## Java 代码（java version "1.8.0\_201"）

|  |
| --- |
| package com.test;  import java.time.LocalDate;  import java.time.format.DateTimeFormatter;  import java.util.concurrent.atomic.AtomicReference;  /\*\*  \* @Auther: SunBC  \* @Date: 2019/1/15 12:29  \* @Description:  \*/  public class test {  public void testLettureRedis(){  String str = "1"+"2"+"3";  String str1 = "4";  String str2 = str + str1;  String str3 = "1234";  String str4 = str1+"4";  String Str5 = str1 + new String("4");  System.out.println(str == str2);//false  System.out.println(str == str4); //false  System.out.println(str == str5); //false  }  } |

##### 反编译后的代码：

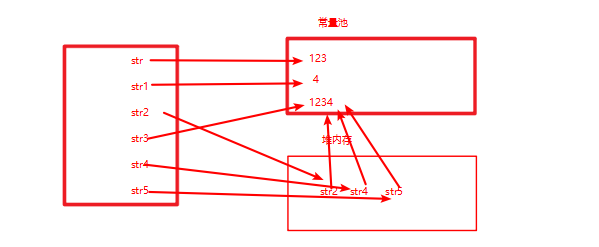
|  |
| --- |
| Classfile /C:/Users/admin/test.class  Last modified 2019-4-18; size 578 bytes  MD5 checksum 4abd7d22d817ac12aa8cd979aa021936  Compiled from "test.java"  public class com.test.test  minor version: 0  major version: 52  flags: ACC\_PUBLIC, ACC\_SUPER  Constant pool:  #1 = Methodref #12.#20 // java/lang/Object."<init>":()V  #2 = String #21 // 123  #3 = String #22 // 4  #4 = Class #23 // java/lang/StringBuilder  #5 = Methodref #4.#20 // java/lang/StringBuilder."<init>":()V  #6 = Methodref #4.#24 // java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  #7 = Methodref #4.#25 // java/lang/StringBuilder.toString:()Ljava/lang/String;  #8 = String #26 // 1234  #9 = Class #27 // java/lang/String  #10 = Methodref #9.#28 // java/lang/String."<init>":(Ljava/lang/String;)V  #11 = Class #29 // com/test/test  #12 = Class #30 // java/lang/Object  #13 = Utf8 <init>  #14 = Utf8 ()V  #15 = Utf8 Code  #16 = Utf8 LineNumberTable  #17 = Utf8 testLettureRedis  #18 = Utf8 SourceFile  #19 = Utf8 test.java  #20 = NameAndType #13:#14 // "<init>":()V  #21 = Utf8 123  #22 = Utf8 4  #23 = Utf8 java/lang/StringBuilder  #24 = NameAndType #31:#32 // append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  #25 = NameAndType #33:#34 // toString:()Ljava/lang/String;  #26 = Utf8 1234  #27 = Utf8 java/lang/String  #28 = NameAndType #13:#35 // "<init>":(Ljava/lang/String;)V  #29 = Utf8 com/test/test  #30 = Utf8 java/lang/Object  #31 = Utf8 append  #32 = Utf8 (Ljava/lang/String;)Ljava/lang/StringBuilder;  #33 = Utf8 toString  #34 = Utf8 ()Ljava/lang/String;  #35 = Utf8 (Ljava/lang/String;)V  {  public com.test.test();  descriptor: ()V  flags: ACC\_PUBLIC  Code:  stack=1, locals=1, args\_size=1  0: aload\_0  1: invokespecial #1 // Method java/lang/Object."<init>":()V  4: return  LineNumberTable:  line 13: 0  public void testLettureRedis();  descriptor: ()V  flags: ACC\_PUBLIC  Code:  stack=4, locals=7, args\_size=1  0: ldc #2 // String 123  2: astore\_1  3: ldc #3 // String 4  5: astore\_2  6: new #4 // class java/lang/StringBuilder  9: dup  10: invokespecial #5 // Method java/lang/StringBuilder."<init>":()V  13: aload\_1  14: invokevirtual #6 // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  17: aload\_2  18: invokevirtual #6 // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  21: invokevirtual #7 // Method java/lang/StringBuilder.toString:()Ljava/lang/String;  24: astore\_3  25: ldc #8 // String 1234  27: astore 4  29: new #4 // class java/lang/StringBuilder  32: dup  33: invokespecial #5 // Method java/lang/StringBuilder."<init>":()V  36: aload\_2  37: invokevirtual #6 // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  40: ldc #3 // String 4  42: invokevirtual #6 // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  45: invokevirtual #7 // Method java/lang/StringBuilder.toString:()Ljava/lang/String;  48: astore 5  50: new #4 // class java/lang/StringBuilder  53: dup  54: invokespecial #5 // Method java/lang/StringBuilder."<init>":()V  57: aload\_2  58: invokevirtual #6 // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  61: new #9 // class java/lang/String  64: dup  65: ldc #3 // String 4  67: invokespecial #10 // Method java/lang/String."<init>":(Ljava/lang/String;)V  70: invokevirtual #6 // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;  73: invokevirtual #7 // Method java/lang/StringBuilder.toString:()Ljava/lang/String;  76: astore 6  78: return  LineNumberTable:  line 16: 0  line 17: 3  line 18: 6  line 19: 25  line 20: 29  line 21: 50  line 23: 78  }  SourceFile: "test.java" |

String str2 = str + str1; 语句对应红色部分，虚拟机优化会先创建一个StringBuilder 然后通过append追加字符串，最后通过tostring（）返回结果。

String str4 = str1+"4"; 语句对应绿色部分，同上

String Str5 = str1 + new String("4"); 对应黄色部分，同上

分析：



代码

System.out.println(str == str2);//false

System.out.println(str == str4); //false

System.out.println(str == str5); //false

三个比较均返回false，是由于str