DAY-26

------

CONCATINATION OF TWO STRINGS.

Concatination of two strings can be done in two ways:

1. using the + operator

2. using the concate() method.

EXAMPLE:1

---------

class Demo

{

public static void main(String[] args)

{

String s1 = "sagar" + "ram";

String s2 = "sagar" + "ram";

System.out.println(s1==s2); // true

System.out.println(s1.equals(s2)); // true

}

}

EXAMPLE:2

----------

class Demo

{

public static void main(String[] args)

{

String s1 = "sagar";

String s2 = "sagarram";

String s3 = s1+"ram";

System.out.println(s2==s3); // false

System.out.println(s2.equals(s3)); // true

}

}

NOTE: Whenever we try to concatinatewith the use of variables of strings type then the memory will be allocated in non-constant pool.

In the above program string value is getting concatinated with a refrence variable hence string varibale s3 is allocated in non constant pool.

value+value --> constant pool

ref + value --> non-constant pool

EXAMPLE:3

---------

class Demo

{

public static void main(String[] args)

{

String s1 = "sagar";

s1.concat("ram");

System.out.println(s1); --> sagar

}

}

EXAMPLE:4

---------

class Demo

{

public static void main(String[] args)

{

String s1 = "sagar";

String s2 = "sagarram";

String s3 = "sagar";

s1.concat(s3);

s1= s1+"kumar";

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

System.out.println(s1==s3);

}

}

output:

sagarkumar

sagarram

sagar

false

EXAMPLE:5

---------

class Demo

{

public static void main(String[] args)

{

String s1 = "sagar";

String s2 = "sagarram";

System.out.println(s1 == s2); // false

String s3 = new String("sagar");

String s4 = new String("sagarram");

System.out.println(s3 == s4); // false

System.out.println(s2.equals(s4)); // true

String s5 = "sagarram";

String s6 = "sagar"+"ram";

String s7 = s1+"ram";

System.out.println(s6==s7); // false

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

System.out.println(s4);

System.out.println(s5);

System.out.println(s6);

System.out.println(s7);

}

}

OUTPUT:

-------

false

false

true

false

sagar

sagarram

sagar

sagarram

sagarram

sagarram

sagarram

WHY STRINGS ARE IMMUTABLE IN JAVA?

refer diagram

NOTE: in java one object can have multiple refrence variables.In the same way any modification done on the object will result in creation of brand

new object.

In a project if a string object is same for multiple refrences then instead of creating mutiple string object we create a single string object which

will be shared among multiple refrence variable.

Hence it improves the performance of the program.These strings can be created without using the new keyword and memory for such strings will be allocated

in constant pool.

The problem in this apporach is since multiple ref are pointing to same object modification made by one of the ref variable on object will affect the

other reference variables also,

inorder to overcome this problem strings are made it as immutable.ie once the string is created it can not be altered any modification perdformed will

result in creation of brand new object.

inbuilt methods in strings:

--------------------------

EXAMPLE:

----------

class Demo

{

public static void main(String[] args)

{

String s1 = new String("studyonline");

String s2 = new String("STUDYONLINE");

System.out.println(s1);

System.out.println(s2);

System.out.println(s1.toUpperCase());

System.out.println(s2.toLowerCase());

System.out.println(s1.length());

System.out.println(s1.startsWith("stu"));

System.out.println(s1.endsWith("ine"));

System.out.println(s1.contains("dyon"));

System.out.println(s1.indexOf("y"));

System.out.println(s1.charAt(4));

}

}

output:

-------

studyonline

STUDYONLINE

STUDYONLINE

studyonline

11

true

true

true

4

y

--------------------------------------------------------------------------------------------------------------------------------------------------