JAVA ASSIGNMENT
ON
WRAPPER CLASSES
ARRAYS,MATH

Name : J.Divya

Assignment

Wrapper Classes

- The Wrapper classes provides mechanism to convert primitive into object and object into primitive.
- For every primitive type we are having Wrapper class.
- we can wrap a primitive value into a wrapper class object.
- Wrapper classes are mainly used in collections because there we don’t have any primitive types .

In Java we are having Autoboxing and unboxing features which will convert primitives into objects and objects into primitives.

Autoboxing:

The automatic conversion of primitive types to the object of their corresponding wrapper classes is known as autoboxing.

For example : conversion of int to Integer, long to Long, double to Double, etc.

Unboxing:

Automatically converting an object of a wrapper class to its corresponding primitive type is known as unboxing.

For example – conversion of Integer to int, Long to long, Double to double, etc.

Primitive Datatypes and their respective Wrapper Classes:

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean

Program for Autoboxing and unboxing:

```
Class Demo{
    public static void main(String[] args)
    {
        byte a = 1;
        Byte byteobj = new Byte(a);
        int b = 10;
        Integer intobj = new Integer(b);
        float c = 18.6f;
        Float floatobj = new Float(c);
        double d = 250.5;
        Double doubleobj = new Double(d);
        char e = 'a';
        Character charobj = e;
        // printing the values from objects

        System.out.println("Byte object byteobj: "
                           + byteobj);
        System.out.println("\nInteger object intobj: "
                           + intobj);
        System.out.println("\nFloat object floatobj: "
                           + floatobj);
        System.out.println("\nDouble object doubleobj: "
                           + doubleobj);
```

```
        System.out.println("\nCharacter object charobj: "
                           + charobj);
       //unwrapping objects to primitive data  types
        byte bv = byteobj;
        int iv = intobj;
        float fv = floatobj;
        double dv = doubleobj;
        char cv = charobj;
       // printing the values

        System.out.println("byte value, bv: " + bv);
        System.out.println("\nint value, iv: " + iv);
        System.out.println("\nfloat value, fv: " + fv);
        System.out.println("\ndouble value, dv: " + dv);
        System.out.println("\nchar value, cv: " + cv);
    }
}
```

Character Wrapper Class:

The Character class wraps a value of the primitive datatype char. It offers a number of useful class (i.e., static) methods for manipulating characters. You can create a Character object with the Character constructor.

Methods in Character Wrapper class:

1. **isLetter()**

Determines whether the specified char value is a letter.

2. **isDigit()**

Determines whether the specified char value is a digit.

3. **isWhitespace()**

Determines whether the specified char value is white space.

4. **isUpperCase()**

Determines whether the specified char value is uppercase.

5. **toUpperCase()**

Returns the uppercase form of the specified char value.

6. **toLowerCase()**

Returns the lowercase form of the specified char value.

7. **toString()**

Returns a String object representing the specified character value that is, a one-character string.

Methods in Java Array Class:

The Arrays class of the java.util package contains several static methods that can be used to fill, sort, search, etc in arrays.

1.**compare(array 1, array 2):**

Compares two arrays passed as parameters lexicographically.

2. asList():

Returns a fixed-size list backed by the specified Arrays.

3. sort(array):

Sorts the complete array in ascending order.

4 .sort(array,fromindex,endindex):

Sorts the specified range of array in ascending order.

5.toString() : It returns a string representation of the contents of this array. The string representation consists of a list of the array's elements, enclosed in square brackets ("[]").

Java Math Class:

Java Math class provides several methods to work on math calculations like min(), max(), avg(), sin(), cos(), tan(), round(), ceil(), floor(), abs() etc.

Basic Methods in Java Math Class:

1) Math.pow: It returns the value of first argument raised to the power to second argument.

2) Math.sqrt: It is used to return the square root of a number.

3) Math.min: It is used to return the Smallest of two values.

4) Math.Max: It return largest of two values.

5) Math.log: It returns the natural logarithm of a double value.

6) Math.cbrt:It is used to return the cube root of a number.

7) Math.round:It is used to round the numbers to the nearest decimal values.

8) Math.abs:It will return the absolute value of the given value.