

二、常用类（面试常考）

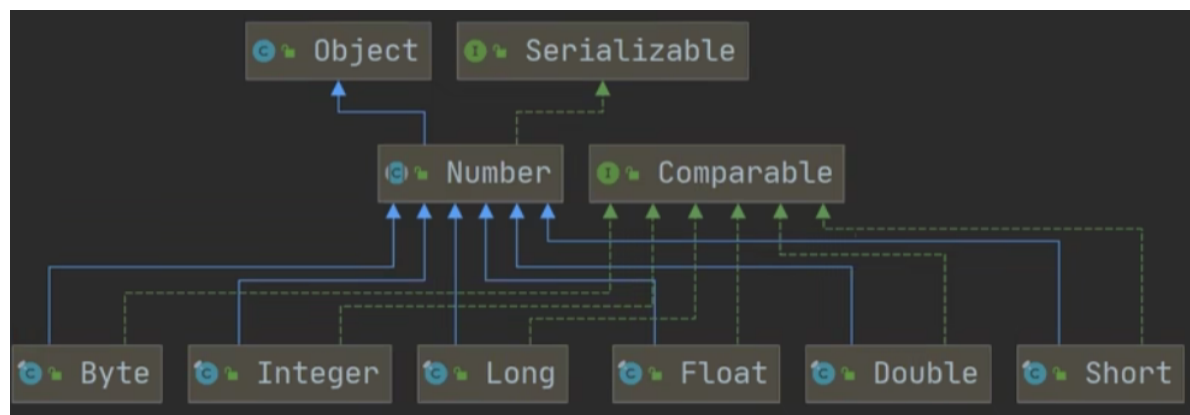
- 重点部分：

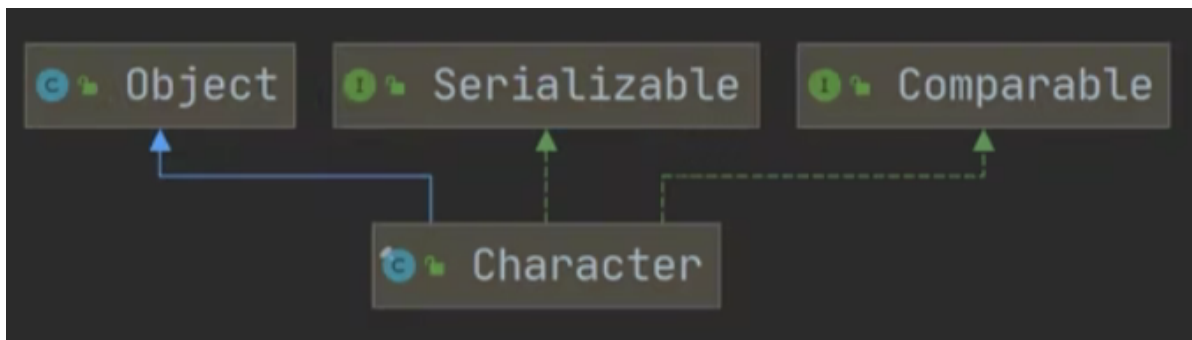
- - String
 - StringBuffer
 - StringBuilder

一、包装类（Wrapper）

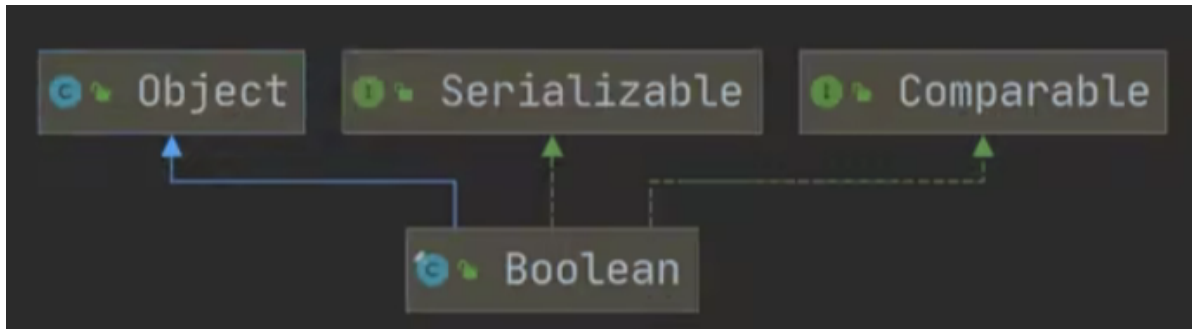
包装类	
<ul style="list-style-type: none">● 包装类的分类 WrapperType.java1. 针对八种基本数据类型相应的引用类型—包装类2. 有了类的特点，就可以调用类中的方法。	
基本数据类型	包装类
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

- 继承关系：





o



o

二、手动装箱，拆箱 & 自动装箱，拆箱

```

package com.wrapper;

public class Integer01 {
    public static void main(String[] args) {
        /*演示int <--> Integer 的装箱 和 拆箱
        * 1.jdk5前是手动装箱和拆箱
        * 2.*/
        //手动装箱
        int n1 = 100;
        Integer integer = new Integer(n1); //装箱方法1
        Integer integer1 = Integer.valueOf(n1); //装箱方法2
        //手动拆箱 Integer --> int
        int i = integer.intValue();

        //jdk5之后,就可以自动装箱和拆箱了
        int n2 = 200;
        Integer integer2 = n2; //底层使用的方法就是Integer.valueOf(n2)
        //简写:
        //Integer n2 = 200;
    }
}
  
```

- 练习:

```
//以下代码是否ok
Double d = 100d;//ok, 底层是Double.valueOf(100d)
Float f = 1.5f;//ok,底层是Float.valueOf(1.5f)
```

经典面试题1:

```
package Excer;

public class Excer01 {
    public static void main(String[] args) {
        /*以下输出结果相同吗?分别是什么*/
        /*三元运算符,?前真取第一个,
        又因为三元算符看作一个整体,
        里头的类型取最大的那个这里最大的类型是的是Double
        所以虽然真取Integer但是最终结果按照Double来衡量
        结果为:1.0*/
        Object obj1 = true ? new Character('a') : new Double(2.0);
        System.out.println(obj1);//Integer的话:1.0,Character的话:97.0
    }
}
```

```
Object obj2;
if(true)
    obj2 = new Integer(1);
else
    obj2 = new Double(2.0);
System.out.println(obj2);//1
输出什么? 1, 分别计算
```

- Excer03:

```
package com.wrapper;

public class wrapperVSString {
    public static void main(String[] args) {
        /*包装类(eg:Integer) --> String
        Integer i = 100;//自动装箱
        //方法1:
        String str1 = i + "";//加一个空串就能将100 --> "100"
        //方法2:
        String st2 = i.toString();
        //方法3:
        String str3 = String.valueOf(i);
        //传进去的i是一个对象(Integer本身就是来自于Object的)

        //String -->包装类
        String str4 = "1234";
        Integer i1 = Integer.parseInt(str4);
        //方法2:
```

```
Integer i2 = new Integer(str4); //Integer构造器里又parseInt转换代码  
}  
}
```

- 包装类常用方法:

包装类

• Integer类和Character类的常用方法

Integer 类和Character 有些常用的方法，我们一起来使用一下。

- ```
System.out.println(Integer.MIN_VALUE); //返回最小值
System.out.println(Integer.MAX_VALUE); //返回最大值
```
- ```
System.out.println(Character.isDigit('a')); //判断是不是数字  
System.out.println(Character.isLetter('a')); //判断是不是字母  
System.out.println(Character.isUpperCase('a')); //判断是不是大写  
System.out.println(Character.isLowerCase('a')); //判断是不是小写
```
- ```
System.out.println(Character.isWhitespace('a')); //判断是不是空格
System.out.println(Character.toUpperCase('a')); //转成大写
System.out.println(Character.toLowerCase('A')); //转成小写
```

经典面试题2:

## ● Integer类面试题1 WrapperExercise02.java

看看下面代码，输出什么结果？为什么？2min

```
public void method1() {
 Integer i = new Integer(1); i 和 j 为不同的对象
 Integer j = new Integer(1); i == j 判断的是对象是否相同所以为Flase
 System.out.println(i == j); //Flase
 //所以，这里主要是看范围 -128 ~ 127 就是直接返回
 Integer m = 1; //底层 Integer.valueOf(1); -> 阅读源码
 Integer n = 1; //底层 Integer.valueOf(1);
 System.out.println(m == n); //T
 //所以，这里主要是看范围 -128 ~ 127 就是直接返回
 //， 否则，就new Integer(xx);
 Integer x = 128; //底层 Integer.valueOf(1);
 Integer y = 128; //底层 Integer.valueOf(1);
 System.out.println(x == y);
}
```

- Integer ? = ? 的底层源码

```
public static Integer valueOf(int i) {
 if (i >= IntegerCache.low && i <= IntegerCache.high)
 return IntegerCache.cache[i + (-IntegerCache.low)];
 return new Integer(i);
}
底层源码
```

### 经典面试题3:

- 只要有基本数据类型，判断时就是看值是否相等



```
//示例六
Integer i11=127;
int i12=127;
//只有有基本数据类型，判断的是
//值是否相同
System.out.println(i11==i12); //T

//示例七
Integer i13=128;
int i14=128;
System.out.println(i13==i14); //F
```

- 对象不同

```
//示例五
Integer i9=127; //Integer.valueOf(127)
Integer i10=new Integer(127);
System.out.println(i9==i10); //F
```

### 三、String类（经典面试内容）



#### String类较常用构造器(其它看手册):

```
String s1 = new String(); //
String s2 = new String(String original);
String s3 = new String(char[] a);
String s4 = new String(char[] a,int startIndex,int count)
```

- 补充: `String s5 = new String (byte[] b)`
- 重要介绍点:

```
public class String01 {
 public static void main(String[] args) {
 /*1."jack",像这样用双引号括起来的是字符串序列
 * 2.String类有很多构造器重写(以上有常用的构造器重写)
 * 3.String 类实现了接口 Serializable (使得String
 * 可以[串行化:可进行网路传输]),
 * 接口Comparable(使得
 * String可以比较大小)
 * 4.String是final类, 不能被其它的类继承
 * implements java.io.Serializable, Comparable<String>, CharSequence {
 /** The value is used for character storage 改制用于字符存储
 * private final char value[]; 用于存放字符串内容的
 * 5.一定要注意value是final类型, 一旦被赋值, 该值的地址
 * 是不能被修改, 但该地址下的值可被修改
 * 这里的value不能指向新地址, 但单个
 * 字符内容是可以变化的.*/
 String name = "jack";
 //5.例子:
 final char[] value = {'a', 'b'};
 char[] v2 = {'f', 'd'};
 value[0] = 'd'; //value数组内单个字符是可以被修改的
 value = v2; //由于final, 不能将value的地址修改成v2的地址
 }
}
```

