Java means DURGA SOFT..

CORE JAVA

Material

Java™

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Wrapper Classes:

- Collection is an object,it able to store a group of some other Objects.
- In Java all the Collection Objects are represented in the form of some predefined library in "java.util" package.
- In java applications,Collection objects never be stored data directly,Collection objects are able to store data in the form of objects only.
- If we want to store data in Collection objects,first we have to convert primitive data into object type then we have to store object type in Collection object.
- If we want to retrieve data from Collection object,first we have to retrieve the respective object type from Collection then we have to convert data from Object type to the respective primitive data type.
- In the above context,to convert the primitive data into its Object form and to convert the data from Object form to the respective Primitive data type we have to use a set of predefined classes called as "Wrapper Classes".

-----Diagram----(CollectionObject.png)

- Java has provided all the wrapper classes w.r.t the primitive data types as part of "java.lang" package.

-------Diagram----(WrapperClassesDiagram.png)

- Java has provided all the wrapper classes as immutable classes,whose objects will not allow modifications on their content.

1.If we want to convert the data from primitive type to its object type or wrapper type then we have to use the following approaches.

a)Using Wrapper classes constructors:

To convert the data from primitive type to its respective wrapper type we have to use the following constructors from each and every wrapper class.

public XXX(xxx var)

Where XXX may be Byte,Short,Integer.....

Where xxx may be byte,short,int.....

public Byte(byte b)

public Short(short s)

public Integer(int i)

Ex:

```
int i=10;
Integer in=new Integer(i);
System.out.println(i+" "+in);
```

O/P: 10 10

b)Using valueOf(--) method:

To convert the data from primitive data type to its object type then each and every wrapper class has provided the following method.

public static XXX valueOf(xxx var)

Ex:

public static Byte valueOf(byte b)

public static Short valueOf(short s)

public static Integer valueOf(int i)

Ex:

```
int i=10;
Integer in=Integer.valueOf(i);
System.out.println(i+"  "+in);
```

O/P: 10 10

c)Using Auto-Boxing mechanism:

To convert the data from primitive type to its corresponding Object type,JDK5.0 version has given a direct mechanism like to assign primitive variable to the respective wrapper type variable called as "Auto-Boxing".

Ex:

```
int i=10;
Integer in=i;
System.out.println(i+"  "+in);
```

O/P: 10 10

2)If we want to convert the data from Wrapper type to primitive type then we have to use the following approaches:

a)Using xxxValue() method:

To convert the data from wrapper type to its respective primitive data type,Each and every wrapper class has provided the following method.

public xxx xxxValue()

Where xxx may be byte,short,int,.....

public byte byteValue()

public short shortValue()

publicintintValue()

Ex:

```
Integer in=new Integer(10);

int i=in.intValue();

System.out.println(in+"  "+i);
```

O/P: 10 10

b)Using Auto-Unboxing Mechanism:

To convert the data from Wrapper type to primitive type,JDK5.0 version has provided a direct mechanism like to assign wrapper type variable to the respective primitive type variable called as "Auto-Unboxing".

Ex:

```
Integer in=new Integer(10);

int i=in;

System.out.println(in+" "+i);
```

O/P:10 10

3)If we want to convert the data from String data type to wrapper type then we have to use the following approaches

a)Using Constructor from Wrapper Classes:

To convert the data from String type to the respective wrapper type,all the wrapper classes have provided the following constructors.

```
public XXX(String data)

public Byte(String data)

public Short(String data)

public Integer(String data)
```

Ex:

```
String data="10";

Integer in=new Integer(data);

System.out.println(data+"  "+in);
```

O/P: 10 10

b)Using valueOf(---) method:

To convert the data from String data type to the respective wrapper type,all most all the wrapper classes have provided the following method.

public static XXX valueOf(String data)

public static Byte valueOf(String data)

public static Short valueOf(String data)

public static Integer valueOf(String data)

Ex:

```
String data="10";
Integer in=Integer.valueOf(data);
System.out.println(data+" "+in);
```

O/P: 10 10

4)If we want to convert the data from Wrapper type to String type then we have to use the following approaches:

a)Using toString() method:

To convert the data from wrapper type to String data type,allmost all the wrapper classes have provided the following method.

public String toString()

Ex:

```
Integer in=new Integer(10);

String data=in.toString();

System.out.println(in+"  "+data);
```

O/P: 10 10

b)Using '+' operator:

In Java,if we concatinate any reference variable with "" then internally JVM will access toString() method then the over all resultant value is becoming as String.

Ex:

```
Integer in=new Integer(10);
```

String data=""+in;

System.out.println(in+" "+data);

O/P: 10 10

5)If we want to convert the data from String data type to the respective primitive data type then we have to use the following approaches:

a)Using parseXXX() method:

To convert the data from String data type to the respective primitive type all most all the wrapper classes have provided the following method.

public static xxx parseXXX(String data)

Where xxx may be byte,short,int.....

public static byte parseByte(String data)

public static short parseShort(String data)

public static intparseInt(String data)

Ex:

String data="10";

int i=Integer.parseInt(data);

System.out.println(data+" "+i);

6)If we want to convert the data from primitive data type to String data type then we have to use the following approaches:

a)Using toString(--) method:

To convert the data from primitive data type to String data type,allmost all the wrapper classes have provided the following method.

public static String toString(xxx value)

Where xxx may be byte,short,int,.....

Ex:

```
int i=10;
String data=Integer.toString(i);
System.out.println(i+"  "+data);
```

O/P: 10 10

b)Using '+' operator:

If we concatenate any primitive variable with "" then that primitive variable value is becoming as String value.

Ex:

```
int i=10;
String data=""+i;
System.out.println(i+"    "+data);
```

O/P: 10 10


LEARN FROM EXPERTS ...
COMPLETE JAVA
CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...
COMPLETE .NET
C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF
TESTING TOOLS
MANUAL + SELENIUM
ORACLE
D2K
MSBI
SHARE POINT
HADOOP
ANDROID
C, C++, DS, UNIX
CRT & APTITUDE TRAINING
AN ISO 9001:2008 CERTIFIED
DURGA
Software Solutions®
202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,
9246212143, 8096969696
www.durgasoft.com