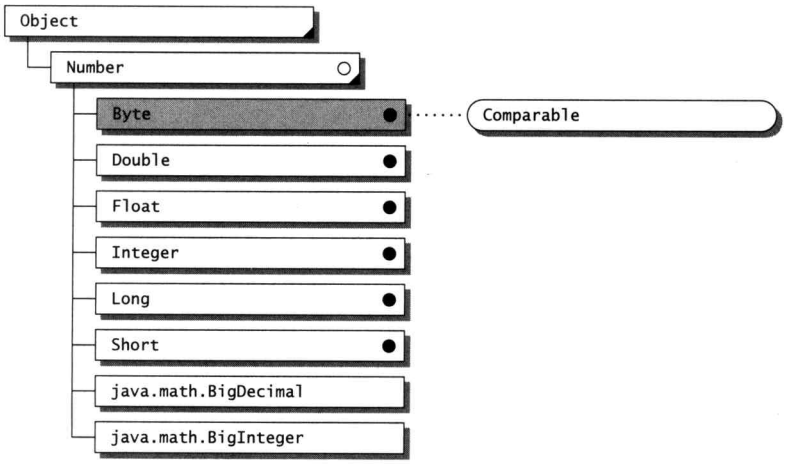
java.lang

Byte



语法

public final class Byte extends Number implements Comparable

描述

Byte(字节)类提供了byte数据值的一个对象包容器。一个在java中的byte数据值是一个8位带符号的整数。这个对象包容器允许字节传输给在接受java对象作为其参数的java类库中的方法。这个类也提供了将数值转化为byte数值或相反操作的方法。

如想获得关于这个类的更多信息，请参见《The Java Class Libraries，Second Edition，Volume 1》。

版本1.2中所作的修改

Byte类实现了Comparable接口，并有compareTo()方法的两个重载形式用来比较和排列Byte对象。这使得Byte对象的向量可以被排序，如下例所示。

```
Byte[] ba = new Byte[]{new Byte((byte)10),
                       new Byte((byte)-4), new Byte((byte)48)};
Arrays.sort(ba);
for (int i = 0; i < ba.length; i++) {
    System.out.println(ba[i]);
}
// -4, 10, 48
```

成员概述	
构造函数	
Byte()	使用一个byte值或一个字符串来构造一个Byte实例。
成员函数	
byteValue()	以byte型检索这个Byte的值。

(续)

成员概述	
doubleValue()	以double型检索这个Byte的值。
floatValue()	以float型检索这个Byte的值。
intValue()	以int型检索这个Byte的值。
longValue()	以long型检索这个Byte的值。
shortValue()	以short型检索这个Byte的值。
字节相关常量	
MAX_VALUE	一个byte可以有的最大值。
MIN_VALUE	一个byte可以有的最小值。
TYPE	代表基本类型byte的Class对象。
比较方法	
compareTo()	比较这个Byte与另一个Byte的带符号的数字值的大小。
equals()	决定是否这个Byte与另一个对象相等。
字符串转化方法	
decode()	分析一个以8位带符号的整数表示的字符串并把它转化为一个Byte。
parseByte()	分析一个以整数表示的字符串并把它转化为一个 byte。
valueOf()	创建一个使用它的字符串表示的 Byte实例。
toString()	产生一个用byte或Byte表示的字符串。
对象方法	
hashCode()	计算这种Byte的哈希码。

参见

Comparable、java.util.Arrays。
《The Java Class Libraries，Second Edition，Volume 1》中的Byte。

1.2 compareTo()

目的 比较这个Byte与另一个Byte的带符号的数字值的大小。

语法
public int compareTo(Byte anotherByte)
public int compareTo(Object anotherByte)

描述 这种方法比较这个 Byte(带符号)与另一个Byte的数值。它返回一个整数标志是否两个数字值相等，如果它们不相等，对它们进行排序。如果两个值相等，它返回 0。如果这个Byte较anotherByte的数值小，返回一个负数。 如果这个Byte较anotherByte的数值大，返回一个正数。
如果anotherByte不是一个Byte实例，则抛出一个ClassCastException。

参数

anotherByte 被比较的非空Byte。

返回 如果这个 Byte与anotherByte的数值相等，返回 0；如果这个 Byte较anotherByte的数值小，返回一个负数；如果这个 Byte较anotherByte的数值大，返回一个正数。

异常

ClassCastException

如果anotherByte**不是**Byte**的一个实例。**

参见 Comparable。

示例

```
Byte b = new Byte((byte)100);
System.out.println(b.compareTo(new Byte((byte)-1)));    // positive
System.out.println(b.compareTo(new Byte((byte)100)));   // 0
System.out.println(b.compareTo(new Byte((byte)-100)));  // positive
System.out.println(b.compareTo(new Byte((byte)101)));   // negative
System.out.println(b.compareTo("abc")); // ClassCastException
```