  Java语言是一个面向对象的语言，但是Java中的基本数据类型却是不面向对象的，这在实际使用时存在很多的不便，为了解决这个不足，在设计类时为每 个基本数据类型设计了一个对应的类进行代表，这样八个和基本数据类型对应的类统称为包装类(Wrapper Class)，有些地方也翻译为外覆类或数据类型类。

         包装类均位于[**Java**](http://lib.csdn.net/base/java).lang包，包装类和基本数据类型的对应关系如下表所示：

包装类对应表

|  |  |
| --- | --- |
| 基本数据类型 | 包装类 |
| byte | Byte |
| boolean | Boolean |
| short | Short |
| char | Character |
| int | Integer |
| long | Long |
| float | Float |
| double | Double |

         在这八个类名中，除了Integer和Character类以后，其它六个类的类名和基本数据类型一样，只是类名的第一个字母大写即可。

         对于包装类说，这些类的用途主要包含两种：

                   a、作为和基本数据类型对应的类类型存在，方便涉及到对象的操作。

                   b、包含每种基本数据类型的相关属性如最大值、最小值等，以及相关的操作方法。

         由于八个包装类的使用比较类似，下面以最常用的Integer类为例子介绍包装类的实际使用。

         1、实现int和Integer类之间的转换

在实际转换时，使用Integer类的构造方法和Integer类内部的intValue方法实现这些类型之间的相互转换，实现的代码如下：

                   int n = 10;

                   Integer in = new Integer(100);

                   //将int类型转换为Integer类型

                   Integer in1 = new Integer(n);

                   //将Integer类型的对象转换为int类型

                   int m = in.intValue();

         2、Integer类内部的常用方法

                  在Integer类内部包含了一些和int操作有关的方法，下面介绍一些比较常用的方法：

                   a、parseInt方法

                            public static int parseInt(String s)

该方法的作用是将数字字符串转换为int数值。在以后的界面编程中，将字符串转换为对应的int数字是一种比较常见的操作。使用示例如下：

                                     String s = “123”;

                                     int n = Integer.parseInt(s);

则int变量n的值是123，该方法实际上实现了字符串和int之间的转换，如果字符串都包含的不是都是数字字符，则程序执行将出现异常。(说明：异常的概念将在下一章进行讲述)

另外一个parseInt方法：

         public static int parseInt(String s, int radix)

则实现将字符串按照参数radix指定的进制转换为int，使用示例如下：

         //将字符串”120”按照十进制转换为int，则结果为120

         int n = Integer.parseInt(“120”,10);

         //将字符串”12”按照十六进制转换为int，则结果为18

         int n = Integer.parseInt(“12”,16);

         //将字符串”ff”按照十六进制转换为int，则结果为255

         int n = Integer.parseInt(“ff”,16);

这样可以实现更灵活的转换。

                   b、toString方法

                            public static String toString(int i)

                            该方法的作用是将int类型转换为对应的String类型。

                            使用示例代码如下：

                                     int m = 1000;

                                     String s = Integer.toString(m);

                            则字符串s的值是”1000”。

                            另外一个toString方法则实现将int值转换为特定进制的字符串：

                                     public static int parseInt(String s, int radix)

                            使用示例代码如下：

                                     int m = 20;

                                     String s = Integer.toString(m,16);

                            则字符串s的值是”14”。

         其实，JDK自从1.5(5.0)版本以后，就引入了自动拆装箱的语法，也就是在进行基本数据类型和对应的包装类转换时，系统将自动进行，这将大大方便程序员的代码书写。使用示例代码如下：

                   //int类型会自动转换为Integer类型

                   int m = 12;

                   Integer in = m;

                   //Integer类型会自动转换为int类型

                   int n = in; //int n=(int)in;

         所以在实际使用时的类型转换将变得很简单，系统将自动实现对应的转换。