Java means DURGA SOFT..

CORE JAVA

Material

India's No.1 Software Training Institute

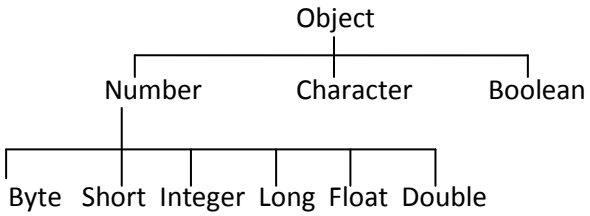
DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Wrapper classes

- *Java is an Object oriented programming language so represent everything in the form of the object, but java supports 8 primitive data types these all are not part of object.*
- *To represent 8 primitive data types in the form of object form we required 8 java classes these classes are called wrapper classes.*
- *All wrapper classes present in the **java.lang** package and these all classes are **immutable** classes.*

Wrapper classes hierarchy:-



Wrapper classes constructors:-

```
Integer i = new Integer(10);
Integer i1 = new Integer("100");
Float f1= new Float(10.5);
Float f1= new Float(10.5f);
Float f1= new Float("10.5");
Character ch = new Character('a');
```

datatypes	***wrapper-class***	**constructors**
byte	*Byte*	*byte,String*
short	*Short*	*short,String*
int	*Integer*	*int,String*
long	*Long*	*long,String*
float	*Float*	*double,float,String*
double	*Double*	*double,String*
char	*Character*	*char*
boolean	*Boolean*	*boolean,String*

***Note :-** To create wrapper objects all most all wrapper classes contain two constructors but Float contains three constructors(float,double,String) & char contains one constructor(char).*

toString():-

- *toString() method present in Object class it returns class-name@hashcode.*
- *String,StringBuffer classes are overriding toString() method it returns content of the objects.*
- *All wrapper classes overriding toString() method to return content of the wrapper class objects.*

Example :-

```
class Test
{       public static void main(String[] args)
        {       Integer i1 = new Integer(100);
                System.out.println(i1);
                System.out.println(i1.toString());

                Integer i2 = new Integer("1000");
                System.out.println(i2);
                System.out.println(i2.toString());

                Integer i3 = new Integer("ten");//java.lang.NumberFormatException
                System.out.println(i3);
        }
}
```

*In above example for the integer constructor we are passing **"1000"** value in the form of String it is automatically converted into Integer format.*
*In above example for the integer constructor we are passing **"ten"**in the form of String but this String is unable to convert into integer format it generate exception **java.lang.NumberFormatException.***

Example:-conversion of wrapper to String by using toString() method

```
class Test
{       public static void main(String[] args)
        {       Integer i1 = new Integer(100);
                Integer i2 = new Integer("1000");
                System.out.println(i1+i2);//1100
                //conversion [wrapper object - String]
                String str1 = i1.toString();
                String str2 = i2.toString();
                System.out.println(str1+str2);//1001000
        }
}
```

Example:-

- *In java we are able to call toString() method only on reference type but not primitive type.*
- *If we are calling toString() method on primitive type then compiler generate error message.*

```
class Test
```

```
{       public static void main(String[] args)
        {       Integer i1 = Integer.valueOf(100);
                System.out.println(i1);
                System.out.println(i1.toString());

                int a=100;
                System.out.println(a);
                //System.out.println(a.toString()); error:-int cannot be dereferenced
        }
}
```

valueOf():-

in java we are able to create wrapper object in two ways.

a) By using constructor approach

b) By using valueOf() method

- ✓ valueOf() method is used to create wrapper object just it is alternate to constructor approach and it a static method present in wrapper classes.

Example:-

```
class Test
{       public static void main(String[] args)
        {       //constructor approach to create wrapper object
                Integer i1 = new Integer(100);
                System.out.println(i1);

                Integer i2 = new Integer("100");
                System.out.println(i2);

                //valueOf() method to create Wrapper object
                Integer a1 = Integer.valueOf(10);
                System.out.println(a1);

                Integer a2 = Integer.valueOf("1000");
                System.out.println(a2);
        }
}
```


Example :-conversion of primitive to String.

```
class Test
{       public static void main(String[] args)
        {       int a=100;
                int b=200;
                System.out.println(a+b);

                //primitive to String object
                String str1 = String.valueOf(a);
                String str2 = String.valueOf(b);
                System.out.println(str1+str2);
        }
}
```


XxxValue():- *it is used to convert wrapper object into correspondingprimitive value.*

Example:-

```
class Test
{       public static void main(String[] args)
        {       //valueOf() method to create Wrapper object
                Integer a1 = Integer.valueOf(10);
                System.out.println(a1);

                Integer a2 = Integer.valueOf("1000");
                System.out.println(a2);

                //xxxValue()   [wrapper object into primitive value]
                int x1 = a1.intValue();
                byte x2 = a1.byteValue();
                double x3 = a1.doubleValue();
                System.out.println("int value="+x1);
                System.out.println("byte value="+x2);
                System.out.println("double value="+x3);
```

```
        }
}
```

parseXXX():- *it is used to convert String into corresponding primitive value& it is a static method present in wrapper classes.*

Example :-

```
class Test
{       public static void main(String[] args)
        {       String str1="100";
                String str2="100";
                System.out.println(str1+str2);
                //parseXXX()  converion of String to primitive type
                int a1 = Integer.parseInt(str1);
                float a2 = Float.parseFloat(str2);
                System.out.println(a1+a2);
        }
}
```

1) ***primitive ----->Wrapper Object***
   Integer i = Integer.valueOf(100);
2) ***wrapper object ----> primitive***
   byte b = i.byteValue();
3) ***String value ----> primitive***
   String str="100";
   int a = Integer.parseInt(str);
4) ***primitive valu ----> String Object***
   int a=100;
   int b=200;
   String s1 = String.valueOf(a);
   String s2 = String.valueOf(b);
   System.out.println(s1+s2);//100200
5) ***String value ---->Wrapper object***
   Integer i =Integer.valueOf("1000");
6) ***wrapper object --->String object***
   Integer i = new Integer(1000);
   String s = i.toString();

Autoboxing and Autounboxing:-(introduced in the 1.5 version)

- *Up to 1.4 version to convert primitive/String into Wrapper object we are having two approaches*
  - ***Constructor approach***
  - ***valueOf() method***
- *Automatic conversion of primitive to wrapper object is called autoboxing.*
- *Automatic conversion of wrapper object to primitive is called autounboxing.*

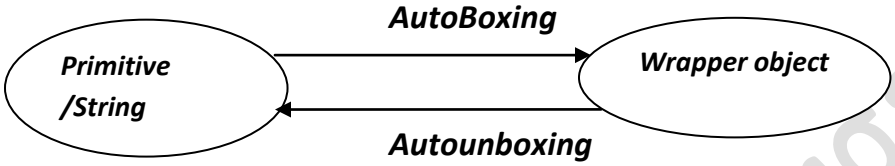
Example:-

```
class Test
{       public static void main(String[] args)
```

```
        {       //autoboxing    [primitive - wrapper object]
                Integer i = 100;
                System.out.println(i);
                System.out.println(i.toString());

                //autounboxing [wrapper object - primitive]
                int a = new Integer(100);
                System.out.println(a);
        }
}
```

Automatic conversion of the primitive to wrapper and wrapper to the primitive:-



Factory method:-

- One java class method returns same class object or different class object is called factory method.
- There are three types of factory methods in java.
  - ***Instance factory method.***
  - ***Static factory method.***
  - ***Pattern factory method.***
- The factory is called by using class name is called static factory method.
- The factory is called by using reference variable is called instance factory method.
- One java class method is returning different class object is called pattern factory method.

Example:-

```
class Test
{       public static void main(String[] args)
        {       //static factory method
                Integer i = Integer.valueOf(100);
                System.out.println(i);
```

```
        Runtime r = Runtime.getRuntime();
        System.out.println(r);

        //instance factory method
        String str="ratan";
        String str1 = str.concat("soft");
        System.out.println(str1);

        String s1="sravyainfotech";
        String s2 = s1.substring(0,6);
        System.out.println(s2);

        //pattren factory method
        Integer a1 = Integer.valueOf(100);
        String ss = a1.toString();
        System.out.println(ss);

        StringBuffersb = new StringBuffer("ratan");
        String sss = sb.toString();
        System.out.println(sss);
    }
}
```