

JAVA字符谜题2:ABC PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022_JAVA_E5_AD_97_E7_AC_A6_c97_138386.htm 这个谜题要问的是一个悦耳的问题，下面的程序将打印什么呢？

```
public class ABC{ public static void main(String[] args){ String letters = "ABC". char[] numbers = { ' 1 ', ' 2 ', ' 3 ' }. System.out.println(letters " easy as " numbers). } }
```

可能大家希望这个程序打印出ABC easy as 123。遗憾的是，它没有。如果你运行它，就会发现它打印的是诸如ABC easy as [C@16f0472之类的东西。为什么这个输出会如此丑陋？尽管char是一个整数类型，但是许多类库都对其进行了特殊处理，因为char数值通常表示的是字符而不是整数。例如，将一个char数值传递给println方法会打印出一个Unicode字符而不是它的数字代码。字符数组受到了相同的特殊处理：println的char[]重载版本会打印出数组所包含的所有字符，而String.valueOf和StringBuffer.append的char[]重载版本的行为也是类似的。然而，字符串连接操作符在这些方法中没有被定义。该操作符被定义为先对它的两个操作数执行字符串转换，然后将产生的两个字符串连接到一起。对包括数组在内的对象引用的字符串转换定义如下[JLS 15.18.1.1]：如果引用为null，它将被转换成字符串"null"。否则，该转换的执行就像是不用任何参数调用该引用对象的toString方法一样；但是如果调用toString方法的结果是null，那么就用字符串"null"来代替。那么，在一个非空char数组上面调用toString方法会产生什么样的行为呢？数组是从Object那里继承的toString方法[JLS 10.7]，规范中描述到：“返回一个字符串

，它包含了该对象所属类的名字，' @ ' 符号，以及表示对象散列码的一个无符号十六进制整数” [Java-API]。有关Class.getName的规范描述到：在char[]类型的类对象上调用该方法的结果为字符串"[C"。将它们连接到一起就形成了在我们的程序中打印出来的那个丑陋的字符串。有两种方法可以订正这个程序。你可以在调用字符串连接操作之前，显式地将一个数组转换成一个字符串：System.out.println(letters " easy as " String.valueOf(numbers)). 或者，你可以将System.out.println调用分解为两个调用，以利用println的char[]重载版本：System.out.print(letters " easy as "). System.out.println(numbers). 请注意，这些订正只有在你调用了valueOf和println方法正确的重载版本的情况下，才能正常运行。换句话说，它们严格依赖于数组引用的编译期类型。下面的程序说明了这种依赖性。看起来它像是所描述的第二种订正方式的具体实现，但是它产生的输出却与最初的程序所产生的输出一样丑陋，因为它调用的是println的Object重载版本，而不是char[]重载版本。 class ABC2{ public static void main(String[] args){ String letters = "ABC". Object numbers = new char[]{ ' 1 ' , ' 2 ' , ' 3 ' }. System.out.print(letters " easy as "). System.out.println(numbers). } } 总之，char数组不是字符串。要想将一个char数组转换成一个字符串，就要调用String.valueOf(char[])方法。某些类库中的方法提供了对char数组的类似字符串的支持，通常是提供一个Object版本的重载方法和一个char[]版本的重载方法，而之后后者才能产生我们想要的行为。对语言设计者的教训是：char[]类型可能应该覆写toString方法，使其返回数组中包含的字符。更一般地讲

，数组类型可能都应该覆写toString方法，使其返回数组内容的一个字符串表示。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