**1.String、StringBuffer、StringBuilder三者的異同？**

String:不可變的字符序列；底層使用char[]存儲

StringBuffer:可變的字符序列；線程安全的，效率低；底層使用char[]存儲

StringBuilder:可變的字符序列；jdk5.0新增的，線程不安全的，效率高；底層使用char[]存儲

**2. StringBuffer與StringBuilder的內存解析**

以StringBuffer為例：

String str = new String();//char[] value = new char[0];

String str1 = new String("abc");//char[] value = new char[]{'a','b','c'};

StringBuffer sb1 = new StringBuffer();//char[] value = new char[16];

底層創建了一個長度是16的數組。

System.out.println(sb1.length());//0

sb1.append('a');//value[0] = 'a';

sb1.append('b');//value[1] = 'b';

StringBuffer sb2 = new StringBuffer("abc");

//char[] value = new char["abc".length() + 16];

//問題1. System.out.println(sb2.length());//3

//問題2. 擴容問題:如果要添加的數據底層數組盛不下了，那就需要擴容底層的數組。

默認情況下，擴容為原來容量的2倍 + 2，同時將原有數組中的元素複製到新的數組中。指導意義：開發中建議大家使用：StringBuffer(int capacity) 或

StringBuilder(int capacity)

3.對比String、StringBuffer、StringBuilder三者的執行效率

從高到低排列：StringBuilder > StringBuffer > String

4. StringBuffer、StringBuilder中的常用方法

總結：

增：append(xxx)

刪：delete(int start,int end)

改：setCharAt(int n ,char ch):修改一個字符 /  
 replace(int start, int end, String str):修改一段字符串

查：charAt(int n )

插：insert(int offset, xxx)

長度：length();

遍歷(少用)：for() + charAt() / toString():都直接print出來