|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | String | StringBuffer | StringBuilder | |  |
| 可变类 | X | √ | √ |  |  |
| 可以封装字符串 | √ | √ | √ |  |  |
| 赋值来初始化 | √ | X |  |  |  |
| 数据连接 | + | Append() | Append() |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**区别一**:String类是不可变的字符串，它的底层是一个用final修饰的字符数组,不适合于频繁修改的字符串操作上，往往使用StringBuffer类方便用户进行内容修改，如String.htm第84行,

public final class String implements Serializable, Comparable, CharSequence

**区别二**：在String类中没有append(),使用“+”作为数据的连接操作，而在StringBuffer类中使用append()方法（方法定义：public StringBuffer append(数据类型 变量)）进行数据连接。

Stringbuffer第329行

public synchronized StringBuffer append(StringBuffer stringBuffer)

{

if (stringBuffer == null)

return append("null");

synchronized (stringBuffer)

{

int len = stringBuffer.count;

ensureCapacity\_unsynchronized(count + len);

VMSystem.arraycopy(stringBuffer.value, 0, value, count, len);

count += len;

}

return this;

}

区别三 两个类的定义结构不同

**String类**

public final class String implements Serializable, Comparable, CharSequence

**StringBuffer类**

public final class StringBuffer implements Serializable, CharSequence

**区别四**：String类和StringBuffer类的转换。

**1.将String转换为StringBuffer类对象**

方式一：利用StringBuffer类的构造方法（public StringBuffer(String str)）。接收String类对象，可以将String类对象实例化为StringBuffer类对象。

方式二：利用StringBuffer类中的append()方法（public StringBuffer append(String str)）。利用append()方法像StringBuffer类中增加了一个String类对象，这样就相当于将String类对象转换为StringBuffer类对象。

**2.将StringBuffer转换为String类对象**

方式一：利用toString()方法可以将StringBuffer转换为String类对象

public String toString()

{

// The string will set this.shared = true.

return new String(this);

}

实际上所有的类都会继承Object类的toString方法，所以所有的类对象都可以转换为String类对象。

方法二：利用String类的构造方法（public String(StringBuffer buffer)）实现StringBuffer类与String类的转换。

相同：String类与StringBuffer类中定义的常用方法，两个类的大部分方法可以互补.

public StringBuffer append(数据类型 变量) ——普通方法——用于数据追加操作

public StringBuffer reverse() ——普通方法——用于字符串反转操作

public StringBuffer insert(int offset,数据类型 变量)——普通方法——在指定位置追加内容

发现以上方法都返回StringBuffer类对象，所以可以使用代码链的方式一直调用StringBuffer类的方法，例如：对象.append().insert()....。