
包装类及常用方法简介

JAVA 是一种面向对象语言，JAVA 中的类把方法与数据连接在一起，构成了自包含式的处理单元。但在 JAVA 中不能定义基本类型(primitive type)对象，为了能将基本类型视为对象进行处理，并能连接相关的方法，JAVA 为每个基本类型都提供了包装类。如 int 型数值的包装类 Integer，boolean 型数值的包装类 Boolean 等，这样便可以把这些基本类型转换为对象来处理了。下面将介绍 JAVA 中提供的各种包装类。

一、Integer

java.lang 包中的 Integer 类、Long 类和 Short 类，分别将基本类型 int、long 和 short 封装成一个类。由于这些类都是 Number 的子类，区别就是封装不同的数据类型，其包含的方法基本相同，下面以 Integer 类为例介绍：

Integer 类在对象中包装了一个基本类型 int 的值。该类的对象包含一个 int 类型的字段。此外，该类提供了多个方法，能在 int 类型和 String 类型之间互相转换，同时还提供了处理 int 类型时非常有用的其他一些常量和方法。

1、 构造方法有两种：

a) 以 int 型变量作为参数创建 Integer 对象，实例代码如下：

```
Integer number=new Integer(7);
```

b) 以 String 型变量作为参数创建 Integer 对象，实例代码如下：

```
Integer number=new Integer("7");
```

2、 常用方法：

Integer 类的常用方法

方法	返回值	功能描述
byteValue()	byte	以 byte 类型返回该 Integer 的值
compareTo (Integer anotherInteger)	int	在数字上比较两个 Integer 对象。如果这两个值相等，则返回 0；如果调用对象的数值小于 anotherInteger 的数值，则返回负值；如果调用对象的数值大于 anotherInteger 的数值，则返回正值
equals (Object IntegerObj)	boolean	比较此对象与指定对象是否相等
intValue ()	int	以 int 型返回此 Integer 对象
shortValue ()	short	以 short 型返回此 Integer 对象
toString()	String	返回一个表示该 Integer 值的 String 对象
valueOf(String str)	Integer	返回保存指定的 String 值的 Integer 对象
parseInt(String str)	int	返回包含在由 str 指定的字符串中的数字的等价整数值

3、 常用的常量：

- a) MAX_VALUE: 表示 int 型可取的最大值
- b) MIN_VALUE: 表示 int 型可取的最小值
- c) SIZE：表示以二进制补码形式表示 int 值的位数
- d) TYPE: 表示基本类型 Class 实例

示例如：

```
int maxint = Integer.MAX_VALUE;
```

```
int minint = Integer.MIN_VALUE;
```

```
int intsize = Integer.SIZE;
```

```
System.out.println("int 类型可取的最大值"+maxint);
```

```
System.out.println("int 类型可取的最小值"+minint);
```

```
System.out.println("int 类型的二进制位数"+intsize);
```

输出结果为：

int类型可取的最大值2147483647

int类型可取的最小值-2147483648

int类型的二进制位数32

二、 Boolean

Boolean 类将基本类型为 boolean 的值包装在一个对象中。一个 Boolean 类型的对象只包含一个类型为 boolean 的字段。此外，此类还为 boolean 和 String 的相互转换提供了许多方法，并提供了处理 boolean 时非常有用的其他一些常量和方法。

1、 构造方法：

e) 创建一个表示 value 参数的 boolean 对象，实例如下：

```
Boolean b = new Boolean(true);
```

f) 以 String 变量作为参数，创建 boolean 对象。此时，如果传入的字符串不为 null，且忽略大小写后的内容等于“true”，则生成 Boolean 对象值为 true，反之为 false。

示例如：

```
Boolean b1 = new Boolean("ok");
```

```
System.out.println(b1);
```

```
Boolean b2=new Boolean("TRUE");
```

```
System.out.println(b2);
```

输出结果为：

```
false  
true
```

2、 常用方法：

Boolean 类的常用方法

方法	返回值	功能描述
booleanValue ()	boolean	将 Boolean 对象的值以对应的 boolean 值返回
equals (Object obj)	boolean	判断调用该方法的对象与 obj 是否相等，当且仅当参数不是 null，而且与调用该方法的对象一样都表示同一个 boolean 值的 Boolean 对象时，才返回 true
parseBoolean(String s)	boolean	将字符串参数解析为 boolean 值
toString()	String	返回表示该 boolean 值的 String 对象
valueOf(String s)	boolean	返回一个用指定的字符串表示值的 boolean 值

其中 booleanValue()的功能是：将 booleanValue 对象的值以对应的 boolean 值返回。

示例如：

```
Boolean b1 = new Boolean("OK");
```

```
Boolean b2 = new Boolean(false);
```

```
System.out.println("b1:"+b1.booleanValue());
```

```
System.out.println("b2:" + b2.booleanValue());
```

结果如图：

```
b1:false  
b2:false
```

3、 常用的常量：

- a) TRUE:对应基值 true 的 Boolean 对象；
- b) FALSE:对应基值 false 的 Boolean 对象；
- c) TYPE:表示基本类型 Class 实例

三、 Byte

Byte 类将基本类型为 byte 的值包装在一个对象中，一个 byte 类型的对象只包含一个类型为 byte 的对象。此外，该类还为 byte 和 String 之间相互转换提供方法，并提供了一些处理 byte 时非常有用的常量。

1、 构造方法

Byte 类提供了两种构造方法的重载形式来创建 BYTE 类对象：

- a) Byte(byte value) 通过这种方法创建的 byte 对象 ,可表示指定的 byte 值。

例： byte mybyte=45;

```
Byte b = new Byte(mybyte);
```

- b) Byte(String str) 通过这种方法创建的 byte 对象 ,可表示 String 参数指定的 byte 值

例：Byte mybyte = new Byte("12")

2、 常用方法

Byte 类的常用方法

方法	返回值	功能描述
byteValue ()	byte	以一个 byte 值返回 Byte 对象
compareTo (Byte anotherByte)	int	在数字上比较两个 Byte 对象
doubleValue ()	double	以一个 double 值返回此 Byte 值
intValue ()	int	以一个 int 值返回此 Byte 的值
parseByte (String str)	byte	将 String 型参数解析成等价的字节 (byte) 形式
toString ()	String	返回此 Byte 的值的 String 对象
valueOf (String str)	Byte	返回一个保持指定 String 所输出的值的 Byte 对象
equals (Object obj)	boolean	将此对象与指定对象比较 ,如果调用该方法的对象与 obj 相等 , 则返回 true , 否则返回 false

3、 常用的常量

- a) MIN_VALUE: 最小值
- b) MAX_VALUE: 最大值
- c) SIZE: 二进制
- d) TYPE: 表示基本类型 byte 的 class 实例

四、Character

Character 类在对象中包装一个基本类型为 char 的值。一个 Character 对象包含类型为 char 的单个字段。

1、 构造方法

Character 类的构造方法的语法如下：

Character (char value)

2、 常用方法

Character 类的常用方法

方法	返回值	功能描述
charvalue ()	char	返回此 Character 对象的值
compareTo (CharacteranotherCharacter)	int	根据数字比较两个 Character 对象，若这两个对象相等则返回 0
equals (Object obj)	Boolean	将调用该方法的对象与指定的对象相比较
toUpperCase (char ch)	char	将字符参数转换为大写
toLowerCase (char ch)	char	将字符参数转换为小写
toString ()	String	返回一个表示指定 char 值的 String 对象
charValue ()	char	返回此 Character 对象的值
isUpperCase (char ch)	boolean	判断指定字符是否是大写字符
IsLowerCase (char ch)	boolean	判断指定字符是否是小写字符

示例如：

```
Character mychar1=new Character('A');
```

```
Character mychar2=new Character('a');
```

```
System.out.println(mychar1+"是大写字母吗  
"+Character.isUpperCase(mychar1));
```

```
System.out.println(mychar2+"是小写字母吗  
"+Character.isLowerCase(mychar2));
```

结果如下：

```
A是大写字母吗true  
a是小写字母吗true
```

五、Double

Double 和 Float 包装类是对 double、float 基本类型的封装，他们都是 Number 类的子类，又都是对小数进行操作，所以常用基本方法相同，Double 类在对象中包装一个基本类型为 double 的值，每个 double 类都包含一个 double 类型的字段。

- 1、 构造方法
- 2、 Double 类提供了两种构造方法来获得 Double 类对象
 - a) Double (double value): 基于 double 参数创建 Double 类对象；
 - b) Double (String str): 构造一个新分配的 Double 对象，表示用字符串表示的 Double 类型的浮点值
- 3、 常用方法

Double 类的常用方法

方法	返回值	功能描述
byteValue()	byte	以 byte 形式返回 Double 对象值(通过强制转换)
compareTo(Double d)	int	对两个 Double 对象进行数值比较。如果两个值相等，则返回 0；如果调用对象的数值小于 d 的数值，则返回负值；如果调用对象的数值大于 d 的

		数值，则返回正值
equals (Obj)	boolean	将此对象与指定的对象相比较
intValue()	int	以 int 形式返回 Double 值
isNaN ()	boolean	如果此 double 值是非数字 (NaN) 值，则返回 true；否则返回 false
toString ()	string	返回此 Double 对象的字符串表示形式
valueOf (String str)	double	返回保存参数字符串 str 表示的 double 值的 Double 对象
doubleValue ()	double	以 double 形式返回此 Double 对象
longValue ()	long	以 long 形式返回 double 的值 (通过强化转换为 long 类型)

六、Number

抽象类 Number 是父类，Number 的子类必须提供将表示的数值转换成 byte、double/float/long/int/short 的方法。Number 类的方法被 Number 的各子类所实现，常用方法如下：

Number 类的方法

方法	返回值	功能描述
byteValue ()	byte	以 byte 形式返回指定的数值
intValue ()	int	以 int 形式返回指定的数值
floatValue ()	float	以 float 形式返回指定的数值
shortValue ()	short	以 short 形式返回指定的数值
longValue ()	long	以 long 形式返回指定的数值
doubleValue ()	double	以 double 形式返回指定的数值