

Scanner

2018年7月1日 14:58

```
package cn.itcast_01;

import java.util.Scanner;

/*
 * Scanner:用于接收键盘录入的数据
 * JDK5后出现
 *
 * System类下有一个静态字段:
 *         public static final InputStream in;标准的输入流, 对应键盘录入
 *         InputStream is = System.in;
 *
 * class Demo{
 *         public static final int x = 10;
 *         public static final Student s = new Student();
 * }
 * int y = Demo.x
 * Student s = Demo.s;
 */
public class ScannerDemo {
    public static void main(String[] args) {
        // 创建对象
        Scanner sc = new Scanner(System.in);
        // Scanner中方法可以把键盘录入的字符串自动转换成int double等类型
        int x = sc.nextInt();
        System.out.println("x=" + x);
    }
}
```

scanner类方法

2018年7月7日 0:36

```
package cn.itcast_02;

import java.util.Scanner;

/*
 * 基本格式：
 * public boolean hasNextXxx():判断是否为某种类型的元素
 * public Xxx nextInt():获取该元素ss
 * 举例：
 * public boolean hasNextInt()
 * public int nextInt()
 * 注意：
 * InputMismatchException 输入与所需不匹配
 */
public class ScannerDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        if (sc.hasNextInt()) {
            int x = sc.nextInt();
            System.out.println("x=" + x);
        } else {
            System.out.println("输入数据有误");
        }
    }
}
```

Scanner类小问题

2018年7月7日 0:54

```
package cn.itcast_03;

import java.util.Scanner;

/*
 * 常用的两个方法：
 *      public int nextInt():获取一个int类型的数
 *      public String nextLine():获取一个String类型的数
 * 问题：
 *      先获取数值再获取字符串会出现问题
 *      因为回车换行符号的问题
 * 解决方法：
 *      1.先获取一个数值后，在创建一个新的键盘录入对象
 *      2.把所有的数据先按字符串获取，再按要求转换
 */

public class ScannerDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // 获取两个int类型的值
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println("a" + a + ",b" + b);

        // 获取两个String类型的值
        String s1 = sc.nextLine();
        String s2 = sc.nextLine();
        System.out.println("s1" + s1 + ",s2" + s2);

        //先获取一个字符串，再获取一个int
        String s3 = sc.nextLine();
        int c = sc.nextInt();
        System.out.println("s3" + s3 + ",c" + c);

        //先获取一个int值，再获取一个字符串
        String s4 = sc.nextLine();
    }
}
```

```
        int d = sc.nextInt();  
        System.out.println("s4" + s4 + ",d" + d);  
    }  
}
```

字符串构造方法

2018年7月9日 9:23

```
package cn.itcast_01;
```

```
/*
 * 字符串：就是由多个字符组成的一串数据，也可以看作是一个字符数组
 * 通过查看API可知：
 *     1.字符串值"abc"可以看成是一个字符串对象
 *     2.字符串是常量，一旦被赋值，就不能被改变
 * 构造方法：
 *     public String ():空构造
 *     public String(byte[] bytes):把字节数组转换成字符串
 *     public String(byte[] bytes,int index,int length):把字节数组一部分转换为字符串
 *     public String(char[] value):把字节数组转换成字符串
 *     public String(char[] value,int index,int length):把字节数组一部分转换为字符串
 *     public String(String original):把字符串常量值转换为字符串
 * 字符串的方法：
 *     public int length(): 返回此字符串的长度
 */
```

```
public class StringDemo {
    public static void main(String[] args) {
        // public String ()
        String s1 = new String();
        System.out.println("s1=" + s1);
        System.out.println("s1.length()=" + s1.length());
        System.out.println("-----");

        // public String(byte[] bytes)
        byte[] bys = { 97, 98, 99, 100, 101 };
        String s2 = new String(bys);
        System.out.println("s2=" + s2);
        System.out.println("s2.length()=" + s2.length());
        System.out.println("-----");

        //public String(byte[] bytes,int index,int length)
        String s3 = new String(bys,1,3);
        System.out.println("s3=" + s3);
        System.out.println("s3.length()=" + s3.length());
        System.out.println("-----");

        //public String(char[] value)
        char[] chs = {'a','b','c','王','若','潇'};
        String s4 = new String(chs);
        System.out.println("s4=" + s4);
        System.out.println("s4.length()=" + s4.length());
        System.out.println("-----");

        //public String(char[] value,int index,int length)
```

```
s1=
s1.length()=0
-----
s2=abcde
s2.length()=5
-----
s3=bcd
s3.length()=3
-----
s4=abc王若潇
s4.length()=6
-----
s5=c王
s5.length()=2
-----
s6=abcdef
s6.length()=6
-----
s7=abcde
s7.length()=5
```

```
String s5 = new String(chs,2,2);
System.out.println("s5=" + s5);
System.out.println("s5.length()=" + s5.length());
System.out.println("-----");
```

```
//public String(String original)
String s6 = new String("abcdef");
System.out.println("s6=" + s6);
System.out.println("s6.length()=" + s6.length());
System.out.println("-----");
```

```
//字符串面值"abcdef"也可以看成是字符串对象
String s7 = "abcde";
System.out.println("s7=" + s7);
System.out.println("s7.length()=" + s7.length());
```

```
}
```

```
}
```

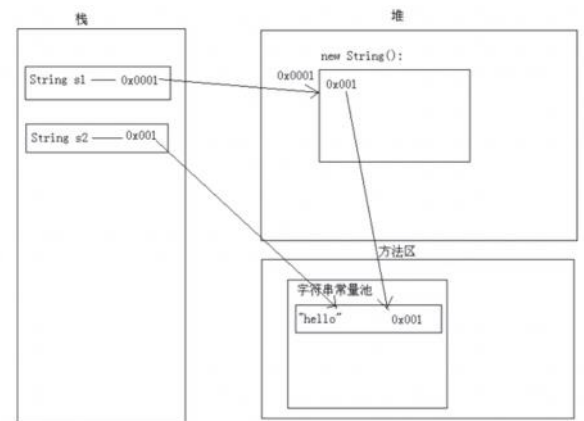
区别

2018年7月9日 10:54

```
package cn.itcast_01;
```

```
public class StringDemo2 {  
    public static void main(String[] args) {  
        String s1 = new String("hello");  
        String s2 = "hello";  
  
        System.out.println(s1 == s2); // false  
        System.out.println(s1.equals(s2)); // true  
    }  
}
```

```
+  
String s1 = new String("hello");  
String s2 = "hello";  
System.out.println(s1 == s2); // false  
System.out.println(s1.equals(s2)); // true
```



区别：前者创建了两个对象，后者创建了一个对象

面试题

2018年7月9日 11:03

```
package cn.itcast_01;

import java.net.StandardSocketOptions;

public class StringDemo3 {
    public static void main(String[] args) {
        String s1 = "hello";
        String s2 = "world";
        String s3 = "helloworld";
        System.out.println(s3 == s1 + s2); // false
        System.out.println(s3.equals(s1 + s2)); // true
        System.out.println(s3 == "hello" + "world"); // true
        System.out.println(s3.equals("hello" + "world")); // true

        //字符串如果是变量相加，先开空间，再拼接
        //字符串如果是常量相加，先拼接，再在常量池中找，如果有就直接返回，否则再
        创建
    }
}
```


string类判断方法

2018年7月9日 15:22

```
package cn.itcast_02;
```

```
/*
```

```
 * 判断功能（非静态方法）
```

```
 * boolean equals(object obj):比较字符串的内容是否相同
```

```
 * boolean equalsIgnoreCase(String str): 比较字符串的内容是否相同，忽略大小写
```

```
 * boolean contains(String str):判断大字符串中是否包含小字符串
```

```
 * boolean startsWith(String str):判断字符串是否以str开头
```

```
 * boolean endsWith(String str):判断字符串是否以str结尾
```

```
 * boolean isEmpty():判断字符串内容是否为空
```

```
*/
```

```
public class StringDemo {
```

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

```
        String s1 = "helloworld";
```

```
        String s2 = "helloworld";
```

```
        String s3 = "Helloworld";
```

```
        // boolean equals(object obj)
```

```
        System.out.println("equals:" + s1.equals(s2));
```

```
        System.out.println("equals:" + s1.equals(s3));
```

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

```
        // boolean equalsIgnoreCase(String str)
```

```
        System.out.println("equals:" + s1.equalsIgnoreCase(s2));
```

```
        System.out.println("equals:" + s1.equalsIgnoreCase(s3));
```

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

```
        // boolean contains(String str)
```

```
        System.out.println("equals:" + s1.contains("hello")); // true
```

```
        System.out.println("equals:" + s1.contains("hw")); // false
```

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

```
        // boolean startsWith(String str)
```

```
        System.out.println("startswith:" + s1.startsWith("h"));
```

```
        System.out.println("startswith:" + s1.startsWith("hello"));
```

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

```
        // boolean isEmpty()
```

```
        System.out.println("isempty:" + s1.isEmpty());
```

```
        String s4 = "";
```

equals:true

equals:false

equals:true

equals:true

equals:true

equals:false

startswith:true

startswith:true

isempty:false

isempty:true

```
        System.out.println("isempty:" + s4.isEmpty());  
    }  
}
```

登录验证

2018年7月9日 15:56

```
package cn.itcast_02;

import java.util.Scanner;

public class StringTest {
    public static void main(String[] args) {
        String username = "admin";
        String password = "admin";
        for (int x = 0; x < 3; x++) {
            Scanner sc = new Scanner(System.in);
            System.out.println("请输入用户名");
            String name = sc.nextLine();
            System.out.println("请输入密码");
            String pwd = sc.nextLine();

            if (name.equals(username) && pwd.equals(password)) {
                System.out.println("登录成功");
                break;
            } else {
                if (x != 2) {
                    System.out.println("登录失败,你还有" + (2 - x) + "次机会");
                } else {
                    System.out.println("账号被冻结");
                }
            }
        }
    }
}
```

String类的获取功能

2018年7月9日 17:02

```
package cn.itcast_03;

/*
 * String类的获取功能（public）：
 *      int length():获取字符串的长度
 *      char charAt(int index):获取指定索引处的字符
 *      int indexOf(int ch):返回指定字符在次字符串中第一次出现处的索引
 *          为什么是int: 'a'和97都可以代表a
 *      int indexOf(String str):返回指定字符在次字符串中第一次出现处的索引
 *      char charAt(int index,int fromIndex):返回指定字符在次字符串中指定位置后
第一次出现处的索引
 *      char charAt(String str,int fromIndex)
 *      String substring(int start):从指定位置截取字符串，默认到末尾
 *      String substring(int start, int end)
 */
public class StringDemo {
    public static void main(String[] args) {
        String s = "helloworld";

        // int length()
        System.out.println("s.length:" + s.length());

        // char charAt(int index)
        System.out.println("charat:" + s.charAt(7));

        // int indexOf(int ch)
        System.out.println("indexof:" + s.indexOf('l'));

        // int indexOf(String str)
        System.out.println("indexof:" + s.indexOf("owo"));

        // char charAt(int index,int fromIndex)
        System.out.println("indexof:" + s.indexOf('l', 4));
        System.out.println("indexof:" + s.indexOf('l', 40)); // -1

        // String substring(int start) 包含start这个索引
    }
}
```

```
System.out.println("substring:" + s.substring(5));// world

// String substring(int start, int end)包左不包右
System.out.println("substring:" + s.substring(3, 8));// lowor

}

}
```

遍历字符串

2018年7月9日 23:02

//遍历字符串中的每一个字符

```
public class StringTest {  
    public static void main(String[] args) {  
        String s = "helloworld";  
        for (int x = 0; x < s.length(); x++) {  
            System.out.println(s.charAt(x));  
        }  
    }  
}
```

练习

2018年7月9日 23:08

//统计一个字符串大写字符, 小写字符, 数字字符出现的次数

```
public class StringTest2 {  
    public static void main(String[] args) {  
        String s = "Hello123World";  
        int bigCount = 0;  
        int smallCount = 0;  
        int numCount = 0;  
  
        for (int x = 0; x < s.length(); x++) {  
            char ch = s.charAt(x);  
            if (ch >= 'a' && ch <= 'z') {  
                smallCount++;  
            } else if (ch >= 'A' && ch <= 'Z') {  
                bigCount++;  
            } else {  
                numCount++;  
            }  
        }  
        System.out.println("大写字母"+bigCount+"个");  
        System.out.println("小写字母"+smallCount+"个");  
        System.out.println("数字字符"+numCount+"个");  
    }  
}
```

String的转换功能

2018年7月9日 23:24

```
package cn.itcast_04;

/*
 * String的转换功能
 *      byte[] getBytes():把字符串转换为字节数组
 *      char[] toCharArray():把字符串转换为字符数组
 *      static String valueOf(char[] chs):把字符数组转成字符串
 *      static String valueOf(int i):把int类型数据转成字符串
 *      valueOf可以把任意类型的数据转换成字符串
 *      String toLowerCase():把字符串转换成小写
 *      String toUpperCase():把字符串转换成大写
 *      String concat(String str):把字符串拼接
 */
public class StringDemo {
    public static void main(String[] args) {
        String s = "JavaSE";

        byte[] bys = s.getBytes();
        for(int x=0;x<bys.length;x++) {
            System.out.println(bys[x]);
        }
        System.out.println("-----");

        char[] chs = s.toCharArray();
        for(int x=0;x<chs.length;x++) {
            System.out.println(chs[x]);
        }
        System.out.println("-----");

        String ss=String.valueOf(chs);
        System.out.println(ss);
        System.out.println("-----");

        int i =100;
        String sss= s.valueOf(i);
        System.out.println(sss);
        System.out.println("-----");
    }
}
```



```
System.out.println(s.toLowerCase());  
System.out.println(s);//s本身不变  
System.out.println("-----");
```

```
System.out.println(s.toUpperCase());  
System.out.println("-----");
```

```
String s1="hello";  
String s2="world";  
String s4=s1.concat(s2);  
System.out.println(s4);
```

```
}
```

```
}
```

练习

2018年7月9日 23:46

```
package cn.itcast_04;

//首字母大写
public class StringTest {
    public static void main(String[] args) {
        String s = "helloWORLD";
        String s1 = s.substring(0, 1);
        String s2 = s.substring(1);
        String s3 = s1.toUpperCase();
        String s4 = s2.toLowerCase();
        String s5 = s3.concat(s4);
        System.out.println(s5);
    }
}
```

string的其它功能

2018年7月10日 11:29

```
package cn.itcast_05;
```

```
/*
 * String类的其它功能:
 * 替换功能:
 *      String replace(char old,char new)
 *      String replace(String old,String new)
 * 去除字符串两端空格
 *      String trim()
 * 按字典顺序比较两个字符串
 *      int comparaTo(String str)
 *      int compareToIgnoreCase(String str):不区分大小写
 */
```

```
public class StringDemo {
    public static void main(String[] args) {
        // 替换功能
        String s1 = "helloworld";
        String s2 = s1.replace('l', 'k');
        System.out.println(s2);
        String s3 = s1.replace("owo", "ak47");
        System.out.println(s3);

        // 去除功能
        String s4 = " hello world ";
        String s5 = s4.trim();
        System.out.println(s5);

        // 按字典顺序比较两个字符串
        String s6 = "hello";
        String s7 = "hello";
        String s8 = "abc";
        String s9 = "xyz";
        System.out.println(s6.compareTo(s7)); // 0
        System.out.println(s6.compareTo(s8)); // 7
        System.out.println(s6.compareTo(s9)); // -16
    }
}
```

```
public class StringTest {
    public static void main(String[] args) {
        String s1="hello";
        String s2="hel";
        System.out.println(s1.compareTo(s2)); //2
    }
}
```

练习

2018年7月10日 12:03

```
package cn.itcast_07;
```

```
//把代码用功能实现
```

```
public class StringTest2 {
```

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


```
        int[] arr = { 1, 2, 3 };
```

```
        String result = arrToStr(arr);
```

```
        System.out.println(result);
```

```
    }
```

静态方法只能访问静态方法



```
    public static String arrToStr(int[] arr) {
```

```
        String s = "";
```

```
        s += "[";
```

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

```
            if (x == arr.length - 1) {
```

```
                s += arr[x] + "];
```

```
            } else {
```

```
                s += arr[x] + ", ";
```

```
            }
```

```
        }
```

```
        return (s);
```

```
    }
```

```
}
```

字符串的反转

2018年7月10日 12:35

```
import java.util.Scanner;
```

```
//字符串反转
```

```
public class StringTest3 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("请输入一个字符串");  
        String s = sc.nextLine();  
        String result="";  
        char[] chs =s.toCharArray();  
        for(int x=chs.length-1;x>=0;x--) {  
            result+=chs[x];  
        }  
        System.out.println("反转后的结果是"+result);  
    }  
}
```

练习

2018年7月10日 16:00

```
package cn.itcast_07;

/*
 *    统计大串中小串的个数
 */
public class StringTest4 {
    public static void main(String[] args) {
        String s1 =
            "woaijavawozhenaijavawozhendeijavawoshizhenaijavaniaijavama";
        String s2="java";
        System.out.println("出现次数为"+counts(s1, s2));

    }
    public static int counts(String s1,String s2) {
        int num=0;
        int index=0;
        index=s1.indexOf(s2);
        while(index!=-1) {
            num++;
            index=index+s2.length();
            s1=s1.substring(index);
            index=s1.indexOf(s2);
        }
        return num;
    }
}
```