**字符串使用示例**

输出：

s.length:10

----------------------

charAt:c

----------------------

indexOf:3

----------------------

indexOf:0

----------------------

indexOf:3

indexOf:-1

----------------------

substring:kkabcmm

----------------------

substring:kkabc

**【例1】String类常用方法1。**

public class StringTest{

public static void main(String[] args) {

// 定义一个字符串对象

String s = “abckkabcmm”;

// int length():获取字符串的长度。

System.out.println(“s.length:” + s.length());

System.out.println(“----------------------“);

// char charAt(int index):获取指定索引位置的字符

System.out.println(“charAt:” + s.charAt(7));

System.out.println(“----------------------“);

// int indexOf(int ch):返回指定字符在此字符串中第一次出现处的索引。

System.out.println(“indexOf:” + s.indexOf(‘k’));

System.out.println(“----------------------“);

// int indexOf(String str):返回指定字符串在此字符串中第一次出现处的索引。

System.out.println(“indexOf:” + s.indexOf(“abc”));

System.out.println(“----------------------“);

// int indexOf(int ch,int fromIndex):返回指定字符在此字符串中从指定位置后第一次出现处的索引。

System.out.println(“indexOf:” + s.indexOf(‘k’, 0));

System.out.println(“indexOf:” + s.indexOf(‘k’, 5)); // -1

System.out.println(“----------------------“);

// String substring(int start):从指定位置开始截取字符串,默认到末尾。包含start这个索引

System.out.println(“substring:” + s.substring(3));

System.out.println(“----------------------“);

// String substring(int start,int end):从指定位置开始到指定位置结束截取字符串。包括start索引但是不包end索引

System.out.println(“substring:” + s.substring(3, 8));

}

**}**

**【例2】String类常用方法2**

public class StringTest{

74

97

118

97

83

69

----------------

J

a

v

a

S

E

----------------

100

----------------

toLowerCase:javase

toUpperCase:JAVASE

----------------

s5:JavaSE100

s6:JavaSE100

public static void main(String[] args) {

// 定义一个字符串对象

String s1 = "JavaSE";

// byte[] getBytes():把字符串转换为字节数组。

byte[] bys = s1.getBytes();

for (int x = 0; x < bys.length; x++) {

System.out.println(bys[x]);

}

System.out.println("----------------");

// char[] toCharArray():把字符串转换为字符数组。

char[] chs = s1.toCharArray();

for (int x=0;x<chs.length;x++) {

System.out.println(chs[x]);

}

System.out.println("----------------");

// static String valueOf(int i):把int类型的数据转成字符串。

int i=100;

String s2= String.valueOf(i);

System.out.println(s2);

System.out.println("----------------");

// String toLowerCase():把字符串转成小写。

System.out.println("toLowerCase:" + s1.toLowerCase());

// String toUpperCase():把字符串转成大写。

System.out.println("toUpperCase:" + s1.toUpperCase());

System.out.println("----------------");

// String concat(String str):把字符串拼接。

String s3 = "hello";

String s4 = "world";

//+可以是字符串或者数字及其他基本类型数据，而concat只能针对字符串操作

String s5 = s1 + s2;

String s6 = s1.concat(s2);

System.out.println("s5:" + s5);

System.out.println("s6:" + s6);

}

}

**【例3】StringBuffer类的添加、替换、反转功能**

public class StringBufferTest{

public static void main(String[] args) {

// 创建字符串缓冲区对象

StringBuffer sbf = new StringBuffer();

// 添加数据

sbf.append("hello");

sbf.append("world");

sbf.append("java");

System.out.println("sbf:" + sbf);

// public StringBuffer replace(int start,int end,String str):从start开始到end用str替换

// 需求：把world这个数据替换为"节日快乐"

sbf.replace(5, 10, "节日快乐");

System.out.println("sbf:" + sbf);

sbf:helloworldjava

sbf:hello节日快乐java

sbf2:国中爱我

sbf2\_reverse:我爱中国

// 创建字符串缓冲区对象

StringBuffer sbf2 = new StringBuffer();

// 添加数据

sbf2.append("国中爱我");

System.out.println("sbf2:" + sbf2);

// public StringBuffer reverse()，反转

sbf2.reverse();

System.out.println("sbf2\_reverse:" + sbf2);

}

}

**【例4】包装类使用举例**

public class IntegerTest{

public static void main(String[] args) {

//将123转换为八进制，返回字符串

String str=Integer.toString(123,8);

System.out.println(str);

173

123

123

123

123

123

//将“173”（八进制）转换为十进制

int x=Integer.parseInt("173",8);

System.out.println(x);

//将字符串转为int型

int i = Integer.parseInt("123");

int j = Integer.valueOf("123").intValue();

System.out.println(i);

System.out.println(j);

//将int转换为字符串类型

String s1 = String.valueOf(123);

String s2 = Integer.toString(123);

System.out.println(s1);

System.out.println(s2);

}

}