jdk 源码：Integer.toUnsignedString0

作者：Maggie
原文链接：https://ld246.com/article/1537868508493
来源网站：链滴

1. 应用：转换成对应进制的字符串

```
// 转换成二进制字符串
public static String toBinaryString(int i) {
        return toUnsignedString0(i, 1);
    }

// 转换成八进制字符串
public static String toOctalString(int i) {
        return toUnsignedString0(i, 3);
    }

// 转换成十六进制字符串
public static String toHexString(int i) {
        return toUnsignedString0(i, 4);
    }

// 1-二进制， 3-八进制(三位二进制数表示一位八进制数)， 4-十六进制(四位二进制数表示一位十六
制数)
private static String toUnsignedString0(int val, int shift) {
        // assert shift > 0 && shift <=5 : "Illegal shift value";
        int mag = Integer.SIZE - Integer.numberOfLeadingZeros(val);
        int chars = Math.max(((mag + (shift - 1)) / shift), 1);
        char[] buf = new char[chars];

        formatUnsignedInt(val, shift, buf, 0, chars);

        // Use special constructor which takes over "buf".
        return new String(buf, true);
    }
```

2. 测试代码

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

        System.out.println(Integer.toBinaryString(10));
        System.out.println(Integer.toOctalString(10));
        System.out.println(Integer.toHexString(10));

        System.out.println(Integer.toBinaryString(-1073741824));
        System.out.println(Integer.toOctalString(-10));
        System.out.println(Integer.toHexString(-10));

    }
```

运行结果：

```
1010
12
a
11000000000000000000000000000000
37777777766
```

原文链接：jdk 源码：Integer.toUnsignedString0

fffffff6

3. 源码分析

```
int mag = Integer.SIZE - Integer.numberOfLeadingZeros(val);
```

● 32减去numberOfLeadingZeros， 表示实际表示此数字只需要的位数。比如10的二进制补码是000 0000 0000 0000 0000 0000 0000 1010，它实际只需要1010这4位数字就可以代表10. 所以10的mg就是4

```
int chars = Math.max(((mag + (shift - 1)) / shift), 1);
```

● 表示转换成字符数组所需要的长度，其中shift的值1-二进制，3-八进制，4-十六进制。比如10的ma是4，如果要转成二进制shift=1，则chars=4,因为需要长度为4的字符数组来存放1010。 如果要转成6进制a,则shift=4，得出chars=1，因为只需要长度为1的字符串数组来存放结果a.

```
char[] buf = new char[chars];
formatUnsignedInt(val, shift, buf, 0, chars);
```

● formatUnsignedInt(val, shift, buf, 0, chars)此方法将数字转换为字符数据存放在buf中

作者 @没有故事的老大爷

别想太多，想不过来

1

4. formatUnsignedInt(val, shift, buf, 0, chars)方法分析

```
    static int formatUnsignedInt(int val, int shift, char[] buf, int offset, int len) {
        int charPos = len;
        int radix = 1 << shift;
        int mask = radix - 1;
        do {
            buf[offset + --charPos] = Integer.digits[val & mask];
            val >>>= shift;
        } while (val != 0 && charPos > 0);

        return charPos;
    }
```

● charPos字符位置

● radix 根据shift计算出进制

● mask 掩码，比进制radix小1

● 循环体： 巧妙之处是val & mask通过与运算和Integer.digits数组的设计计算出对应进制的最后一数据，然后val再向右便宜

比如val=10，shift=3。表示将10转为8进制12.则radix=8, mask=7

循环第一次：buf1 = Integer.digits[1010 & 0111] = Integer.digits[0010] = 2

```
    final static char[] digits = {
        '0' , '1' , '2' , '3' , '4' , '5' ,
        '6' , '7' , '8' , '9' , 'a' , 'b' ,
```

```
        'c' , 'd' , 'e' , 'f' , 'g' , 'h' ,
        'i' , 'j' , 'k' , 'l' , 'm' , 'n' ,
        'o' , 'p' , 'q' , 'r' , 's' , 't' ,
        'u' , 'v' , 'w' , 'x' , 'y' , 'z'
    };
```

然后val右移三位：1010>>>3 = 0001

循环第二次：buf[0] = Integer.digits[0001 & 0111] = Integer.digits[0001] = 1

至此：buf = ['1','2']， 也就将10转为8进制12。二进制和十六进制原理一样。

作者 @没有故事的老大爷

经营自己的圈子

1