

一，string

1，string类：构造字符串对象（常量对象：字符串常量对象是用双引号括起来的字符序列）

①字符串的字符使用Unicode字符编码，一个字符占俩个字节

②构造方法：`string s1 = new string ()`；`string s2 = new string (string s)`；

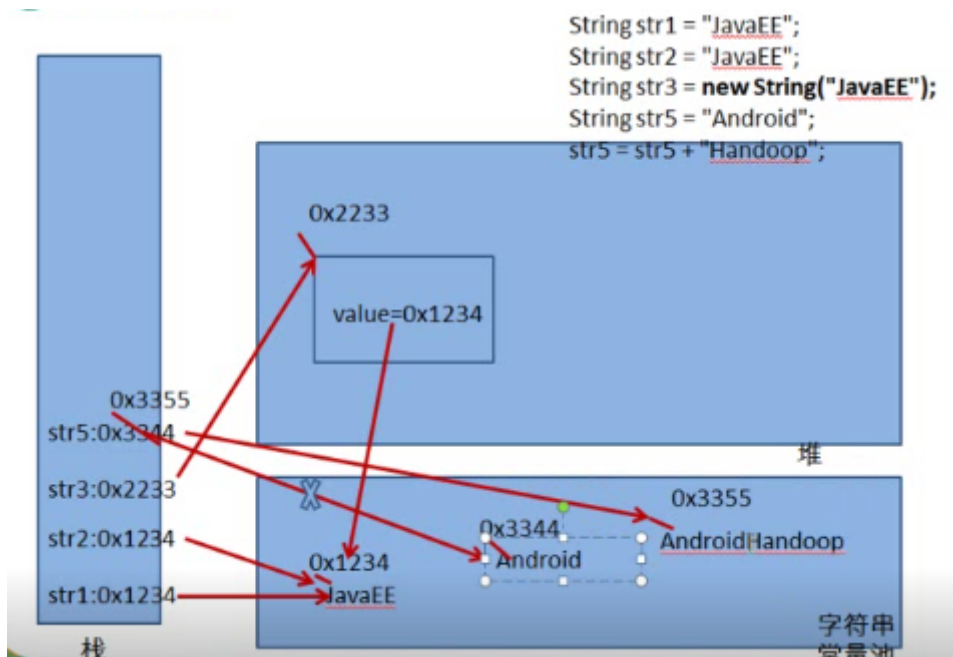
`string s3 = new string (char[]a)`

③string是一个final类，是不可以继承的，代表不可变的字符序列，底层使用char型数组进行存储

④字符串是不可变的，一个字符串对象一旦被配置，其内容是不可以改变的

“Javaee” + “Android” 和 “JavaeeAndroid” 是一样的，用==比较的结果是一个true

内存结构：



①：字符串反转：

@Test

```
public static String test2(String string ,int start,int end){  
    char[] c = string.toCharArray();  
    String string2 = test1(c, start, end);  
    return string2;  
}
```

@Test

```
public static String test1(char[]c,int start,int end){  
    for(int i = start,j = end;i < j;i++,j--){
```

```

        char temp = c[i];
                c[i] = c[j];
                c[j]= temp;
    }
    //将字符数组c--->字符串
    return new String(c);
}

```

②判断一个字符串在另一个字符串出现的次数

```

@Test
public static int test3(String s,String string1){
    int count = 0;
    int len = 0;
    while((len =string1.indexOf(s)) != -1){
        count++;
        string1 = string1.substring(len+s.length());
    }
    return count;
}

```

③获取一个字符串的最大相同子串

```

public static java.util.List<String> test4(String s1,String s2){
    String maxstrString = (s1.length()>s2.length())?s1:s2;
    String minString = (s1.length()<s2.length())?s1:s2;
    int len = minString.length();
    java.util.List<String>list = new ArrayList<String>();
    for(int i = 0;i<len;i++){
        for(int x = 0,y = len-i;y <= len;x++,y++){
            String string = minString.substring(x, y);
            if (maxstrString.contains(string)) {
                list.add(string);
            }
        }
    }
    if (list.size() != 0) {
        return list;
    }
}

```

```
        return null;
    }
}
```

④对字符串进行自然顺序排序

```
public static String test5(String string) {
    char [] c =string.toCharArray();
    Arrays.sort(c);
    return new String(c);
}
```

常用方法:

//将字符数组c--->字符串

return new String(c);

String:代表不可变的字符序列。底层使用char[]存放。
String 是final的。

```
public int length()
public char charAt(int index)
public boolean equals(Object anObject)
public int compareTo(String anotherString)
public int indexOf(String s)
public int indexOf(String s ,int startpoint)
public int lastIndexOf(String s)
public int lastIndexOf(String s ,int startpoint)
public boolean startsWith(String prefix)
public boolean endsWith(String suffix)
public boolean regionMatches(int firstStart,String other,int otherStart
```

string是一个可变的字符串序列，一旦被赋值，就不能被修改

①length ()；（方法）返回字符串的长度

②charAt (int index)；返回指定索引位置index的字符，index从0开始

③equals (object obj)；比较俩个字符串是否相等，相等返回true，否则返回false

④compareTo (string anotherstring)；俩个字符串依次进行比较，返回一个整型数据

⑤indexOf (string s)；返回s字符串在当前字符串中首次出现的位置，若没有，返回-1

⑥lastIndexOf (String s)；返回s字符串最后依次出现的位置，若不存在，返回-1

⑦startsWith (string prefix)；判断当前字符串是否是以prefix开始

⑧endsWith (string suffix)；判断当前字符串是否是以suffix结尾

⑨regionMatches (int first, string other, int onatherstart, int length)；判断当前字符串从first开

始的子串与other从onatherstart开始length长的字符串是否equals，equals则返回true，否则返回

false

- ①substring (int startpoint) ; 返回startpoint开始到结尾的位置的所有字符串
- ②substring (int start, int end) ; 返回从start到end结束的左闭右开的子串。start可以从0开始
- ③replace (char oldchar, char newchar) ; 将当前字符串中所有的oldchar都变为newchar返回一个
新的字符串
- ④replaceAll (string old, string new) 将当前字符串中的所有的old字符串都变为new字符串，返回一
个新的字符串
- ⑤trim () ; 去除当前字符串中首尾出现的空格，若有多个则去除多个
- ⑥concat (string str) ; 连接当前字符串与str
- ⑦[]split (string regex) ; 按照regex将当前字符串进行拆分，拆分为多个字符串，整体返回值为

string[]

```
//2. 将一个字符串进行反转。将字符串中指定部分进行反转。比如将“abcdefg”反转为“abfedcg”
public static String reverseString(String str,int start,int end){
    char[] c = str.toCharArray();//字符串--->
    return reverseArray(c,start,end);
}
public static String reverseArray(char[] c,int start,int end){
    for(int i = start,j = end;i < j;i++,j--){
        char temp = c[i];
        c[i] = c[j];
        c[j] = temp;
    }
    //字符数组--->字符串
    return new String(c);
}
```

2，字符串与基本数据类型、包装类之间的转换

- ①字符串----->基本数据类型、包装类：调用相应的包装类的parseXxx (string str) ;
- ②基本数据类型、包装类----->字符串：调用字符串重载的valueOf () 方法

3，字符串与字节数组之间

- ①字符串----->字节数组：调用字符串的getBytes () 方法；
- ②字节数组----->字符串：调用字符串的构造器new string (字节数组) ;

4，字符串与字符数组之间

- ①字符串----->字符数组：调用字符串的toCharArray () 方法；
- ②字符数组----->字符串：调用字符串的构造器new string (字符数组) ;

* StringBuffer是一个可变的序列

* 总结:

* ①添加: append ()

* ②插入: insert (int index, string s1) 在指定位置插入s1

* ③反转: reverse () 反转stringbuffer

* ④查: charAt (int n)

* ⑤修改: setCharAt (int index, char ch) 用ch修改index位置的字符

* ⑥长度: length ()

* ⑦删除: delete (int i, int j); 删除i到j处的元素

*

* StringBuilder是一个可变的字符序列, jdk5.0之后增加的

* 线程不安全, 效率高于stringbuffer

* 效率从高到低: stringbuilder>stringbuffer>string