

笔记：JAVA学习之“复用类” Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/585/2021\\_2022\\_\\_E7\\_AC\\_94\\_E8\\_AE\\_B0\\_EF\\_BC\\_9AJ\\_c104\\_585518.htm](https://www.100test.com/kao_ti2020/585/2021_2022__E7_AC_94_E8_AE_B0_EF_BC_9AJ_c104_585518.htm)

1.toString ( ) 每一个非基本类型都有一个toString ( ) 方法；当编译器需要从对象获取一个string时，该对象的toString ( ) 方法就会被调用。示例：

```
class WaterSource { private String s. WaterSource() { s = "WaterSource Constructed". } public String toString() {return s.} public static void main(String[] args) { WaterSource w = new WaterSource(). System.out.println(w). } }
```

打印的结果是：WaterSource Constructed 2.继承 当创建一个类时，总是在继承。因此，除非已经明确指出要从其他类中继承，否则就是在隐式的从JAVA的标准根类Object进行继承。继承的关键字

：extends 关键字super表示超类（或者基类）的意思。因此，表达式super.scrub ( ) 将调用超类的scrub ( ) 方法。 3.初始化

基类 在构造器中调用基类构造器来执行初始化，而基类构造器具有执行基类初始化所需要的所有知识和能力。对于无参数的基类构造器，会被派生类默认调用；对于带参数的基类构造器，派生类必须使用关键字super显示地调用，并配以适当的参数。

4.try与finally try { ... } finally { ... } 关键字try表示，下面的块（用{}括起来的代码）是所谓的保护区，这意味着它需要被异常处理。其中一项特殊的处理是无论try块怎样退出的，保护区后的finally子句中的代码总是要被执行的。这里的finally子句表示的是“无论发生什么事，一定要执行后面跟着的块”。

5.protected关键字 protected关键字指明“就类用户而言，这是private的，但对于任何继承于此类的导出类或者

其他任何位于同一个包内的类来说，它是可以访问的”。6. 向上转型“为新的类提供方法”并不是继承技术中最重要 的方面，其最重要的方面是用来表现新类和基类之间的关系。这种关系可以用“新类是现有类的一种类型”加以概括。由于继承可以确保基类中所有的方法在到处类中也同样有效，所以能够向基类发送的所有信息同样也向可以导出类发送。

示例 

```
class Instrument { public void play () {
System.out.println("Instrument play"). } static void tune (Instrument
i) { i.play(). } } public class Wind extends Instrument { public static
void main(String[] args) { Wind flute = new Wind().
```

```
Instrument.tune(flute). //upcasting } }
```

在tune中，程序代码可以对Instrument和它所有的到处类起作用，这种将Wind引用转换为Instrument引用的动作，称之为“向上转型”。7.继承使用慎用继承，其使用场合仅限于你确信使用该技术确实有效的情况。到底是改用组合还是使用继承，一个最清晰的判断方法就是问一问自己是否需要从新类向基类进行向上转型。如果必须向上转型，则继承是必要的；但如果不需要，则应当好好考虑自己是否需要继承。8.final关键字 用final关键字定义编译期常量：这种常量必须是基本类型，并且以关键字final表示。定义常量时，必须赋值。按照惯例：既是static又是final的常量将用大写表示：定义为public，则可以被用于包之外；定义为static，则强调只有一份；定义为final，则说明它是一个常量。当final关键字后定义的域是对象引用时，final使引用恒定不变（一旦引用被初始化指向一个对象，就无法再把它改为指向另一个对象）。然而，对象其自身确实是可以修改的。空白final：空白final指被声明为final但又未给定初

值的域。无论什么情况，编译器都确保空白final在使用前必须被初始化。但是，空白 final在关键字final的使用上提供了更大的灵活性。为此，一个类中的final域就可以做到根据对象而有所不同，却又保持其恒定不变的特性。

**final参数：**JAVA允许在参数列表中以声明的方式将参数指明为final.这意味着你无法在方法中更改参数引用所指向的对象。

**final方法：**把方法锁定，以防止任何继承类修改它的含义。类中所有的private方法都隐式地指定为final的。由于无法取用private方法，所以就无法覆盖它。可以对private方法添加final修饰词，但这并不能给该方法增加任何额外的意义。

**final类：**将某个类定义为final时，表明了你不打算继承该类，而且也不允许别人这样做。

### 9.初始化及类的加载

一般来说，类的代码在初次使用时才加载。这通常指加载发生于创建类的第一个对象之时，但是当访问static域或者static方法时，也会发生加载。当一个类在加载时，编译器注意到它有一个基类，则加载完该类，继续加载基类。如果基类还有其自身的基类，则继续加载基类的基类。依次类推。类加载完成之后，根基类的static初始化会被执行，然后是下一个到处类的staic初始化被执行，依次类推。至此，类的加载完毕，可以创建对象了。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)