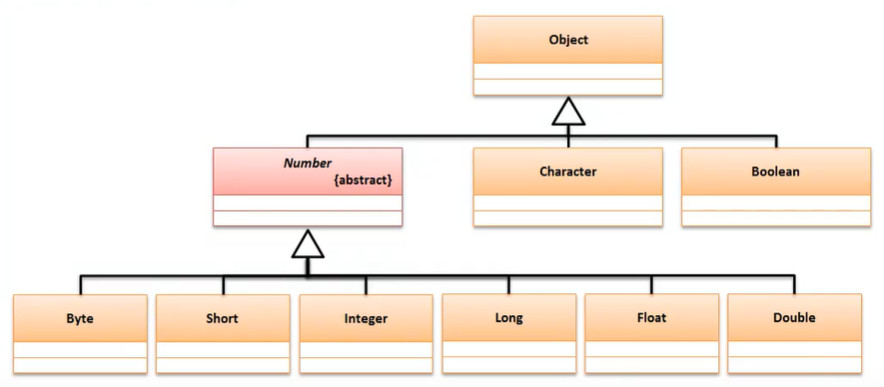
系统自带的包装类：8个。（自动装箱操作需要考虑类型）

Byte、Short、Integer、Long、Float、Double、Character、Boolean。

对象型包装类：Character、Boolean

数值型包装类：Byte、Short、Integer、Long、Float、Double

结构关系：

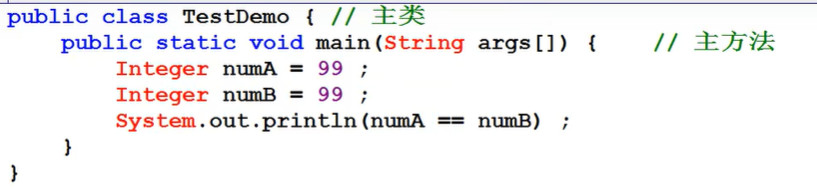




装箱拆箱：这里将基本数据类型与引用类型进行转换。子类与父类或者接口的上转型或者下转型。应当把它作为一种思路。

基本类型与引用类型的自动装箱与拆箱，不用再调用构造方法：Integer i=99;

**重点：**



结果true。

（结论：使用自动装箱与拆箱功能时当Integer的值**直接赋予**在-128-127之间的值时，会将对象进行入池即是放入常量池中进行保存。如果超过了此范围，就会在堆区中建立对象，那么就只能使用equals（）进行比较。**注意，与String不同的是，即使不是直接赋值那么也符合上述的规则**）

自动装拆箱的原理，可以用来验证：

[java包装类 —— 自动装箱与拆箱、常量池内存复用（底层原理）\_demo\_yo的博客-CSDN博客](https://blog.csdn.net/demo_yo/article/details/114993429)

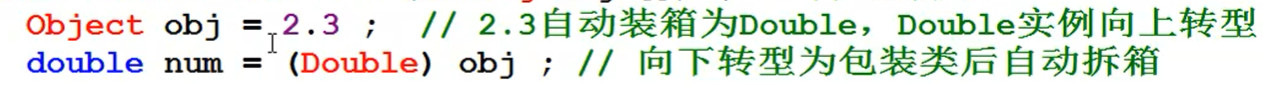
再补充一点：关于常量池，仅仅只是编译的时候进行分配，所以非final仅仅只有引用声明赋值的时候才会存在将值放入常量池的问题，其他的是必然会在堆内存中开辟空间，然后再将值指向，所以只有String只有直接在声明时赋值时才会将栈引用指向常量池，除此之外，如果你不调用intern，则一定会在Heap中进行空间分配然后赋值其中。

（常量池也是堆区中一部分，仅仅是因为常量的定义，不可修改，所以分割出来让其做到一个充分的利用，别想的太复杂）

常量池属于堆区，共享区域，而堆区主要用于容纳对象。

[被final修饰的变量在哪存储\_带着BAT大厂的面试问题：一文理解关键字final\_weixin\_39762464的博客-CSDN博客](https://blog.csdn.net/weixin_39762464/article/details/111129971)

静态终结型变量一定要在定义时进行初始化，不能再静态代码块中进行初始化。



Object从此一同参数界。

