**java源码解析**

1. String
2. 不变性

不可变指的是类值一旦被初始化，就不能被改变。如果被修改，将会是新的类。

|  |
| --- |
| // 该final保证了String类不会被继承，所有对String的操作方法，都不会被重写  public final class String implements java.io.Serializable, Comparable<String>, CharSequence {  // 该类中未有公开对value操作的方法，该final保证了value值的不变性  private final char value[];  } |

**使用场景：**HashMap的key建议使用不可变类，如：String。

1. 字符串乱码

当对String进行字节转化时，getBytes()和new String都使用UTF-8。

|  |
| --- |
| **public static void** testGetBytes() {  String str = **"你好"**;  **byte**[] bytes = str.getBytes(StandardCharsets.***UTF\_8***);  String newStr = **new** String(bytes, StandardCharsets.***UTF\_8***);  System.***out***.println(**"还原后的str = "** + newStr); } |

1. substring

public String substring(int beginIndex, int endIndex)

public String substring(int beginIndex)：开始位置，结束位置为文本末尾

substring()底层调用的是字符数组范围截取的方法：Arrays.copyOfRange(字符数组，开始位置，结束位置)

1. replace替换所有匹配字符串

当我们想要删除某些字符时，可以使用replace(str, “”)

1. 拆分和合并

拆分split方法：split(String regex, int limit)

第二个参数来限制我们需要拆分成几个元素，如果limit比实际能拆分的个数小，按照limit的个数进行拆分，数组的最后一项将包含所有超出最后匹配的定界符的输入。

如果limit为0或者负数，则会进行尽可能多的次数。同时，如果limit为0，则会将结尾空字符丢弃【如果不带limit，默认为0】。

|  |
| --- |
| **public static void** testSplit() {  String s = **"boo:and:foo"**;  s.split(**"o"**); *// ["b", "", ":and:f"]* s.split(**"o"**, 2); *// ["b", "o:and:foo"]* s.split(**"o"**, 8); *// ["b", "", ":and:f", "", ""]* s.split(**"o"**, -2); *// ["b", "", ":and:f", "", ""]* } |

limit的拆分结果空值是拆分不掉的，仍然成为结果数组的一员。