

【济南中心】JAVA 编程阶梯：基础篇之第十二章

- 常见对象 Scanner：

如果一个程序总是按照既定的流程运行，无需处理用户动作，这个程序总是比较简单的。实际上绝大部分程序都需要处理用户动作，包括接收用户的键盘录入，鼠标动作等。Scanner 类就是 JDK5 以后用于获取用户的键盘输入的一个类。

构造方法

```
public Scanner(InputStream source)
```

基本格式

hasNextXxx() 判断是否还有下一个输入项,其中 Xxx 可以是 Int,Double 等。如果需要判断是否包含下一个字符串，则可以省略 Xxx

nextXxx() 获取下一个输入项。Xxx 的含义和上个方法中的 Xxx 相同

默认情况下，Scanner 使用空格，回车等作为分隔符

常用方法

```
public int nextInt():获取一个 int 类型的值
```

```
public String nextLine():获取一个 String 类型的值
```

举个小 Demo：

```
class Demo {  
  
    public static void main(String[] args) {  
        System.out.println("请输入一个 int 类型的值");  
        int nextInt = new Scanner(System.in).nextInt();  
    }  
}
```

```

        System.out.println("请输入一个 String 类型的值");
    }

    String nextLine = new
Scanner(System.in).nextLine();

    System.out.println("输入的 int 的值是:" + nextInt
+ ", 输入的 String 的值是: " + nextLine);
}
}

```

• 常见对象 String :

字符串是由多个字符组成的一串数据(字符序列), 字符串可以看成是字符数组

构造方法

```

class Demo {

    public static void main(String[] args) {
        // A:String():无参构造
        String s = new String();
        s = "hello";
        System.out.println(s);
        // B:String(byte[] bys):传一个字节数组作为参数
        byte[] bys = { 97, 98, 99, 100, 101 };
        String s1 = new String(bys);
        System.out.println(s1);
        // C:String(byte[] bys,int index,int length):把
字节数组的一部分转换成一个字符串
        String s2 = new String(bys, 1, 2);
        System.out.println(s2);
        // D:String(char[] chs):传一个字符数组作为参数
        char[] chs = { 'a', 'b', 'c', 'd', 'e' };
        String s3 = new String(chs);
        System.out.println(s3);
        // E:String(char[] chs,int index,int length):把
字符数组的一部分转换成一个字符串
        String s4 = new String(chs, 1, 2);
        System.out.println(s4);
        // F:String(String str):把一个字符串传递过来作为
参数
        String s5 = new String(s);
        System.out.println(s5);
    }
}

```

```
        // G:直接把字符串常量赋值给字符串引用对象(最常用)

        String str = "hello";
        System.out.println(str);
    }
}
```

String 类的判断功能

boolean equals(Object obj):比较字符串的内容是否相同,区分大小写

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):判断字符串是否以某个指定的字符串开头

boolean endsWith(String str):判断字符串是否以某个指定的字符串结尾

boolean isEmpty():判断字符串是否为空。

String 类的常见面试题

String s = new String("hello");创建了几个对象？

两个，一个"hello"字符串对象，在方法区的常量池；一个 s 对象，在堆内存。

```
class Demo {

    public static void main(String[] args) {
        // 1. 判断定义为 String 类型的 s1 和 s2 是否相等
        String s1 = "abc";
        String s2 = "abc";
        System.out.println(s1 == s2); // true
        System.out.println(s1.equals(s2)); // true

        // 2. 判断定义为 String 类型的 s1 和 s2 是否相等
        String s3 = new String("abc");
        String s4 = "abc";
        System.out.println(s3 == s4); // false
        System.out.println(s3.equals(s4)); // true
    }
}
```

```

        // 3. 判断定义为 String 类型的 s1 和 s2 是否相等
        String s5 = "a" + "b" + "c";
        String s6 = "abc";
        System.out.println(s5 == s6); // true
        System.out.println(s5.equals(s6)); // true

        // 4. 判断定义为 String 类型的 s1 和 s2 是否相等
        String s7 = "ab";
        String s8 = "abc";
        String s9 = s7 + "c";
        System.out.println(s9 == s8); // false
        System.out.println(s9.equals(s8)); // true
    }
}

```

String 类的获取功能

int length(): 获取字符串的长度。

char charAt(int index): 获取指定索引位置的字符

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。

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。

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。

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

int lastIndexOf(): 用法和 indexOf 一样

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

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。

字符串的遍历

```

class Demo {

    public static void main(String[] args) {
        String str = "abcdef";
        for (int i = 0; i < str.length(); i++) {
            System.out.print(str.charAt(i)+
", ");
        }
    }
}

```

String 类的转换功能

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 类型的数据转成字符串。

注意：String 类的 valueOf 方法可以把任意类型的数据转成字符串。

```

/**
 * 取出一个字符串中字母出现的次数。如：字符串："abcdekka27qoasd123q"，输出格
 * 式为：a(3)b(1)c(1)k(2)...
 */
public class Demo {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String s = "abcdekka27qoasd123q"; // 定义一个字
        符串类型变量，存放欲统计的字符串
        count(s);
    }

    public static void count(String s) {
        char[] c = s.toCharArray(); // 将字符串转换为字
        符数组，以便操作

        int[] count = new int[c.length]; // 创建一个 int
        数组，长度等于字符串（字符数组）的长度，用于存放字符出现的次数
        /**
         * 循环统计字符出现的次数，将次数存入 count 数组
         * i 代表 char 数组中正在统计的字符的角标，j 代表被比较字符的角标

```

```

        */
        for (int i = 0; i < c.length; i++) {
            if (!Character.isLetter(c[i]))
                continue;
            for (int j = 0; j < i; j++) { //
                if (c[i] == c[j])
                    count[j]++; // 角标大的元素对应 count 中数值设为-1，并进行下个字符的判断；
            }
            count[i] = -1;
            continue;
        }
        count[i]++; // 如果当前字符与前面每一个字符都不同，则代表它是第一次出现，计数加 1
    }

    // 进行循环输出
    for (int i = 0; i < c.length; i++) {
        if (count[i] == 0) // 如果字符 c[i] 对应角标为 0，则表示它不是字母或者不是第一次出现的字母，不做输出
            continue;
        System.out.print(c[i] + "(" + count[i] + ")"); // 按要求格式输出
    }
}

```

String 类的其他功能

tring 的替换功能及案例演示

String replace(char old,char new)

String replace(String old,String new)

String 的去除字符串两空格及案例演示

String trim()

int compareTo(String str)

int compareToIgnoreCase(String str)

String toLowerCase():把字符串转成小写

String toUpperCase():把字符串转成大写

String concat(String str):把字符串拼接

```
/**
 * 编写程序，循环接收用户从键盘输入多个字符串，直到输入“end”时循环结束， 并将所有已输入的
 * 字符串按字典顺序倒序打印。
 * @author Somnus
 *
 */
public class Demo {
    private static StringBuilder sb = new StringBuilder();

    public static void main(String[] args) {
        // 循环接收键盘输入的字符串
        while (true) {
            // 接收键盘输入的字符串
            String s = new Scanner(System.in).nextLine();
            // 判断输入的是否是：end
            if (s.equals("end")) {
                // 是，退出 JVM
                sb.reverse();
                System.out.println(sb);
                System.exit(0);
            }
            sb.append(s);
        }
    }
}
```



识别二维码 关注黑马程序员视频库
免费获得更多 IT 资源