DAY-28

-------

MUTABLE STRINGS

Whenever the content of the string is frequently changing then dont create the string using String class rather create using StringBuffer or StringBuilder

class.

If we try to create a string when content are frequently changing using the string class for every modification new string objects will be created.

Hence it is not recommanded to create such Strings using string class.

EXAMPLE:1

--------

class Demo

{

public static void main(String[] args)

{

//String s1 = new String("sagar");

//String s2 = "sagar";

StringBuffer s1 = new StringBuffer("sagar");

System.out.println(s1);

}

}

OUTPUT:

-------

sagar

EXAMPLE:2

----------

class Demo

{

public static void main(String[] args)

{

String s1 = new String("sagar");

String s2 =s1.concat("ram");

System.out.println(s1);

System.out.println(s2);

}

}

OUTPUT:

-------

sagar

sagarram

EXAMPLE:3

----------

class Demo

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("sagar");

System.out.println(s1);

s1.append("ram");

System.out.println(s1);

}

}

OUTPUT:

-------

sagar

sagarram

EXPLAIN THE DIFFERENCE B/W == AND EQUALS() METHOD IN STRING CLASS AND STRINGBUFFER CLASS.

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

String class --> == will compare the memory address of two string objects.

equals() method will compare the data present in the two String objects.

StringBuffer --> == will compare the memory address of two string objects.

equals() method will compare the memory address of the two String objects.

EXAMPLE:4

---------

class Demo

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("sagar");

StringBuffer s2 = new StringBuffer("sagar");

System.out.println(s1);

System.out.println(s2);

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

}

}

OUTPUT:

-------

sagar

sagar

false

EXAMPLE:5

----------

class Demo

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("sagar");

StringBuffer s2 = new StringBuffer("sagar");

System.out.println(s1);

System.out.println(s2);

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

}

}

INTERNAL IMPLEMENATION OF THE MUTABLE STRING

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

Do refer the diagram:

EXAMPLE:1

----------

class Demo

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer();

System.out.println(s1);

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

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

s1.append("students");

System.out.println(s1);

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

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

s1.append(" prepare for exams");

System.out.println(s1);

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

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

s1.append(" and get placed in a good company");

System.out.println(s1);

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

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

}

}

OUTPUT:

---------

16

0

students

16

8

students prepare for exams

34

26

students prepare for exams and get placed in a good company

70

59

ensureCapacity() method.

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

In order to set the capacity for paticular value we can use ensureCapacity() method.

EXAMPLE:1

---------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer();

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

s1.ensureCapacity(1500);

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

}

}

OUTPUT:

-------

16

1500

EXAMPLE:2

---------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer(1500);

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

}

}

OUTPUT:

-------

1500

EXAMPLE:3

---------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("sagar");

System.out.println(s1.capacity()); // 16 + number of characters

// 16 + 5 = 21

}

}

OUTPUT:

-------

21

FEW INBUILT METHODS WITH RESPECT TO STRINGBUFFER CLASS

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

1. charAt(int index) --> to get the character at specific index.

2. setCharAt(int index, char ch) --> to set the character at the given index.

3. deleteCharAt(int index) --> to delete the character at the given index.

EXAMPLE:1

----------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("sagar");

System.out.println(s1);

System.out.println(s1.charAt(2));

s1.setCharAt(2,'p');

System.out.println(s1);

s1.deleteCharAt(2);

System.out.println(s1);

}

}

OUTPUT:

-------

sagar

g

sapar

saar

insert(int index, String s) method:

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

It is used to add string or int or float etc to given index.

EXAMPLE:1

---------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("sagar");

System.out.println(s1);

s1.insert(5,"is a good boy..!");

System.out.println(s1);

s1.insert(0,6);

System.out.println(s1);

}

}

OUTPUT:

-------

sagar

sagaris a good boy..!

6sagaris a good boy..!

delete(int begin index, int end index) method:

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

It is used to delete char fromm the begging index to end-1 index

EXAMPLE:

--------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("StudyOnline");

System.out.println(s1);

s1.delete(1,5);

System.out.println(s1);

}

}

OUTPUT:

-------

StudyOnline

SOnline

reverse() method:

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

It is used to reverse the given String

EXAMPLE:

--------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("StudyOnline");

System.out.println(s1);

s1.reverse();

System.out.println(s1);

}

}

OUTPUT:

-------

StudyOnline

enilnOydutS

setLength() method:

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

It is used to set the length of the string.

EXAMPLE:

--------

class Demo1

{

public static void main(String[] args)

{

StringBuffer s1 = new StringBuffer("StudyOnline");

System.out.println(s1);

s1.setLength(5);

System.out.println(s1);

}

}

OUTPUT:

-------

StudyOnline

Study