量。

并且我们在定义final变量时，必须显式地进行初始化，指定初始值，否则会出现“The blank final field xxx may not have been initialized”的异常提示。变量值的初始化，可以在两个地方：一是在变量定义处，即在final变量定义时直接给其赋值；二是在构造方法或静态代码块中。这些地方只能选其一，不能同时既在定义时赋值，又在构造方法或静态代码块中另外赋值。

我们在开发时，通常会把final修饰符和static修饰符一起使用，来创建类的常量。

2. 特性

根据修饰变量的作用范围，比如在修饰局部变量和成员变量时，final会有不同的特性：

- final修饰局部变量时，在使用之前必须被赋值一次才能使用；
- final修饰成员变量时，如果在声明时没有赋值，则叫做“空白final变量”，空白final变量必须在构造方法或静态代码块中进行初始化。

根据修饰变量的数据类型，比如在修饰基本类型和引用类型的变量时，final也有不同的特性：

- final修饰基本类型的变量时，不能把基本类型的值重新赋值，因此基本类型的变量值不能被改变。
- final修饰引用类型的变量时，final只会保证引用类型的变量所引用的地址不会改变，即保证该变量会一直引用同一个对象。因为引用类型的变量保存的仅仅是一个引用地址，所以final修饰引用类型的变量时，该变量会一直引用同一个对象，但这个对象本身的成员和数据是完全可以发生改变的。

3. 语法

final修饰变量时，常用的语法格式如下：

```
final String 变量名=变量值;
```

我们在使用final声明变量时，一般会要求变量名的单词全部大写，且变量名由多个单词组成时，多个单词之间用下划线“_”分隔开，比如“*SCHOOL_NAME*”。

4. 案例

4.1 修饰局部变量

以下案例是final修饰局部变量时的用法和特性。

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class FinalDemo {
7
8      // final修饰局部变量，该变量使用之前要赋初值
9      public void declareFinal() {
10         // 先声明变量
11         final int x;
12         // 再赋初值，该值只能赋一次，否则会报错。
13         x = 200;
14         //The final local variable x may already have been assigned
15         //x = 400;
16         System.out.println("x=" + x);
17
18         // 声明的同时赋值
19         final int y = 300;
20         System.out.println("y=" + y);
21     }
22
23 }
```

从上述案例中可知，局部常量最好是在声明时就进行初始化。而且常量只能被赋值一次，否则会出现编译错误：“The final local variable xxx may already have been assigned”。

4.2 修饰成员变量

以下案例是final修饰成员变量时的用法和特性。

```
Java | 复制代码
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class FinalDemo {
7      //final修饰成员变量
8
9      // 实例常量
10     final int a = 10; // 直接赋值
11     final int b; // 空白final变量，需要在构造方法中进行初始化
12
13     // 静态常量
14     final static int c = 20; // 直接赋值
15     final static int d; // 空白final变量，需要在静态代码块中进行初始化
16
17     static {
18         // 初始化静态变量
19         d = 40;
20     }
21
22     FinalDemo() {
23         // 初始化成员变量
24         b = 20;
25         // 不能第二次赋值，否则会发生编译错误
26         // The final local variable b may already have been assigned
27         //b = 30;
28     }
29
30 }
```

从上述代码中可知，空白final变量，可以在构造方法或静态代码块中初始化。

4.3 修饰基本类型的变量

以下案例是final修饰基本类型变量时的用法和特性。**final修饰基本类型的变量时**，不能把基本类型的变量重新赋值，因此基本类型的变量值不能被改变，否则会出现“The final local variable x cannot be assigned. It must be blank and not using a compound assignment”的异常。

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class FinalTypeDemo {
7
8      public static void main(String[] args) {
9          //final修饰基本类型的变量，变量值不可变
10         final int x=10;
11
12         //The final local variable x cannot be assigned. It must be blank
13         and not using a compound assignment
14         //x=20;
15         System.out.println("x="+x);
16     }
17 }
```

4.4 修饰引用类型的变量

以下案例是final修饰引用类型变量时的用法和特性。**final修饰引用类型的变量时**，final只会保证引用类型的变量所引用的地址不会改变，即保证该变量会一直引用同一个对象，否则会出现“Array constants can only be used in initializers”或者“The final local variable user cannot be assigned. It must be blank and not using a compound assignment”的异常。

```
1  import java.util.Arrays;
2
3  /**
4   * @author 一一哥Sun
5   * QQ: 2312119590
6   * CSDN、掘金、知乎找我哦
7   */
8  public class FinalTypeDemo {
9
10     public static void main(String[] args) {
11         // final修饰数组变量, nums是一个引用变量
12         final int[] nums = { 1, 9, 7, 3 };
13         System.out.println(Arrays.toString(nums));
14
15         //final修饰引用类型时, 引用的地址不可变, 但引用对象本身的数据内容是可变的
16         //Array constants can only be used in initializers
17         //nums= {2,0,8,1};
18
19         // 对数组里的元素赋值修改没问题
20         nums[2] = 10;
21         System.out.println(Arrays.toString(nums));
22
23         // final修饰Person变量, p是一个引用变量
24         final User user = new User();
25         // 改变Person对象的age实例变量, 合法
26         user.setAge(18);
27         System.out.println(user.getAge());
28
29         //对user变量的引用地址重新赋值, 非法
30         //The final local variable user cannot be assigned. It must be bla
31         nk and not using a compound assignment
32         //user = new User(30);
33     }
34 }
```

从上面的两个案例中, 我们可以得知, 使用 final修饰引用类型的变量时, 变量不能被重新赋值, 但我们可以改变该变量所引用对象里的内容。

三. 常量方法

1. 概念

被final修饰的方法称为常量方法，该方法可以被重载，也可以被子类继承，但却不能被重写。当一个方法的功能已经可以满足当前要求，不需要进行扩展，我们就用任何子类来重写该方法，防止该方法的内容被修改。比如Object类中，就有一个final修饰的getClass()方法，Object的任何子类都不能重写这个方法。

2. 语法

final修饰方法时，常用的语法格式如下：

```
修饰符 final 返回值类型 方法名(){  
    //方法体  
}
```

3. 案例

3.1 子类不能重写父类的final方法

如果我们在父类中定义一个final方法，子类继承父类后，**子类不能重写父类中的这个final方法**，否则会出现“Cannot override the final method from Father”异常。但是我们要注意，final方法是可以被重载的！


```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class Father {
7
8      private String name;
9
10     public final void setName(String name) {
11         this.name=name;
12     }
13
14     //重载final方法
15     public final void setName(String firstName,String lastName) {
16         this.name=firstName+lastName;
17     }
18 }
19
20
21 //子类继承父类
22 public class Son extends Father{
23
24     private String name;
25
26     //子类不能重写父类中的final方法!
27     //Cannot override the final method from Father
28     //public void setName(String name) {
29     //    this.name=name;
30     //}
31
32 }
```

3.2 父类中私有的final方法

如果我们在父类中定义了一个私有的private方法，因为它只能在当前类中可见，其子类无法访问该方法，所以子类也就无法重写该方法。如果子类中也定义了一个与父类中完全相同的private方法，这也不算方法重写，这只是重新定义了一个新方法。因此，即使我们使用final修饰private方法，我们也可以在子类中定义与该方法相同的方法。

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class Father {
7
8      private double salary;
9
10     //父类中私有的方法
11     private final void setSalary(double salary) {
12         this.salary=salary;
13     }
14
15 }
16
17 //子类继承父类
18 public class Son extends Father{
19
20     private double salary;
21
22     //子类中定义与父类同名通参的方法，这不属于方法重写！
23     private void setSalary(double salary) {
24         this.salary=salary;
25     }
26
27 }
```

4. 配套视频

与本节内容配套的视频链接如下：

<https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=96&page=96>

四. 常量类

1. 概念

我们知道，通常子类继承父类时，子类可以访问父类的内部数据，并可通过重写父类的方法来改变父类方法的实现细节。但这样做可能会导致一些不安全的因素，所以为了保证某个类不可被继承，我们就可以使用final来修饰这个类。**被final修饰的类称为常量类，这种类不能被继承，也就不能被修改或扩展。**

final类无法被任何其他类继承，这意味着该类在Java的继承树体系中是一个叶子类，比如我们经常使用的**String类**，就是典型的final类。如下图所示：

```
1 public final class String
2 implements java.io.Serializable, Comparable<String>, CharSequence,
3             Constable, ConstantDesc {
```

而final类中的成员，我们可以用final修饰，也可以不用final修饰。比如final类中的方法，因为这些方法本身就属于final类，该类都不能被子类继承，里面的所有方法自然也就不能被子类重写，那么这些方法自然也就成了final型。所以final中的变量和方法，我们都没必要再单独添加final修饰符了。

2. 语法

final修饰类时，常用的语法格式如下：

```
访问修饰符 final class 类名{
    //....
}
```

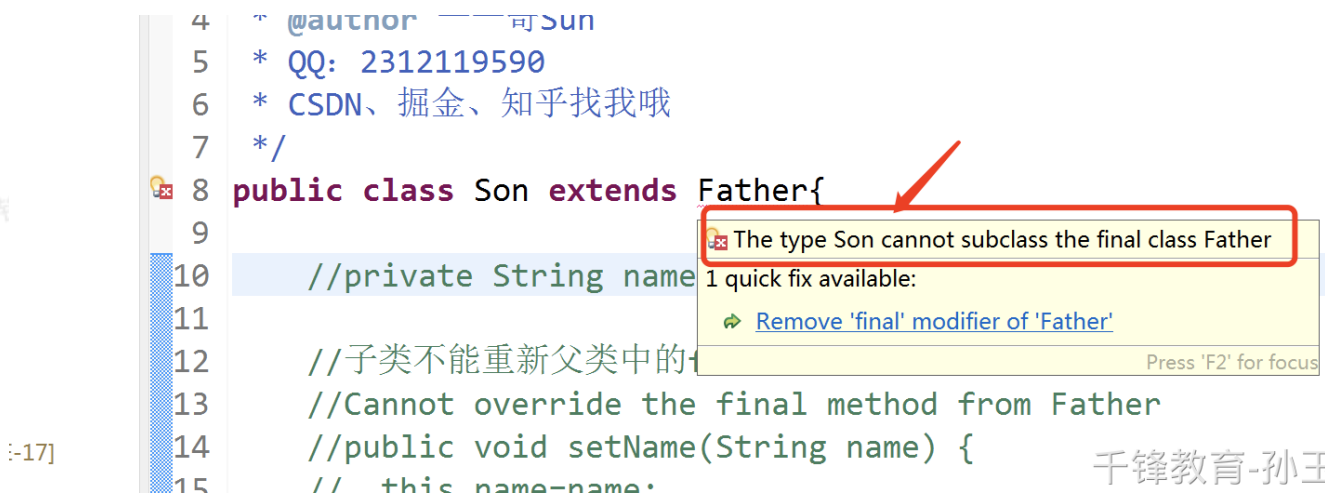
3. 实现案例

```

1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   *
6   * 父类是final类
7   */
8  public final class Father {
9
10     .....
11 }
12
13 //此时子类不能继承父类，否则会出错！
14 public class Son extends Father{
15
16     .....
17 }

```

从上面的代码中我们可以验证得知，final类不能被继承，final类中的方法更不可能被重写，否则会出现“The type Son cannot subclass the final class Father”异常，如下图所示：



千锋教育-孙玉昌

4. 配套视频

与本节内容配套的视频链接如下：

<https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=95&page=95>

-----正片已结束，来根事后烟-----

五. 结语

至此，壹哥就把final与常量、常量方法、常量类等相关内容讲解完毕了，现在你知道final的特性都有哪些吗？最后壹哥给大家总结如下：

- final修饰的变量，其变量值不可被改变，此时的变量被称为常量；
- final修饰引用类型的变量时，引用地址不可变，但对象中的数据可变；
- final修饰的方法不可以被重写；
- final修饰的类不可以被继承，即不能有子类。

另外如果你独自学习觉得有很多困难，可以加入壹哥的学习互助群，大家一起交流学习。

六. 今日作业

Java | 复制代码

```
1 class MyClass{
2     final int value;
3
4     public MyClass(){
5     }
6
7     public MyClass(int value){
8         this.value=value;
9     }
10 }
11
12 public class TestMain{
13     public static void main(String args[]){
14         MyClass mc=new MyClass(10);
15         System.out.println(mc.value);
16     }
17 }
```

根据上述代码，选择正确答案：

- A. 编译通过，输出10；
- B. 编译不通过，把第2行挨骂改为 `final int value=10;`
- C. 编译不通过，把第4行代码改为 `public MyClass(){value=10;}`