

1 Integer 类方法

- 1.1 toXXXString —— 将 int 转成不同进制的 String 形式
- 1.2 parseInt(...) —— 将各种格式的 String 转成 int
- 1.3 valueOf —— 将 String、int 转成 Integer
- 1.4 getInteger —— 获取系统参数值，并转换为 Integer
- 1.5 compare —— int 值比较
- 1.6 unsigned int 的除/取余
- 1.7 取 int 值最高/低位 1 对应的 值
- 1.8 LeadingZeros/TrailingZeros/countBit
- 1.9 int 对应二进制的反转
- 1.10 sinum/sum/max/min
- 1.11 decode/hashCode/stringSize/toUnsignedLong
- 1.12 构造函数 Integer(String)

2 重点方法解析

参考

1 Integer 类方法

- 返回类型标 * 的是重点方法

1.1 toXXXString —— 将 int 转成不同进制的 String 形式

返回类型	方法名	描述
String	toString()	toString(value)
* static String	toString(int i)	有意思，这和 i+"" 区别在哪呢？内部调用 getChar()
* static String	toString(int i, int radix)	将 int 值转成 对应进制的 String 形式
static String	toUnsignedString(int i)	toUnsignedString(int i, 10)
static String	toUnsignedString(int i, int radix)	Long.toUnsignedString(int i, int radix)
static String	toHexString(int i)	toUnsignedString0(i, 4)
static String	toOctalString(int i)	toUnsignedString0(i, 3)
static String	toBinaryString(int i)	toUnsignedString0(i, 1)
* static String	toUnsignedString0(int val, int shift)	内部是全是通过 val & mask, val >>= shift, 所以仅仅是通过 mask 取到每一部分的值对应的 radix, 所以是无符号的
* static int	formatUnsignedInt(...)	toUnsignedString0 内部调用它, 进行的掩模计算

1.2 parseInt(...) —— 将各种格式的 String 转成 int

返回类型	方法名	描述
static int	parseInt(String s)	parseInt(s,10)
* static int	parseInt(String s, int radix)	根据对应的进制，将 s 解析成对应的 int 值，用负数算 (有意思)
static int	parseUnsignedInt(String s)	parseUnsignedInt(s, 10)
* static int	parseUnsignedInt(String s, int radix)	内部调用 Long.parseLong(s,radix)

1.3 valueOf —— 将 String、int 转成 Integer

返回类型	方法名	描述
static Integer	valueOf(String s)	Integer.valueOf(parseInt(s, 10))
static Integer	valueOf(String s, int radix)	Integer.valueOf(parseInt(s, radix))
static Integer	valueOf(int i)	内部走Cache,通过 java.lang.Integer.IntegerCache.high 设置 Cache 的最大范围, [-128,max]

1.4 getInteger —— 获取系统参数值，并转换为 Integer

返回类型	方法名	描述
static Integer	getInteger(String nm)	getInteger(nm, null)
static Integer	getInteger(String nm, int val)	内部调用 getInteger(nm, null)
static Integer	getInteger(String nm, Integer val)	获取系统 property 参数值, v = System.getProperty(nm), 如果为 null, 返回 val 值

1.5 compare —— int 值比较

返回类型	方法名	描述
int	compareTo(Integer anotherInteger)	compare(this.value, anotherInteger.value)
static int	compare(int x, int y)	(x < y) ? -1 : ((x == y) ? 0 : 1)
static int	compareUnsigned(int x, int y)	compare(x + MIN_VALUE, y + MIN_VALUE)

1.6 unsigned int 的除/取余

返回类型	方法名	描述
static int	divideUnsigned(int dividend, int divisor)	(int)(toUnsignedLong(dividend) / toUnsignedLong(divisor))
static int	remainderUnsigned(int dividend, int divisor)	(int)(toUnsignedLong(dividend) % toUnsignedLong(divisor));

1.7 取 int 值最高/低位 1 对应的值

返回类型	方法名	描述
* static int	highestOneBit(int i)	最高位取最高 1 及后面全置0的值, 代码解析
* static int	lowestOneBit(int i)	i & -i : 精辟

1.8 LeadingZeros/TrailingZeros/countBit

返回类型	方法名	描述
* static int	numberOfLeadingZeros(int i)	i 对应 2 进制的开头 0 的个数
* static int	numberOfTrailingZeros(int i)	i 对应 2 进制的末尾 0 的个数
* static int	bitCount(int i)	i 对应 2 进制中 1 的个数

1.9 int 对应二进制的反转

返回类型	方法名	描述
* static int	rotateLeft(int i, int distance)	(i << distance) (i >>> -distance)
* static int	rotateRight(int i, int distance)	(i >>> distance) (i << -distance)
* static int	reverse(int i)	二进制按位反转
* static int	reverseBytes(int i)	二进制按byte反转

```
00000000_00000000_01001101_00011111 -> 原始
11111000_10110010_00000000_00000000 -> 按位反转
00011111_01001101_00000000_00000000 -> 按字节反转
```

1.10 sinum/sum/max/min

返回类型	方法名	描述
static int	signum(int i)	负数返回 -1, 0 返回 0, 正数返回 1
static int	sum(int a, int b)	a + b
static int	max(int a, int b)	Math.max(a, b)
static int	min(int a, int b)	Math.min(a, b)

1.11 decode/hashCode/stringSize/toUnsignedLong

返回类型	方法名	描述
static long	toUnsignedLong(i)	$((\text{long}) x) \& 0\text{xffffffffL}$ ，高32都是0，低32是 int 的值，所以肯定是正
static int	stringSize(int x)	x 的位数，x 必须为非负
int	hashCode()	Integer.hashCode(value)
static int	hashCode(int value)	value
* static Integer	decode(String nm)	根据 nm 判断对应的进制数,根据 nm 判断对应的进制数，再调用 Integer.valueOf(nm.substring(index), radix)

1.12 构造函数 Integer(String)

- 内部调用 parseInt(s,10)
- 如果确定 int 很小，推荐使用 Integer.valueOf(String)：这样可以走 cache

2 重点方法解析

- 留待以后补充

参考

jdk1.8_171