Pg1

Monday, June 1, 2020 7:56 AM

Wrapper Classes

Wrapper class helps the programmer to store the primitive literals in the form of an object.

All the primitive datatypes in java have a corresponding wrapper class to achieve the task.

Therefore, with the help of wrapper class we can convert a primitive type into a non-primitive type and vice-versa.

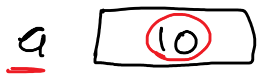
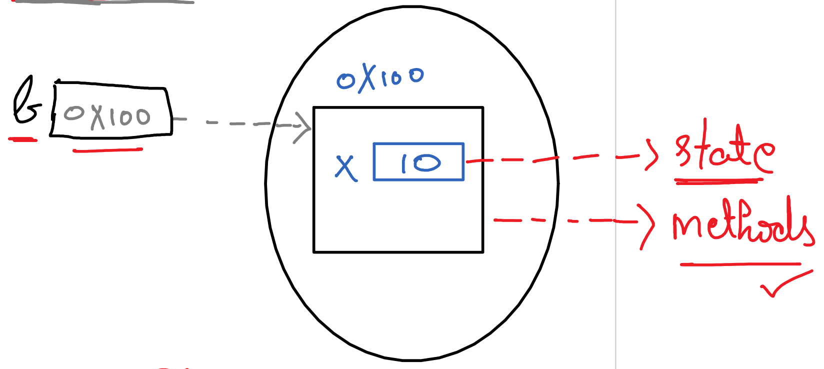
All these wrapper classes are available in java.lang package. Therefore, we can use them without importing.

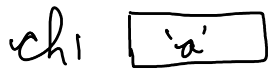
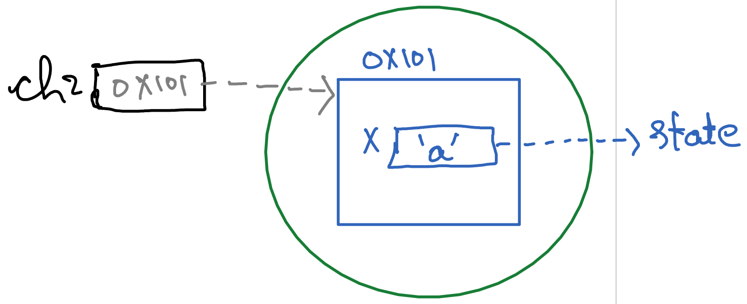
List Of wrapper classes :

primitive type	**Non-primitive type**
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean
-	String

example :

int a = 10 ;	Integer b = 10 ;

int a = 10 ;  a.toString() X	Integer b = 10 ;  b.toString() ✓

char ch1 = 'a' ;

Sop (ch1) ; // a
S.o.p (ch1.hashCode ()); X
CTE
	Character ch2 = 'a' ;
	
	Sop(ch2); // a
	Sop (ch2.hashCode()); ✓

Note :

1. In all wrapper classes, the object class methods such as toString(), equals() and hashCode() is overridden.
2. All wrapper classes implement the following interfaces :
    1. Comparable ✓
    2. Serializable ✓
3. Wrapper classes helps us to convert primitive type to Non-primitive type (Boxing)
4. Wrapper classes helps us to convert a non-primitive into a primitive type (Un Boxing)

Boxing :

The process of converting, a primitive type into a non-primitive type is known as Boxing.

From jdk 1.5 onwards, boxing is done implicitly, hence it is also known as auto-boxing.

Note :

A programmer can explicitly perform Boxing with the help of static method valueOf() present in all the Wrapper classes

Integer a = Integer.valueOf(10) ; // explicit Boxing

Boxing :

primitive	Non-primitive	Boxing
byte	Byte	Byte Byte.valueOf(byte)
short	Short	Short Short.valueOf(short)
int	Integer	Integer Integer.valueOf(int)
long		
float		
double		
char		
boolean		

UnBoxing :

The process of converting, non-primitive type into a primitive type is known as unboxing.

From jdk 1.5 onwards, the process of unboxing is done implicitly by the compiler, hence it is also known as auto unboxing.

Example :

Integer a = 10 ; // auto boxing

int b = a ; // Integer ---> int auto unboxing

int Integer

A programmer can perform Unboxing explicitly with the help a non-static method present in the respective wrapper class:

Primitive	**Non-Primitive**	**Un Boxing** (non-static method)
byte	Byte	byte byteValue()
short	Short	short shortValue()
int	Integer	int intValue()
long	Long	long longValue()
double	Double	double doubleValue()
float	Float	float floatValue()
boolean	Boolean	boolean booleanValue()
char	Character	char charValue()

Note :

1. As we can convert a primitive type in java into a non-primitive type, and Object class being the super most class for all the non-primitive type. It is easy to conclude that we can store anything (i.e. either primitive or non-primitive values) in Object type reference variable.

   Example refer : **workspace/wrapper/src/pack1/W8.java**

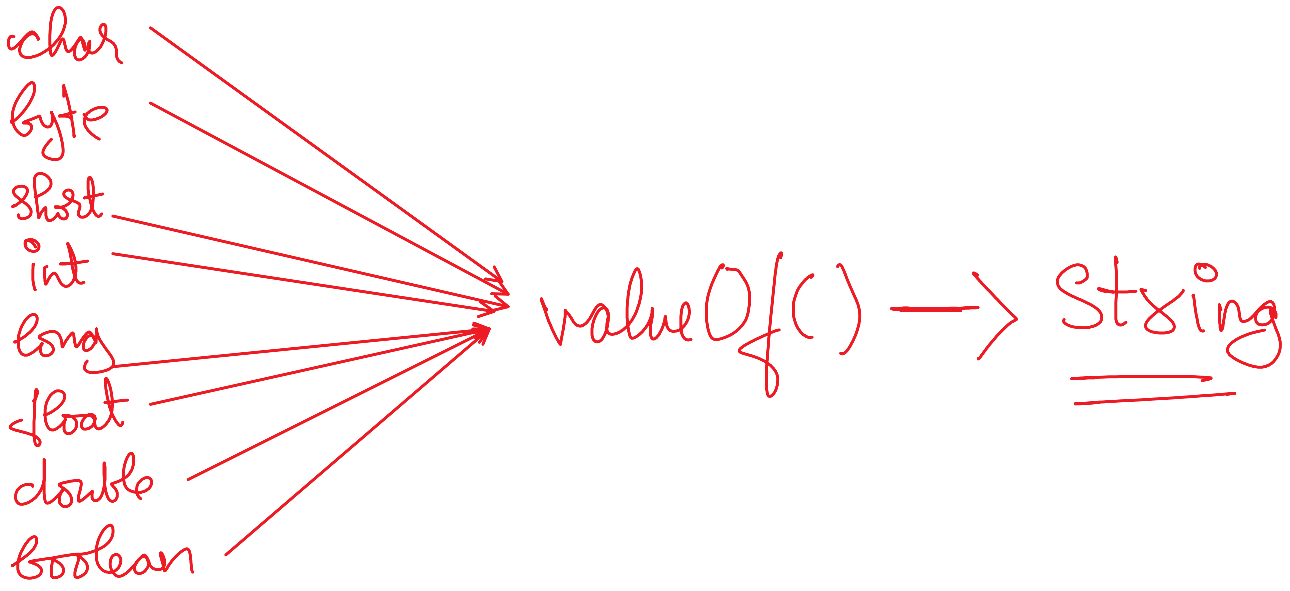
Pg2

Tuesday, June 2, 2020 8:38 AM

In java, all primitive types can be converted to a String type, and a String type can be converted back to their respective primitive types. This process is known as parsing.

1. **<u>Primitive type to String :</u>**

In String class we have a static valueOf() method overloaded, such that it can accept any primitive type and returns a String.



example refer :

2. **<u>String to respective Primitive Type :</u>**

We can convert String to respective primitive type, with the help of static parse method present in every respective Wrapper class.

String	Primitive Type	method used
String	byte	Byte.parseByte(String)
String	short	Short.parseShort(String)

String	int	Integer.pasrseInt(String)
String	long	Long.parseLong(String)
String	char	-----
String	float	Float.parseFloat(String)
String	double	Double.parseDouble(String)
String	boolean	Boolean.parseBoolean(String)

<u>NumberFormatException :</u>

When we try to parse a String into a number, and the given string contains any character other than a digit, we get NumberFormatException.

Example :

Boolean.parseBoolean(String): It returns true, if the string argument is not null and is equal (ignoring case) to the string "true".
It returns false for null for any other argument other than true.
byte b= Byte.parseByte("2.45")-------> NumberFormatException