# 执行速度

StringBuilder>StringBuffer>String

String为字符串常量，而StringBuilder，StringBuffer是字符串变量。即String对象一旦创建之后就不能更改。

例如String str = “abc”;

str = str+”de”;

第一次创建时，是abc，第二行执行的操作是先获取str的值，然后执行拼接，之后将新的值付给str。而不是直接将str的值更改。

# 线程安全

StringBuilder是非线程安全的，可以在单线程的情况下使用。StringBuffer是线程安全的，在多线程的情况下使用。

# ==和equals

String a =”abc”; String b =”abc”;

String c = new String(“abc”);String d = new String(“abc”);

他们四个的hashcode是一样的.

==比较的是地址.

a==b返回的是true因为a和b都是存放在常量池中,这种方式声明的字符串只要是abc他们的地址都是相同的.

c==d返回的是false,这种方式声明的对象是放在堆中的,每次new都会创建一个新的对象,并且地址不同.

String中对equals方法进行了重写:

|  |
| --- |
| if (this == anObject) {  return true;  }  if (anObject instanceof String) {  String anotherString = (String)anObject;  int n = value.length;  if (n == anotherString.value.length) {  char v1[] = value;  char v2[] = anotherString.value;  int i = 0;  while (n-- != 0) {  if (v1[i] != v2[i])  return false;  i++;  }  return true;  }  }  return false; |

如果地址相同直接返回true,由于ab的地址是相同的,所以无论是==还是equals都是返回的true.

cd地址虽说是不同,但是equals方法中如果地址不同,继续判断,如果类型是String,就判断他们的值,只要值相同就返回true,由于cd的值是相同的,所以返回的是true.

StringBuilder和StringBuffer都没重写equals方法,所以调用的都是Object中的equals方法,这个方法都是直接用==比较的.

# New String 和 String的区别

