

## | 今天学习目标

- \* 能够掌握StringBuffer,StringBuilder的使用
- \* 能够掌握String,StringBuffer,StringBuilder的区别
- \* 能够掌握8种基本数据类型的包装类型
  - \* 装箱
  - \* 拆箱
  - \* 自动拆装箱
    - \* 使用常量池
- \* 做茶城的项目

---

## \* 回顾

### \* String

#### \* 三种构建字符串方式：

- \* String str="xiaohei"；字符串常量池

- \* String str1=new String("xiaohei");

- \* char[] chs=new char{'a','b','c'};

- String str2=new String(chs);

#### \* String的本质：char数组+对char数组操作

- private final char value[];

#### \* String 的不可变性

- private final char value[];

#### \* Arrays.copy();

- int [] arr1;

- int [] arr2;

- \* Arrays.copy(arr1,arr2);

- \* arr2=arr1;

## \* String 的常见方法

\* concat-->( + ),charAt(index),compareTo,compareToIgnoreCase

\* equals ( (Object),==(对比是对象地址), 但是String重写它, 让变成对比是内容 )

\* split , trim , indexOf(),lastIndexOf(),contains(),replace(),format(),substring(0,3) ---> [0,3)

\* toCharArray(),toLowerCase(),toUpperCase(),startsWith(),endsWith(),valueOf(),....

## \* 小案例

### \* 字符串常量池

\* 能够掌握StringBuffer,StringBuilder的使用

\* Java语言中的有一个StringBuffer类, 称为字符串缓冲区;

\* 所表示的也是一个字符序列;

\* 这个类型必须用new创建对象, 和String相反, 它是可变的类;

```
4 public static void main(String[] args) {  
5     StringBuffer sb1=new StringBuffer("Hello");  
6     StringBuffer sb2=new StringBuffer("World");  
7     sb1.append(sb2);  
8     System.out.println(sb1.toString());  
9 }  
10 }  
11
```

Problems @ Javadoc Declaration Search Console

<terminated> Test12 (1) [Java Application] C:\Program Files\Java\jre1.8.0\_181\bin\javaw.exe (2018年8月8日 下午8:00)  
HelloWorld

```
1 public static void main(String[] args) {
```

```

2      StringBuffer sb1=new StringBuffer("Hello");
3      StringBuffer sb2=new StringBuffer("World");
4      sb1.append(sb2);
5      System.out.println(sb1.toString());
6  }

```

### \* 业务需求

\* 将一个数字字符串转换成逗号分隔的数字串，即从右边开始每三个数字用逗号分

隔

请输入一串数字: 15988823323123  
15,988,823,323,123

```

5 public class Test13 {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8         System.out.print("请输入一串数字: ");
9         String nums = input.next();
10        StringBuffer str=new StringBuffer(nums);
11        for(int i=str.length()-3;i>0;i=i-3){
12            str.insert(i,',');
13        }
14        System.out.print(str);
15    }
16

```

Problems @ Javadoc Declaration Search Console

<terminated> Test13 (1) [Java Application] C:\Program Files\Java\jre1.8.0\_181\bin\javaw.exe (2018年8月8日 下午8:16)

请输入一串数字: 15988823323123

15,988,823,323,123

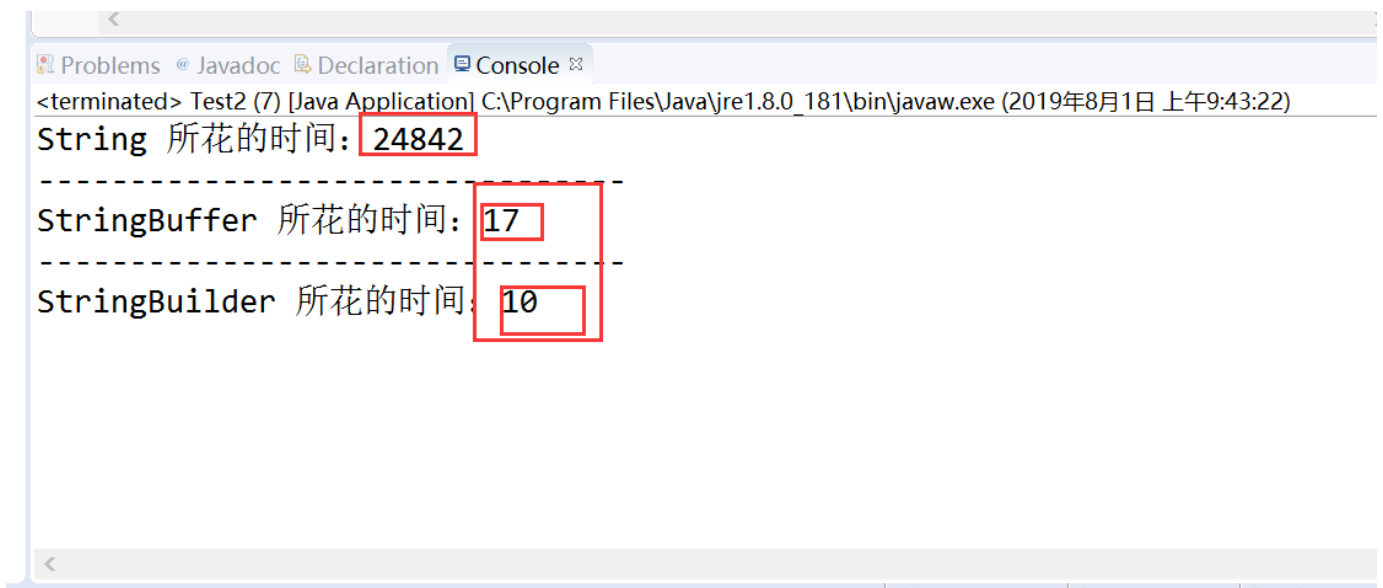
```

1 public static void main(String[] args) {
2     Scanner input = new Scanner(System.in);
3     System.out.print("请输入一串数字: ");
4     String nums = input.next();
5     StringBuffer str=new StringBuffer(nums);
6     for(int i=str.length()-3;i>0;i=i-3){
7         str.insert(i,',');
8     }
9     System.out.print(str);
10 }

```

\* Java语言中的还有一个StringBuilder类，与StringBuffer兼容，但是线程不安全；

\* 能够掌握String,StringBuffer,StringBuilder的区别



```

<terminated> Test2 (7) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2019年8月1日 上午9:43:22)
String 所花的时间: 24842
-----
StringBuffer 所花的时间: 17
-----
StringBuilder 所花的时间: 10

```

```

1 public static void main(String[] args) {
2     long begin=System.currentTimeMillis();
3     String str="xiaohei";
4     for(int i=0;i<100000;i++){
5         str=str+i;
6     }
7     long end=System.currentTimeMillis();
8     System.out.println("String 所花的时间: "+(end-begin));
9     System.out.println("-----");
10 }

```

```

11         begin=System.currentTimeMillis();
12         StringBuffer sb=new StringBuffer();
13         sb.append("xiaohei");
14         for(int i=0;i<100000;i++){
15             sb.append(i+"");
16         }
17         end=System.currentTimeMillis();
18         System.out.println("StringBuffer 所花的时间: "+(end-begin));
19         System.out.println("-----");
20
21         begin=System.currentTimeMillis();
22         StringBuilder sbb=new StringBuilder();
23         sb.append("xiaohei");
24         for(int i=0;i<100000;i++){
25             sbb.append(i+"");
26         }
27         end=System.currentTimeMillis();
28         System.out.println("StringBuilder 所花的时间: "+(end-begin));
29     }

```

### \* 三个类的区别

\* String类是不可变的，对象一旦被创建，就不能被修改；可以使用=直接赋值，此时使用常量池；也可以使用new创建，不使用常量池；

\* StringBuffer是可变的，对象创建后，可以修改；必须使用new关键字；

\* StringBuilder是不同步的，在单线程情况下使用比StringBuffer高效；必须使用new关键字；

### \* 能够掌握8种基本数据类型的包装类型

\* byte ---> Byte(byte+常量+方法)

\* short ---> Short

\* int ---> Integer

\* long ---> Long

\* boolean ---> Boolean

\* char ---> Character

\* float ---> Float

\* double ---> Double

\* 装箱

\* 拆箱

\* 自动拆装箱

\* 使用常量池

\* Java语言中有8个基本数据类型，对应有8个类，这8个类统称包装器类型（Wrapper类）；

\* Byte , Short , Integer , Long , Boolean , Character , Double , Float

\* 使用这8个包装器类，能够把某一种基本数据类型的变量转换成引用类型，从而使用类中的方法，进行更多的操作；

```
public static void main(String[] args) {
```

```
    int i=8;
```

```
    Integer ii=new Integer(i);
```

→ int 转换成Integer

```
    double doubleValue = ii.doubleValue();
```

```
}
```

Integer 转换成double

\* 装箱：基本数据类型转换为包装器类型，称为装箱（boxing）；例如，int型转换为Integer类型；

\* 拆箱：包装器类型转换为基本数据类型，称为拆箱（unboxing）；例如Integer类型转换为int类型；

\* JDK1.5以后，装箱拆箱可以自动进行

```

public class Test15 {
    public static void main(String[] args) {
        //传统的装箱、拆箱
        int a=128;
        Integer ao1=new Integer(a);
        int b1=ao1.intValue();
        //自动装箱
        Integer ii1=a;
        Integer ii2=128;
        //自动拆箱
        int b2=ii1;
        int b3=ii1+ii2;
    }
}

```

```

1 public static void main(String[] args) {
2     //传统的装箱、拆箱
3     int a=128;
4     Integer ao1=new Integer(a);
5     int b1=ao1.intValue();
6     //自动装箱
7     Integer ii1=a;
8     Integer ii2=128;
9     //自动拆箱
10    int b2=ii1;
11    int b3=ii1+ii2;
12 }

```

- \* 在自动装箱拆箱过程中，Java使用到了常量池
- \* 在自动装箱拆箱过程中，只有数值是byte范围内的时候，才使用到常量池，否则都是分配新的内存空间

```

4 public static void main(String[] args) {
5     Integer i1=8;
6     Integer i2=8;
7     Integer io1=new Integer(8);
8     Integer io2=new Integer(8);
9     System.out.println("i1==i2:"+(i1==i2));
10    System.out.println("io1==io2:"+(io1==io2));
11
12 }
13
14 }

```

Problems @ Javadoc Declaration Search Console

<terminated> Test16 [Java Application] C:\Program Files\Java\jre1.8.0\_181\bin\javaw.exe (2018年8月8日 下午9:08:11)

```

i1==i2:true
io1==io2:false

```

```

4 public static void main(String[] args) {
5     Integer i1=128;
6     Integer i2=128;
7     Integer io1=new Integer(128);
8     Integer io2=new Integer(128);
9     System.out.println("i1==i2:"+(i1==i2));
10    System.out.println("io1==io2:"+(io1==io2));
11
12 }
13
14 }

```

Problems @ Javadoc Declaration Search Console

<terminated> Test16 [Java Application] C:\Program Files\Java\jre1.8.0\_181\bin\javaw.exe (2018年8月8日 下午9:08:3)

```

i1==i2:false
io1==io2:false

```