

EffectiveJavaItem5在可以重用对象的时候避免创建对象计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/641/2021\\_2022\\_EffectiveJ\\_c97\\_641697.htm](https://www.100test.com/kao_ti2020/641/2021_2022_EffectiveJ_c97_641697.htm) 一般情况下，最好重用功能上对等的对象，而不是创建新的对象。重用对象可以提高性能，并且样式良好。

如果对象是immutable的(private fields, no setter methods)，那么重用起来就比较方便。不良例子：`String name = new String(`

`“ HelloWorld ”)`。每次执行时，都会创建一个新的String对象。其中构造方法的参数本身就是一个对象，这样的构造方法没有必要，如果这样的语句出现在循环中，将会导致非常多的不必要的对象的创建。修正：`String name = “ HelloWorld`

`”`。该语句创建了单一的对象。这个对象还可以被之后运行在同一个虚拟机总的其他使用相同字符串的代码重用。

```
package com.googlecode.javatips4u.effectivejava.unnecessary. public class UnnecessaryObjectsSample { public static void main(String[] args) { long nanoTime = System.nanoTime(). String name = null. for (int i = 0. i lt. 10000. i ) { // CHANGE TO THIS name = "name". }
```

```
System.out.println(System.nanoTime() - nanoTime).// 867987 System.out.println(name). } }
```

```
可以使用工厂方法来避免创建不必要的对象。 package com.googlecode.javatips4u.effectivejava.unnecessary. import java.util.Calendar. import java.util.Date. import java.util.TimeZone. public class BadMutableObjectUsage { private final Date birthDate = null. // DONT DO THIS! public boolean isBabyBoomer() { //
```

```
Unnecessary allocation of expensive object // Here the two Date objects are immutable. Calendar gmtCal =
```

```
Calendar.getInstance(TimeZone.getTimeZone("GMT")).
gmtCal.set(1946, Calendar.JANUARY, 1, 0, 0, 0). Date boomStart =
gmtCal.getTime(). gmtCal.set(1965, Calendar.JANUARY, 1, 0, 0,
0). Date boomEnd = gmtCal.getTime(). return
birthDate.compareTo(boomStart) &lt; 0. } }
```

上面例子中两个Date对象，即开始时间和结束时间是常量，所以不需要在isBabyBoomer方法中每次被调用时都进行创建，而应该作为类的静态变量进行static初始化。

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)