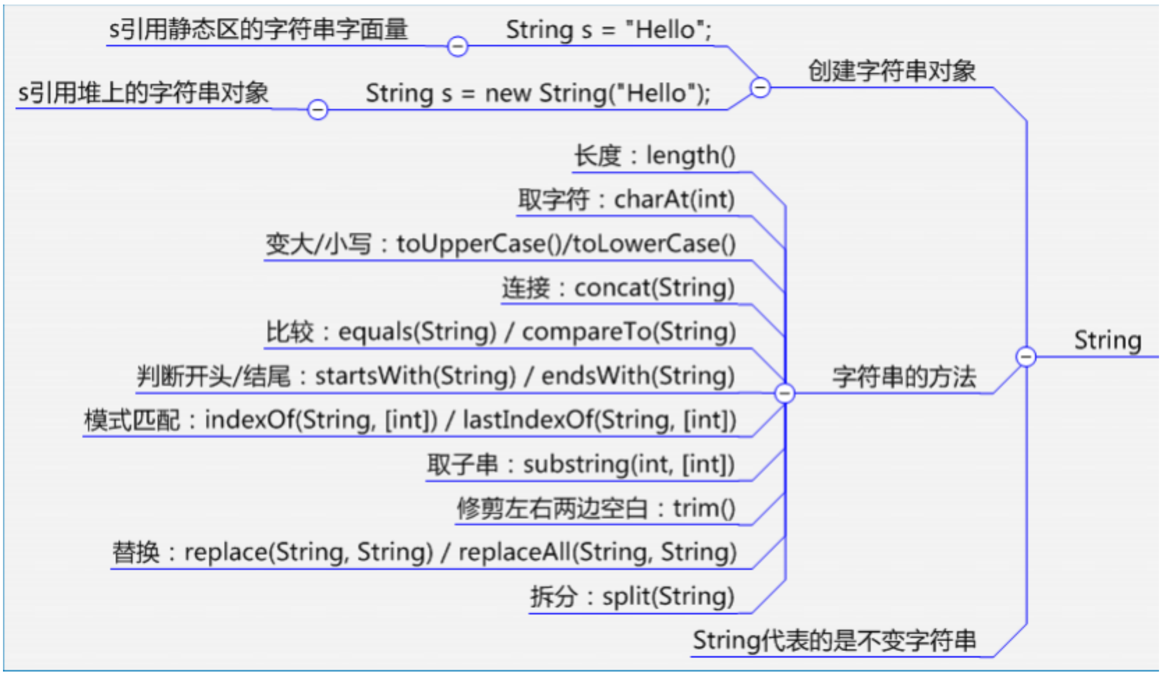
# String类和StringBuffer类

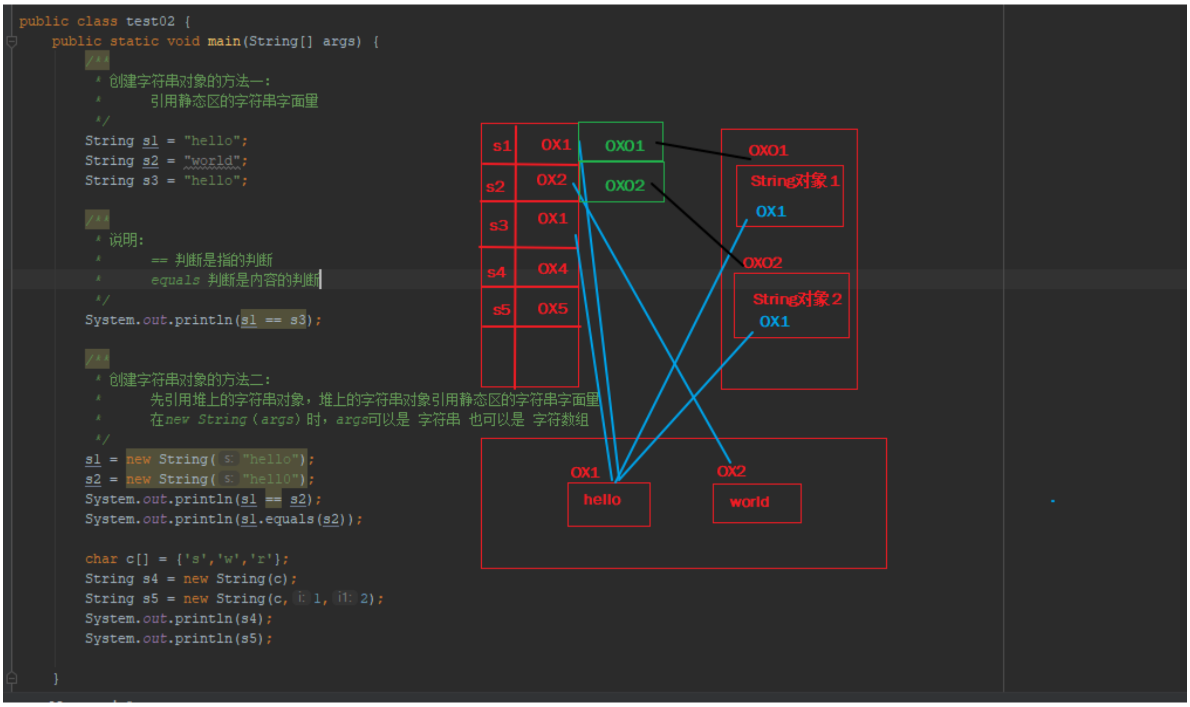
## String类

- java.lang.String类代表不可变的字符序列
- 字符串是一个特殊的对象
- 字符串一旦初始化就不可以被改变


s引用静态区的字符串字面量
String s = "Hello";
s引用堆上的字符串对象
String s = new String("Hello");
创建字符串对象
长度：length()
取字符：charAt(int)
变大/小写：toUpperCase()/toLowerCase()
连接：concat(String)
比较：equals(String) / compareTo(String)
判断开头/结尾：startsWith(String) / endsWith(String)
模式匹配：indexOf(String, [int]) / lastIndexOf(String, [int])
取子串：substring(int, [int])
修剪左右两边空白：trim()
替换：replace(String, String) / replaceAll(String, String)
拆分：split(String)
字符串的方法
String
String代表的是不变字符串

实例一：

```
package 面向对象编程.string类;

public class test02 {
    public static void main(String[] args) {
        /**
         * 创建字符串对象的方法一：
         *     引用静态区的字符串字面量
         */
        String s1 = "hello";
        String s2 = "world";
        String s3 = "hello";

        /**
         * 说明：
         *     == 判断是指的判断
         *     equals 判断是内容的判断
         */
        System.out.println(s1 == s3);

        /**
         * 创建字符串对象的方法二：
         *     先引用堆上的字符串对象，堆上的字符串对象引用静态区的字符串字面量
         *     在new String（args）时，args可以是 字符串 也可以是 字符数组
         */
        s1 = new String("hello");
        s2 = new String("hello");
        System.out.println(s1 == s2);
        System.out.println(s1.equals(s2));

        char c[] = {'s','w','r'};
        String s4 = new String(c);
        String s5 = new String(c,1,2);
        System.out.println(s4);
        System.out.println(s5);

    }
}



结果:
    true
    false
    true
    swr
    wr
```



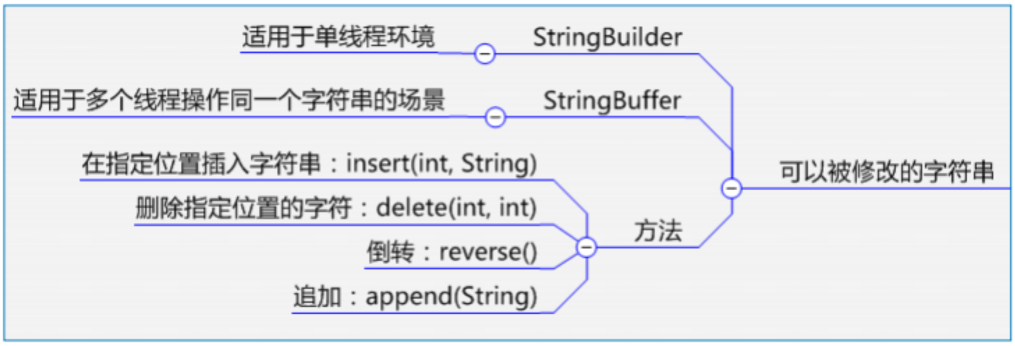
```
public class test02 {
    public static void main(String[] args) {
        /**
         * 创建字符串对象的方法一:
         *      引用静态区的字符串字面里
         */
        String s1 = "hello";
        String s2 = "world";
        String s3 = "hello";

        /**
         * 说明:
         *      == 判断是指的判断
         *      equals 判断是内容的判断
         */
        System.out.println(s1 == s3);

        /**
         * 创建字符串对象的方法二:
         *      先引用堆上的字符串对象，堆上的字符串对象引用静态区的字符串字面里
         *      在new String(args)时，args可以是 字符串 也可以是 字符数组
         */
        s1 = new String( s: "hello");
        s2 = new String( s: "hell0");
        System.out.println(s1 == s2);
        System.out.println(s1.equals(s2));

        char c[] = {'s','w','r'};
        String s4 = new String(c);
        String s5 = new String(c, i: 1, i1: 2);
        System.out.println(s4);
        System.out.println(s5);

    }
```

实例二

```
package 面向对象编程.string类;

public class test03 {
    public static void main(String[] args) {
        String a = "Programming";
        String b = new String("Programming");

        String c = "Program" + "ming";

        System.out.println(a == b);
        System.out.println(a == c);
        System.out.println(a.equals(b));
        System.out.println(a.equals(c));
        System.out.println("Programming".equals(b));
    }
}
```

## StringBuffer类

- java.lang.StringBuffer可以对字符串内容进行增删
- StringBuffer没有重写equals方法
- StringBuffer是可变长度的


适用于单线程环境
StringBuilder
适用于多个线程操作同一个字符串的场景
StringBuffer
在指定位置插入字符串：insert(int, String)
删除指定位置的字符：delete(int, int)
倒转：reverse()
追加：append(String)
方法
可以被修改的字符串

```
package 面向对象编程.StringBuffer;

public class test01 {
    public static void main(String[] args) {
        String s = "Microsoft";
        char[] a = {'a','b','e','o'};

        StringBuffer s1 = new StringBuffer(s);
        s1.append("/").append("IBM").append("/").append("Sun");
        System.out.println(s1);

        StringBuffer s2 = new StringBuffer("数字：");
        for (int i=0;i<10;i++){
            s2.append(i);
        }

        System.out.println(s2);
    }
}
```

**注意：字符串常量相加的字符串变量相加的区别**

例子：

```
package 面向对象编程.StringBuffer;

public class test01 {
    public static void main(String[] args) {
        String s = "Microsoft";
        char[] a = {'a','b','e','o'};
        StringBuffer s1 = new StringBuffer(s);
        s1.append("/").append("IBM").append("/").append("Sun");
        System.out.println(s1);

        StringBuffer s2 = new StringBuffer("数字：");
        for (int i=0;i<10;i++){
            s2.append(i);
        }
        System.out.println(s2);

        String base = "base";

        /**
         * 区别以下代码：
         *      1、字符串常量相加
         *              先在常量池中相加，如果常量池中又就直接返回，没有就创建
         *      2、字符串变量相加
         *              先开辟空间，在拼接。在拼接时，需要创建一个StringBuffer对象，之后将StringBuffer对象转化为String对象。在
         *
         */
        String c = "base" + "java";
        String d = "base" + "java";
        System.out.println(c == d);

        c = base + "java";
        d = base + "java";
        System.out.println(c == d);


        System.out.println(c.equals(d));
        System.out.println((base + "java") == (base + "java"));
    }

}
```