StringBuffer中append方法的底层实现：

// StringBuffer中append()不用创建对象可以连接字符串，想知道是如何完成的。

添加字符串的过程

-->先检查内部char[]数组是否需要扩容

-->如需要扩容则进行扩容，然后将原来元数据copy到新数组中。

-->再将新添加的元数据加入到新char[]数组中

public AbstractStringBuilder append(String str) {

if (str == null)

return appendNull();

int len = str.length();

ensureCapacityInternal(count + len); //扩大数组容量，并复制数组

str.getChars(0, len, value, count); //

count += len;

return this;

}

private void ensureCapacityInternal(int minimumCapacity) {

// overflow-conscious code

if (minimumCapacity - value.length > 0) { //若小于等于零，说明str为空字符串，不做修改

value = Arrays.copyOf(value,

newCapacity(minimumCapacity));

}

}

public void getChars(int srcBegin, int srcEnd, char dst[], int dstBegin) {

//srcBegin 为起始位置0；srcEnd:src的长度 dst[]为复制后的数组value ；  
//dstBegin 原有value的长度，也就是原有count接下来的位置  ;

if (srcBegin < 0) {

throw new StringIndexOutOfBoundsException(srcBegin);

}

if (srcEnd > value.length) {

throw new StringIndexOutOfBoundsException(srcEnd);

}

if (srcBegin > srcEnd) {

throw new StringIndexOutOfBoundsException(srcEnd - srcBegin);

}

System.arraycopy(value, srcBegin, dst, dstBegin, srcEnd - srcBegin);

}

public static native void arraycopy(Object src, int srcPos,

Object dest, int destPos,

int length); //调用其他语言编写的代码和代码库，完成复制