

网站:<http://www.runoob.com>

## 一.Dos命令:

### DOS命令操作

- 1.快捷键：windows+R 呼出DOS窗口。
- 2.输入cmd(大小写不用区分)回车，打开DOS窗口。

常见的命令（基本都不区分大小写）

- 1.切换盘符： 盘符（就是电脑的C,D,E..盘）：（如： D：）
- 2.查看文件或者文件夹： dir
- 3.进入某个文件夹： cd 文件夹名
- 4.返回上一级目录： cd..
- 5.清屏： cls
- 6.删除某个文件： del 文件名
- 7.删除某个文件夹： rd 文件夹名
- 8.退出DOS窗口： exit

```
//编译:java->.class    字节码
//反编译:.class->java

//编译命令
//javac com/lanou/Main.java
//执行命令
//javac com/lanou/Main
//注:执行时要考虑包路径

//编译时指定代码格式
//javac -encoding utf-8 com/lanou/Main.java

//执行时,传参
//java com.lanou.Main a b c

//args:用于接收传过来的参数

System.out.println(Arrays.toString(args));

//新建一个星期的枚举类型

//new:从堆区开辟空间
//构造方法:初始化
Girl girl = new Girl( name: "刘亦菲", gender: "女");
girl.eat();
```

## 二.java运算符

- 算术运算符
- 关系运算符
- 位运算符
- 逻辑运算符
- 赋值运算符

补充疑点:前缀自增自减法(++a,--a): 先进行自增或者自减运算, 再进行表达式运算

后缀自增自减法(a++,a--): 先进行表达式运算, 再进行自增或者自减运算

位运算符:

程序中的所有数在计算机内存中都是以二进制的形式储存的。位运算就是直接对整数在内存中的二进制位进行操作

位运算的特点:

1.& (与)、| (或)、^(异或)、~ (非/取反)

2.>>和<<运算符将二进制位进行右移或者左移操作。

3.>>>运算符将用0填充高位；>>运算符用符号位填充高位，没有<<<运算符。

对于int型，1<<35与1<<3是相同的，而左边的操作数是long型是需要对右侧的操作数作数模64。

与：相同为1，或：有一个为1结果为1，异或：相同为0，不同为1。

```
1 @Test
2 public void test2() {
3
4     //对应的位运算
5     int a = 10; //00001010
6     int b = 7;  //00000111
7
8     //位运算符
9     //按位与:&,同时为1才为1,否则为0;
10    int result = a & b;
11    System.out.println(result);
12
13    //按位或:|,同时为0才为0,否则为1;
14    result = a | b;
15    System.out.println(result);
16
17    //按位异或:^相同为0,不同为1;
18    result = a ^ b;
19    System.out.println(result);
20
21    //按位取反:1变0,0变1
22    byte c = 10; //00001010
23    System.out.println(~c);
24
25    //原码:
26    //反码:
27    //补码:
28
29    //正数的原码,反码,补码都一样
30    //eg:10
31    //原码:00001010
32    //反码:00001010
33    //补码:00001010
```

```

34
35 //eg: -10
36 //原码:10001010
37 //反码:11110101(在源码的基础上,符号为不变,按位取反)
38 //补码:11110110(在反码的基础上,+1)
39
40 //注:数字存到计算机中,是以补码形式出现
41
42 //左移:高位舍去,低位补零
43 byte d = 3;//00000011
44 result = d << 2;//00001100
45 System.out.println(result);
46
47 byte e = -1;
48 result = e << 2;
49 System.out.println(result);
50 //右移:低位舍去,高位补零,正数高位补0,负数高位补1;
51 result = d >> 2;
52 System.out.println(result);
53
54 //无符号右移:忽略了符号位拓展,
55 result = e >>> 2;
56 System.out.println(result);
57
58 }
59

```

```

1 @Test
2 public void test3() {
3     int a = 20;
4     int result = a & 1;
5     if (result == 1) {
6         System.out.println("奇数");
7     } else {
8         System.out.println("偶数");
9     }
10 }

```

```

11
12 @Test
13 public void test4() {
14     //对于任何数x,都有x^x=0,x^0=x;
15     int a = 10, b = 5;
16     a = a ^ b;
17     b = a ^ b;
18     a = a ^ b;
19     System.out.println("a = " + a);
20     System.out.println("b = " + b);
21 }

```

### 三.正则表达式:判断字符串的格式是否正确,判断字符串是否包含某些内容

```

1 @Test
2 public void test6() {
3     //正则表达式:判断字符串的格式是否正确,判断字符串是否包含某些内容
4     String value = "1";
5     if (value.matches("[1-9]|[1-9][0-9]|100")) {
6         //一位的[0-9],\\d
7         //两位\\d\\d,\\d{5,8},最大8位,最小5位.
8         //1-100 [1-9]|[1-9][0-9]|100
9         int number = Integer.parseInt(value);
10        System.out.println(number);
11    } else {
12        System.out.println("格式不正确!!!");
13    }
14
15 }
16
17 @Test
18 public void test7() {
19     //2月13日
20     String str = "2月13日";
21     if (str.matches("\\d{2}月\\d{2}日")) {
22     } else {
23
24     }
25

```

#### 四,数组:遍历数组,打印,排序(冒泡排序)

```
1 @Test
2 public void test8() {
3     //练习:
4     //创建10个元素的整形数组,随机赋值[10,24],
5     //按从小到大进行排序(1.中间变量排序2.通过位运算排序3. Arrays.sort(a))
6     Random random = new Random();
7     int[] a = new int[10];
8     for (int i = 0; i < a.length; i++) {
9         a[i] = random.nextInt(15) + 10;
10    }
11    System.out.print(Arrays.toString(a));
12    Arrays.sort(a);
13    System.out.println();
14    for (int i = 0; i < a.length - 1; i++) {
15        for (int j = 0; j < a.length - 1 - i; j++) {
16            if (a[j] > a[j + 1]) {
17                a[j] = a[j] ^ a[j + 1];
18                a[j + 1] = a[j] ^ a[j + 1];
19                a[j] = a[j] ^ a[j + 1];
20            }
21        }
22    }
23    System.out.println(Arrays.toString(a));
24 }
25 }
```

#### 五.注解与dos命令的使用,java8的新特性