数据装箱：将基本数据类型保存到对应的包装类中，一般用构造方法完成（但是jdk1.9后不再使用）

数据拆箱：从包装类中获取基本数据类型

数值型包装类：Number类中已经定义了拆箱的方法

Boolean：public boolean booleanValue( )

但是JDK1.9之后，对所有包装类中的构造方法都变成了过期处理

这是因为JDK1.5之后提供了自动的装箱与拆箱操作

Integer obj=10；——自动装箱

int num=obj； ——自动拆箱

obj++； ——包装类对象可以直接参与数学运算

自动装箱和拆箱的好处

1. 如上述例子，可以提供自动的数学运算支持
2. 实现Object接受基本数据类型的操作，如下例子

Object obj=19.2;——double 自动包装为Double，并自动向上转型

obj++； ——不能这么干，因为obj类型为Object，并不支持自动的数学运算

double number=obj；——不能这么干，因为obj类型为Object，无法自动拆箱

double number=(Double)obj;——将obj向下转型为Double后，可以自动拆箱

**包装类的比较**

包装类用来存储数值的属性data，有位数的限制

如果data存储的数据超过了位数的限制——用equals( )来比较

没超过——用”==”