WRAPPER CLASS ASSIGNMENT

Submited by

Harish Kumbhar

Date: 19 July 2023

1. WRAPPER CLASSES IN JAVA

The wrapper class in Java provides the mechanism to convert primitive into object and object into primitive.

Or.

Wrapper classes provide a way to use primitive data types (**int, boolean, etc.**.) as objects.

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

AUTOBOXING

The automatic conversion of primitive data type into its corresponding wrapper class is known as **autoboxing**.

For example, byte to Byte, char to Character, int to Integer, long to Long, float to Float, boolean to Boolean, double to Double, and short to Short.

UNBOXING

The automatic conversion of wrapper type into its corresponding primitive type is known as **unboxing**. It is the reverse process of autoboxing.

CREATING WRAPPER OBJECTS

To create a wrapper object, use the wrapper class instead of the primitive type.

```
public class Main {

  public static void main(String[] args) {
    Integer myInt = 5;
    Double myDouble = 5.99;
    Character myChar = 'A';
```

```
        System.out.println(myInt);
        System.out.println(myDouble);
        System.out.println(myChar);
    }
}
```

The following methods are used to get the value associated with the corresponding wrapper object: **intValue(), byteValue(), shortValue(), longValue(), floatValue(), doubleValue(), charValue(), booleanValue().**

```
public class Main {
    public static void main(String[] args) {
        Integer myInt = 5;
        Double myDouble = 5.99;
        Character myChar = 'A';
        System.out.println(myInt.intValue());
        System.out.println(myDouble.doubleValue());
        System.out.println(myChar.charValue());
    }
}
```

EXAMPLE

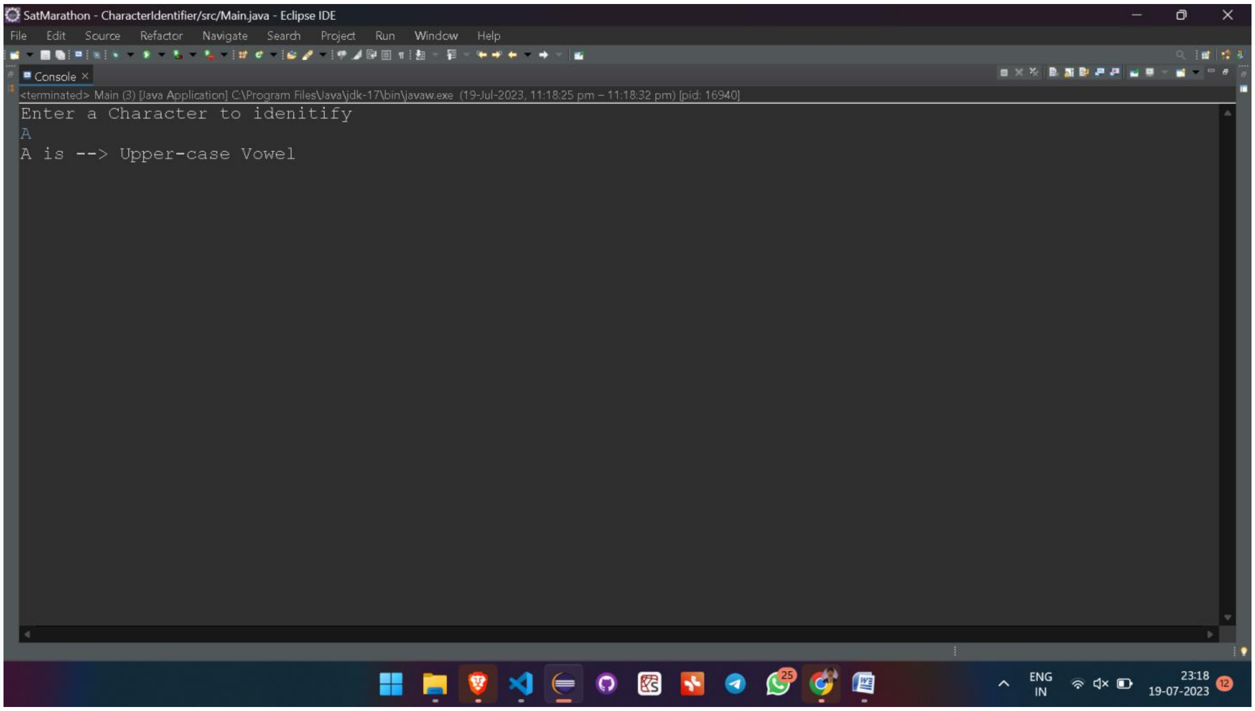
```
class Main {
    public static void main(String[] args) {
        // create primitive types
        int a = 5;
        double b = 5.65;
        //converts into wrapper objects
        Integer obj1 = Integer.valueOf(a);
        Double obj2 = Double.valueOf(b);
        if(obj1 instanceof Integer) {
        System.out.println("An object of Integer is created.");
        }
        if(obj2 instanceof Double) {
        System.out.println("An object of Double is created.");
        }
    }
}
```

valueOf() method to convert the primitive types into objects.

2. IDENTIFY ENTERED CHARACTER

```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a Character to idenitify");
        char ch = sc.next().charAt(0);
        identifyCharacter(ch);
        sc.close();
    }

    public static void identifyCharacter(char ch) {

        if (Character.isLowerCase(ch)) {
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                System.out.println(ch+" is --> Lower-case Vowel");
            }
            else {
                System.out.println(ch+" is --> Lower-case Consonant");
            }
        }

        else if (Character.isUpperCase(ch)) {
            if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
                System.out.println(ch+" is --> Upper-case Vowel");
            }
            else {
                System.out.println(ch+" is --> Upper-case Consonant");
            }
        }

        else if (Character.isDigit(ch)) {
            System.out.println(ch+" is --> Number");
        }

        else {
            System.out.println(ch+" is --> Special Character/Symbols");
        }
    }
}
```

OUTPUT

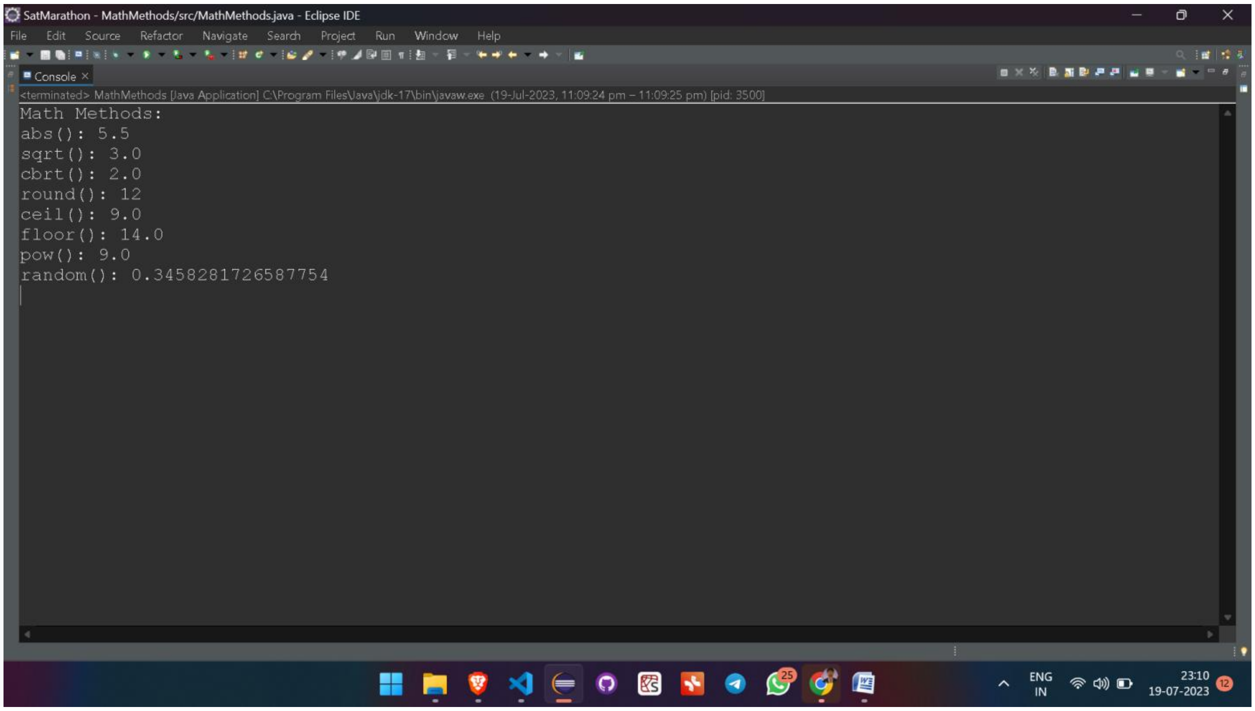

SatMarathon - CharacterIdentifier/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console ×

```
<terminated> Main (3) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (19-Jul-2023, 11:18:25 pm – 11:18:32 pm) [pid: 16940]
Enter a Character to idenitify
A
A is --> Upper-case Vowel
```

3. CLASS MATH METHODS

```
class MathMethods{

   public static void main(String[] args) {

      System.out.println("Math Methods: ");
      System.out.println("abs(): "+Math.abs(-5.5));
      System.out.println("sqrt(): "+Math.sqrt(9));
      System.out.println("cbrt(): "+Math.cbrt(8));
      System.out.println("round(): "+Math.round(11.6));
      System.out.println("ceil(): "+Math.ceil(8.1));
      System.out.println("floor(): "+Math.floor(14.8));
      System.out.println("pow(): "+Math.pow(3,2));
      System.out.println("random(): "+Math.random());

   }
}
```

OUTPUT


SatMarathon - MathMethods/src/MathMethods.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
ENG IN
23:10 19-07-2023

```
<terminated> MathMethods [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (19-Jul-2023, 11:09:24 pm – 11:09:25 pm) [pid: 3500]
Math Methods:
abs(): 5.5
sqrt(): 3.0
cbrt(): 2.0
round(): 12
ceil(): 9.0
floor(): 14.0
pow(): 9.0
random(): 0.3458281726587754
```

4. CLASS ARRAY METHODS

4.1 Array.equals() Method

```
import java.util.Arrays;

public class EqualsMethod {
   public static void main(String[] args) {
        int arr1[] = { 10, 20, 30 };
        int arr2[] = { 30, 20, 10 };
        int arr3[] = { 10, 20, 30 };

        // comparing arr1 and arr2
        boolean retval = Arrays.equals(arr1, arr2);
        System.out.println("arr1 and arr2 equal: " + retval);

        // comparing arr1 and arr3
        boolean retval2 = Arrays.equals(arr1, arr3);
        System.out.println("arr1 and arr3 equal: " + retval2);
     }
}
```

4.2 Array.sort() Method

```
import java.util.Arrays;

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

        int arr[] = { 10, 50, 30, 20, 150, 200, 10, 70 };
        System.out.print("Original Array: ");
        for (int i = 0; i < arr.length; i++) {
          System.out.print(arr[i] + " ");
        }

        System.out.println();

        Arrays.sort(arr);
        System.out.print("Sorted Array: ");
        for (int i = 0; i < arr.length; i++) {
          System.out.print(arr[i] + " ");
        }
      }
}
```

4.3 Array.toString() Method

```
import java.util.Arrays;

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

        int intArr[] = {10, 20, 30, 40, 50};
        char charArr[] = {'A', 'B', 'C' };

        System.out.println("String representation of byte array is: " +
Arrays.toString(intArr));
        System.out.println("String representation of char array is: " +
Arrays.toString(charArr));
      }
}
```

4.3 Array.compare() Method

```
import java.util.Arrays;

public class CampareMathod {
    public static void main(String[] args) {
            int arr1[] = { 10, 20, 30 };
            int arr2[] = { 30, 20, 10 };
            int result = Arrays.compare(arr1, arr2);

            if(result > 0) {
                System.out.println("arr1 is greater");
            } else if (result == 0) {
                System.out.println("both are same");
            } else {
                System.out.println("arr2 is greater");
            }
        }
}
```

4.3 Array.copyOf() Method

```
import java.util.Arrays;

public class CopyOfMethod {
    public static void main(String[] args) {
            short[] arr = { 10, 20, 30, 40, 50, 60, 70};
            System.out.print("Orinial Array: ");
            for (int i = 0; i < arr.length; i++) {
                System.out.print(arr[i] + " ");
            }
            System.out.println();

    //        short[] copyArr = Arrays.copyOf(arr, arr.length);
            short[] copyArr = Arrays.copyOf(arr, 5);
            System.out.print("Copy Array: ");
            for (int i = 0; i < copyArr.length; i++) {
                System.out.print(copyArr[i] + " ");
            }

        }
}
```