Integer：  
属性：

1. private static class IntegerCache //静态内部类 其中存储了Integer数组cache，在这个类的静态代码块中创建了这个数组实例，大小为256，存储了value为-128-127 的Integer对象。
2. @Native public static final int MAX\_VALUE = 0x7fffffff;
3. @Native public static final int MIN\_VALUE = 0x80000000;
4. @Native public static final int SIZE = 32; //位数
5. public static final int BYTES = SIZE / Byte.SIZE; //字节数
6. private final int value; //被final修饰，不能改变大小，只能改变Integer的指向。

构造方法：

1. 没有无参构造
2. public Integer(int value)
3. public Integer(String s) throws NumberFormatException

成员方法：

1. public int compareTo(Integer anotherInteger) //返回1 -1 0
2. public int intValue() //拆箱时默认调用的方法 还有byte、short、long、float、double
3. public static Integer valueOf(int i) //装箱时默认调用的方法 如果i在-128-127之间，则返回IntegerCache静态内部类中的cache数组中已经new好的对象，如果不在这个范围，则新创建Integer对象返回。
4. public static int parseInt(String s, int radix) //将s转为int型，s是radix进制

要点：  
1. 用运算符比较大小的底层实现，调用intValue()返回基本数据类型然后比较(拆箱)。

用compareTo方法比较大小直接比较对象的value。i > j 等价于 i.intValue() > j.intValue()

1. i + j 等价于 i.intValue() + j.intValue()

i = i + j 等价于 i = Integer.valueOf(i.intValue() + j.intValue())

1. i + 5等价于i.intValue() + 5

i = i + 5 等价于i = Integer.valueOf(i.intValue() + 5)

1. Byte、Short、Integer、Long的cache都是-128-127

Chacter的cache是0-127