#### String

不可变类，即一旦一个String对象被创建以后，包含在这个对象中的字符序列是不可改变的，直至这个对象被销毁。

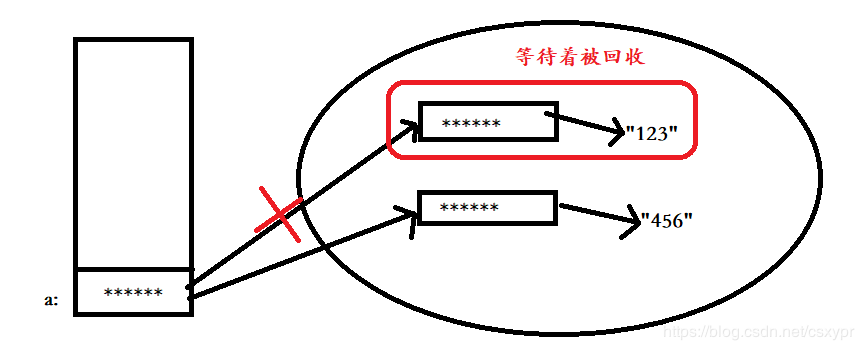
String a = "123";

a = "456";

// 打印出来的a为456

System.out.println(a)

为了明白String类不可变在哪里体现出来，看一张上述a对象的内存存储空间图：



可以看出，再次给a赋值时，并不是对原来堆中实例对象进行重新赋值，而是生成一个新的实例对象，并且指向“456”这个字符串，a则指向最新生成的实例对象，之前的实例对象仍然存在，如果没有被再次引用，则会被垃圾回收。

#### StringBuffer

StringBuffer对象代表一个字符序列可变的字符串，当一个StringBuffer

对象被创建后，可以通过StringBuffer提供的append()、insert()、reserve()、setCharAt()、setLength()等方法可以改变这个字符串对象的字符序列。一旦通过StringBuffer生成了最终想要的字符串，就可以调用它的toString()方法将其转换为一个String对象。

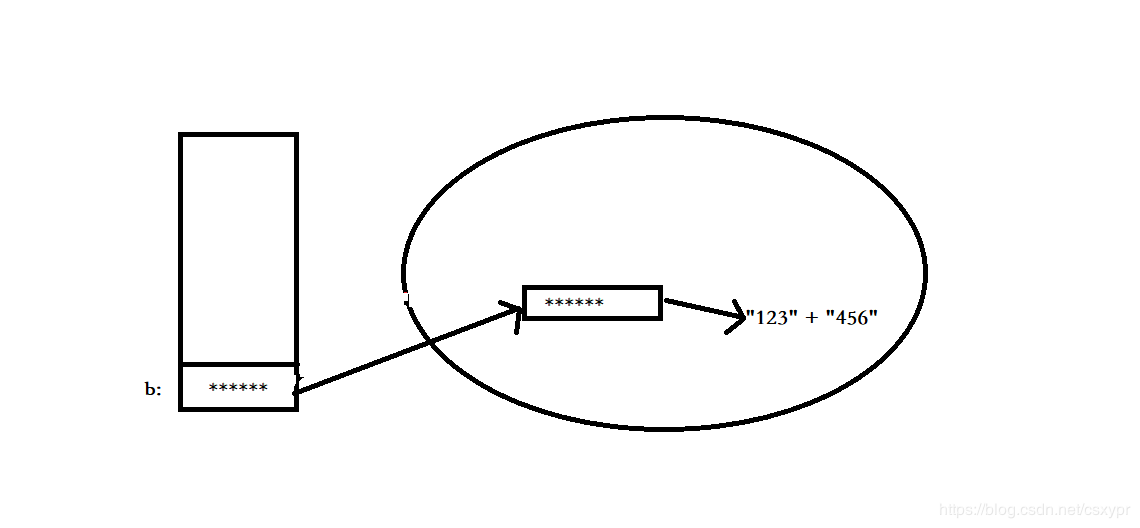
StringBuffer b = new StringBuffer("123");

b.append("456");

// b打印结果为：123456

System.out.println(b);

再看下b对象的内存空间图：



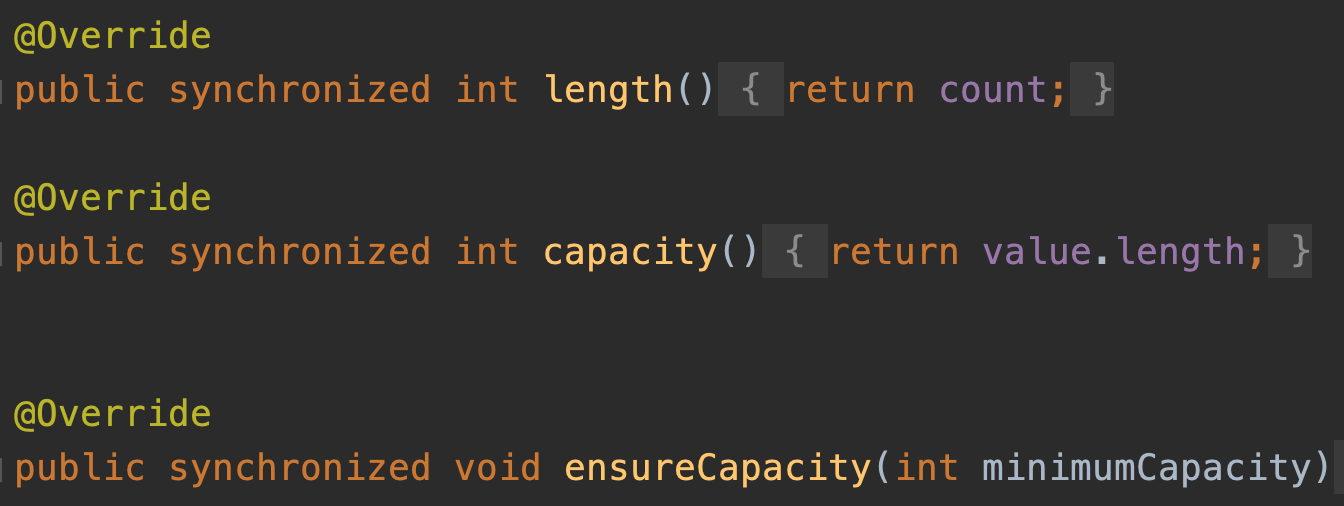
所以说StringBuffer对象是一个字符序列可变的字符串，它没有重新生成一个对象，而且在原来的对象中可以连接新的字符串。

#### StringBuilder

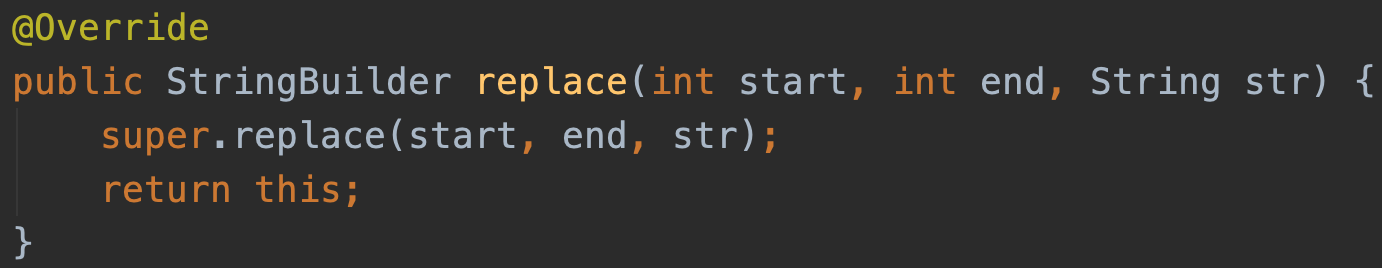
StringBuilder类也代表可变字符串对象。实际上，StringBuilder和StringBuffer基本相似，两个类的构造和方法也基本相同。不同的是：StringBuffer是线程安全的，而StringBuilder则没有实现线程安全功能，所以性能略高。

#### StringBuffer是如何实现线程安全的呢？

StringBuffer类中实现的方法：



StringBuilder类中实现的方法：



StringBuffer类中的方法都添加了synchronized关键字，也就是给这个方法添加个锁，用来保证线程安全。

#### Java9的改进

Java9改进了字符串（包括String、StringBuffer、StringBuilder）的实现。Java9以前字符串采用char[]数组来保存字符，因此字符串的每个字符占2字节；而Java9的字符串采用byte[]数组再加一个encoding-flag字段来保存字符，因此字符串的每个字符只占1字节。所以Java9的字符串更加节省空间，字符串的功能方法也没有受到影响。