1. **包装类**

包装类主要功能是针对基本数据类型的对象转换而实现的，并且随着JDK版本的改变，包装类的功能也在不断的改变，有着更多的支持。

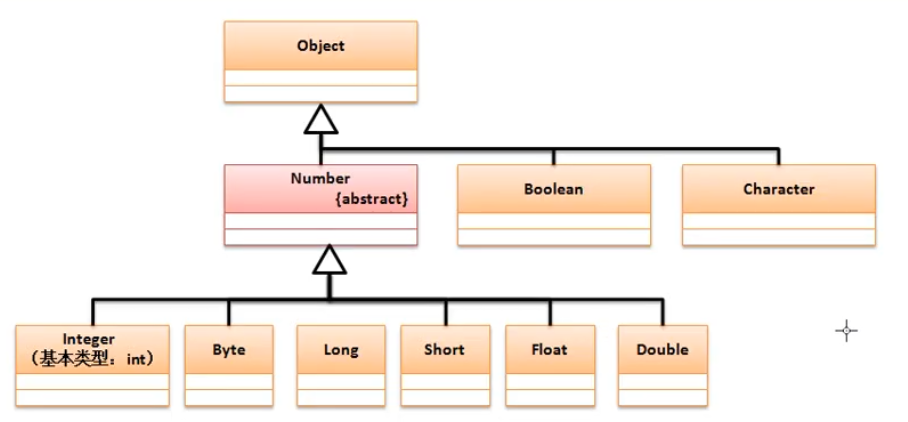
1. . 认识包装类

Object类最大的特点是所有类的父类，并且可以接收所有的数据类型，但是在这个过程中就存在有一个问题:基本数据类型并不是一个类，所以现在如果想要将基本数据类型以类的形式进行处理，那么我们就需要对其进行包装。

范例:以int数据类为例实现一个包装

|  |
| --- |
| package com.company;  /\*\*  \* wraps an int DataType  \*/  class Int{  private int data ;  public Int(int data){  this.data = data ;  }  public int intValue(){  return this.data ;  }  }  public class JavaDemo {  public static void main(String[] args) {  //装箱:将基本数据类型保存在包装类中  Int temp = new Int(10) ;  //拆箱:从包装类中获取基本数据类型；  int x = temp.intValue() ;  }  } |

基本数据类型进行包装处理后可以像对象一样进行引用操作，同时可以使用object来进行接收，所以JDK1.0的时候就提供了包装类的概念。基本数据类型一共有八种，所以提供八种包装类，那么这八种包装类的基本定义如下:



所以可以发现在Java中包装类一共提供有两种类型:

·对象型包装类(Object直接子类):Boolean,Character;

·数值型包装类(Number直接子类):Byte,Short,Integer,Long,Float,Double

Number是一个抽象类：看到Number类都应该明确他是一个数字的概念，这个类定义有如下方法（六个）:

|  |
| --- |
|  |

都是从包装类中获取基本数据类型。

1. 装箱与拆箱

装箱即:将基本数据类型包装成一个类。

·Integer类: public Integer (int value);

·Double类: public Double (double value);

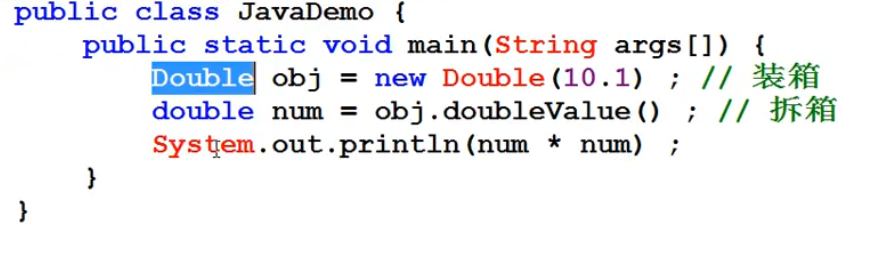
·Boolean类: public Boolean (boolean value)

拆箱:从包装类中获取基本数据类型。

-数值型包装类已经由Number类定义了拆箱方法。

-Boolean: public boolean booleanValue() ;

范例:



我们可以发现从JDK1.9之后，对于所有包装类之中提供的构造方法都变为了过期处理，不建议用户继续使用了。这是因为在JDK1.5之后为了方便提供了自动装箱拆箱处理。

范例；自动装箱与拆箱

|  |
| --- |
|  |

除了数学运算支持以外，使用自动装箱最大的好处是可以实现Object接收基本数据类型操作

范例:Object接收小数

|  |
| --- |
|  |

但是对于包装类需要注意相等判断。

范例:观察相等判断

|  |  |
| --- | --- |
|  |  |

以后进行包装类相等判断的时候一定要使用equals()完成，而包装类本身也要考虑占位长度，如果超过了一位的内容就需要使用equals()如果没有就可以使用 ==