Wrapper Classes

The Main Objectives of Wrapper Classes are:

- To Wrap Primitives into Object Form. So that we can handle Primitives also Just Like Objects.
- To Define Several Utility Methods for the Primitives (Like converting Primitive to the String Form).

Constructors:

- All Most All Wrapper Classes defines 2 Constructors.
- One can take corresponding Primitive as an Argument and the Other can take String as an Argument.

```
Integer I = new Integer(10);
Integer I = new Integer("10");
Double D = new Double(10.5);
Double D = new Double("10.5");
```

- If the String Argument is not representing Number then we will get RE: NumberFormatException.
  Eg: Integer I = new Integer("Ten");
  //RE: java.lang.NumberFormatException: For input string: "Ten"

- Float Class defines 3 Constructors with float, double and String Arguments.

```
Float F = new Float(10.5f); √
Float F = new Float("10.5f"); √
Float F = new Float(10.5); √
Float F = new Float("10.5"); √
```

- Character Class contains only one Constructor with *char* Primitive as an Argument Type.

```
Character ch = new Character('a'); √
Character ch = new Character("a"); ×
```

- Boolean Class contains 2 Constructors with boolean Primitive and String Arguments.
- If we Pass *boolean* Primitive as an Argument the only allowed Values are *true* OR *false*, Where Case and Content Both are Important.

```
Boolean B = new Boolean(true); √
Boolean B = new Boolean(false); √
Boolean B = new Boolean(True); ×
Boolean B = new Boolean(TRUE); ×
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

- If we are passing String Argument then Case and Content Both are Not Important.
- If the Content is Case Insensitive String(true) then it is treated as true. Otherwise it is treated as false.

```
Boolean B = new Boolean("true"); ➜ true
Boolean B = new Boolean("True"); ➜ true

Boolean B = new Boolean("TRUE"); ➜ true
Boolean B = new Boolean("false"); ➜ false

Boolean B = new Boolean("Malaika"); ➜ false
Boolean B = new Boolean("Mallika"); ➜ false
Boolean B = new Boolean("Jareena"); ➜ false
```

```
Boolean X = new Boolean("Yes");
Boolean Y = new Boolean("No");

System.out.println(X); //false
System.out.println(Y); //false

System.out.println(X.equals(Y)); //true
System.out.println(X == y); //false
```

Wrapper Class	Corresponding Constructor Arguments
Byte	byte OR String
Short	short OR String
Integer	int OR String
Long	long OR String
Float	float OR String OR double
Double	double OR String
Character	char
Boolean	Boolean OR String

Note:

- In every Wrapper Class toString() is Overridden for Meaningful String Representation.
- In every Wrapper Class equals() is Overridden for Content Comparision.

Utility Methods

- valueOf()
- xxxValue()
- parseXxx()
- toString()

1) **valueOf():** We can use valueOf() to Create Wrapper Object as Alternative to Constructor.

- **Form 1:** All the Wrapper Classes except Character Class contains a Static valueOf() to Create Wrapper Object for the given String.

public static wrapper valueOf(String s);

```
Integer I = Integer.valueOf('10');
Float F = Float.valueOf("10.5");
Boolean B = Boolean.valueOf("Durga");
```

- **Form 2:** All Integral Wrapper Classes (Byte, Short, Integer, Long) Contains the following valueOf() to Create Wrapper Object for the given specified Radix String.

public static wrapper valueOf(String s, int radix);

- The allowed Range of Radix is 2 to 36.
- Because Numerics(10), Alphabets(26) Finally 10 + 26 =36.

```
Integer I = Integer.valueOf("100", 2);
System.out.println(I); //4
```

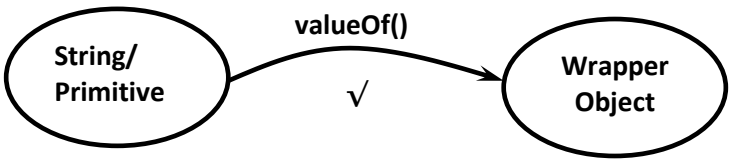
```
Integer I = Integer.valueOf("101", 4);
System.out.println(I); //17
```

- **Form 3:** Every Wrapper Class including Character Class contains the following valueOf() to Convert *Primitive* to *Wrapper* Object Form.

public static wrapper valueOf(primitive p);

```
Integer I = Integer.valueOf(10);
Character ch = Character.valueOf('a');
Boolean B = Boolean.valueOf(true);
```

Version1, Version2 ➔ String to Wrapper Object.
Version3 ➔ Primitive to Wrapper Object.



2) **XxxValue():**

- We can use XxxValue() to find Primitive Value for the given Wrapper Object.
- Every Number Type Wrapper Class [Byte, Short, Integer, Long, Float and Double] contains the following XxxValue() to find Primitive Value for the given Wrapper Object.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Examples

```
public byte byteValue();
public short shortValue();
public int intValue();
public long longValue();
public float floatValue();
public double doubleValue();
```

```
Integer I = Integer.valueOf(130); OR Integer I = new Integer(130);
System.out.println(I.byteValue()); //-126
System.out.println(I.shortValue()); //130
System.out.println(I.intValue()); //130
System.out.println(I.longValue()); //130
System.out.println(I.floatValue()); //130.0
System.out.println(I.doubleValue()); //130.0
```

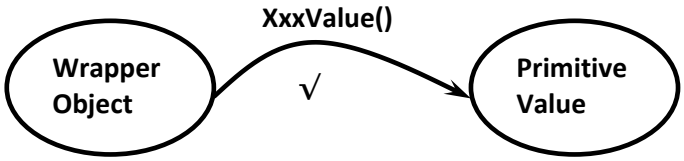
charValue(): Character Class contains charValue() to find char Primitive for the given kind of Character Object. *public char charValue()*

```
Character ch = new Character('a');
char ch1 = ch.charValue();
System.out.println(ch1); //a
```

booleanValue(): Boolean Class contains booleanValue() to Return boolean Primitive for the given Boolean Object. *public boolean booleanValue();*

```
Boolean B = Boolean.valueOf("Durga");
boolean b = B.booleanValue();
System.out.println(b); //false
```

Note: In Total there are 38 ((6 X 6) +1 + 1) xxxValue() Methods are Available.



3) parseXxx(): We can use parseXxx() to Convert String to Primitive.

- Form 1: Every Wrapper Class except Character Class contains the following parseXxx() for converting String to Primitive Type.
  public static primitive parseXxx(String s);

  Eg:
  ```
  int i = Integer.parseInt("10");
  double d = Double.parseDouble("10.5");
  boolean b = Boolean.parseBoolean("true");
  ```

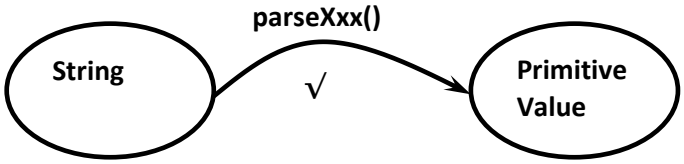
- Form 2: Every Integral Type Wrapper Class contains the following parseXxx() to Convert specified Radix String Form into Primitive.
  public static primitive parseXxx(String s, int radix);

  The allowed Range of Radix is 2 To 36.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Eg:

```
int i = Integer.parseInt("110", 2);
System.out.println(i); //6
```



4) toString();

- **Form 1:**
  - All Wrapper Classes contains an Instance Method toString() for converting Wrapper Object to String Type.
  - This is Overriding Method of Object Class toString().
  - Whenever we are trying to Print Wrapper Object Reference internally this toString() will be Call. *public String toString();*

```
Integer I = new Integer(10);
System.out.println(I); //10 → (I.toString());

String s = I.toString();
System.out.println(s); //10
```

- **Form 2:** Every Wrapper Class including Character Class defines the following Static toString() for converting Primitive to String Representation.
  public static String toString(primitive p);

```
String s = Integer.toString(10);

String s = Boolean.toString(true);

String s3 = Character.toString('a');
```

- **Form 3:** Integer and Long Classes contains the following toString() to Convert Primitive to specified Radix String Form.
  public static String toString(primitive p, int radix);

  The allowed Range of Radix: 2 - 36

Eg:

```
String s = Integer.toString(7, 2);
System.out.println(s); //111
```

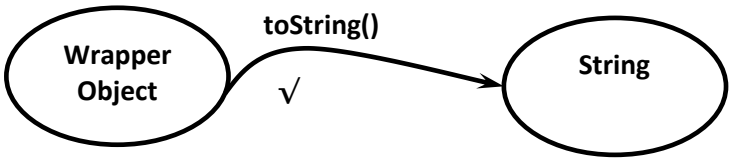
➢ **Form 4:** toXxxString()

Integer and Long Classes contains the following toXxxString() Methods to Return specified Radix String Form.

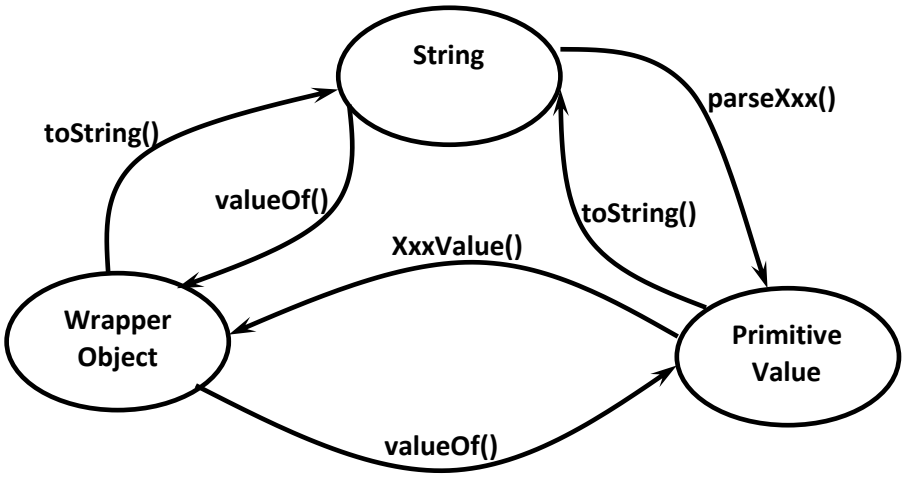
```
public static String toBinaryString(primitive p);

public static String toOctalString(primitive p);

public static String toHexString(primitive p);
```

Examples:

```
String s = Integer.toBinaryString(10);
System.out.println(s); //1010

String s = Integer.toOctalString(10);
System.out.println(s); //12

String s = Integer.toHexString(10);
System.out.println(s); //a
```



Dancing between Wrapper Object, Primitive and String



DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Practice Questions for Wrapper Classes

Q1. Given the code fragment:

```
1)  class Test
2)  {
3)     public static void main(String[] args)
4)     {
5)        Short s1=200;
6)        Integer s2=400;
7)        Long s3=(long)s1+s2;//Line-1
8)        String s4=(String)(s3*s2);//Line-2
9)        System.out.println("Sum is:"+s4);
10)    }
11) }
```

What is the result?

A. Sum is: 600
B. Compilation Fails at Line-1
C. Compilation Fails at Line-2
D. ClassCastException is thrown at Line-1
E. ClassCastException is thrown at Line-2

Answer: C

Q2. Consider the code:

```
1)  public class Test
2)  {
3)     public static void main(String[] args)
4)     {
5)        Boolean[] b = new Boolean[2];
6)        b[0]=new Boolean(Boolean.parseBoolean("true"));
7)        b[1]= new Boolean(null);
8)        System.out.println(b[0]+".."+b[1]);
9)     }
10) }
```

What is the result?

A. true..false
B. true..null
C. Compilation Fails
D. NullPointerException is thrown at runtime

Answer: A

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Q3. Given:

```
1)  public class Test
2)  {
3)     public static void main(String[] args)
4)     {
5)        boolean a = new Boolean(Boolean.valueOf(args[0]));
6)        boolean b = new Boolean(args[1]);
7)        System.out.println(a+".."+b);
8)     }
9)  }
```

And given the commands:
javac Test.java
java Test TRUE null

What is the result?

A. true..null
B. true..false
C. false..false
D. true..true

Answer: B

Q4. Given the code

```
1)  public class Test
2)  {
3)     public static void main(String[] args)
4)     {
5)        String s1="123";
6)        String s2="TRUE";
7)        Integer i1=Integer.parseInt(s1);
8)        Boolean b1= Boolean.parseBoolean(s2);
9)        System.out.println(i1+".."+b1);
10)
11)       int i2= Integer.valueOf(s1);
12)       boolean b2=Boolean.valueOf(s2);
13)       System.out.println(i2+".."+b2);
14)    }
15) }
```

What is the result?

A) 123..true
   123..true

B) 123..true
   123..false

C) 123..false
   123..true

D) Compilation Fails

Answer: A

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com