JAVA SE

(CORE JAVA)

LECTURE-30

Today's Agenda

- String Handling.
- Different classes to handle String
- Constructors and Methods of class String.

String Handling

- Java provides 3 classes to handle Strings as per situation, these are

1. String
2. StringBuffer
3. StringBuilder

*StringBuilder will be covered in the Multithreading chapter.

String class

- String objects in java are immutable i.e. content once stored cannot be changed.

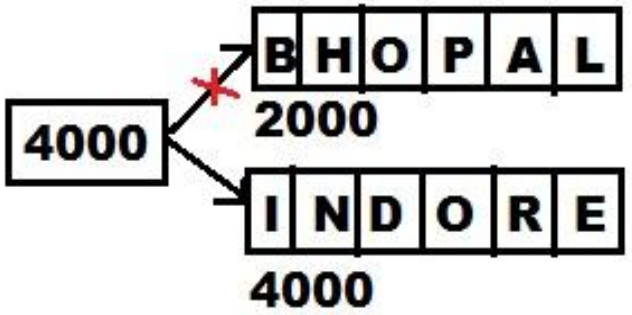
- For Example,

String city="Bhopal";

System.out.println(city);

city="Indore";

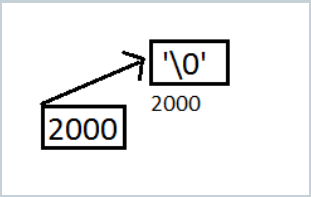
System.out.println(city);



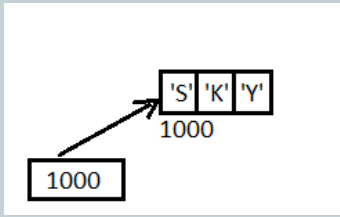
Though the output will change but the Objects won't.

Constructors of String

- **String()**:- String S=new String();

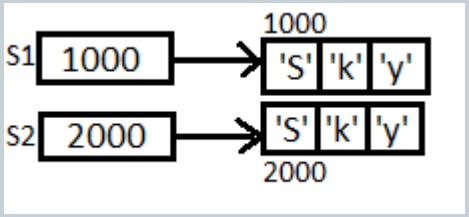


- **String(String)**:- String S=new String("Bhopal");

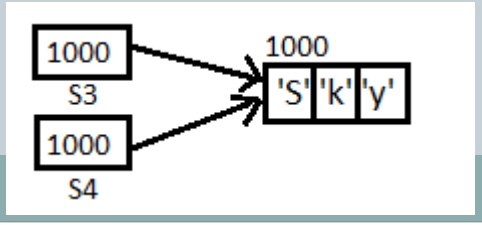


- **Difference in Initialization**:-

String s1=new String("Sky");
String s2=new String("Sky");



String s3="Sky";
String s4="Sky";



Constructors of String

- To check the memory diagram we can compare the object references,

String s1=new String(“Sky");

String s2=new String(“Sky");

String s3=“Sky";

String s4=“Sky";

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

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

```
F:\Java Codes>java Test1
false
true
```

Constructors of String

- **String(char[ ])**:- Coverts a character array to String object.

- **String(char[ ],int1,int2)**:-
  int1- Starting index
  int2- Number of characters to be converted into String

  char arr[ ]={'H', 'e', 'l', 'l', 'o'};

  String s=new String(arr,0,4);

  System.out.println(s); ⟶ **Hell**

- ***In java anything in " "(double quotes) is considered to be a string to be precise a String object.***
- ***Example :- "Bhopal".length(); ⟶ 6***

Methods of String class

- **public boolean equals(Object):-** Derived from Object class. It compares object references when object of any other class is passed. But it compares the strings when a String is passed. So, every class can override equals in its own way.

- **public boolean equalsIgnoreCase(String)**:- Method belongs to String class and ignores case sensitivity.

Methods of String class

- **public int compareTo(String)**:- Method belongs to String class and compares string and returns 0 if true else difference of their ASCIIs.
- **public int compareToIgnoreCase(String)**:- Similar to above method but ignores case sensitivity.
- **public int indexOf(int)** :- Returns index of the character present in the string, which is passed in the argument. If not found returns -1. It is a case sensitive method.
- **public int indexOf(String)**:- Accepts a substring as argument and returns the beginning index where the substring occurs.
- **public int length()**:- Gives length of string.

Methods of String class

- **public char charAt(int)**:-Takes index number and gives character at that index
- **public void getChars(int, int, char[ ], int)**:- Takes multiple characters and pastes their copy to an array of characters.
- **public boolean startsWith(String)**:- Tests if this string starts with the specified prefix.
- **public boolean startsWith(String,int)**:- Tests if this string starts with the specified prefix beginning a specified index.
- **public boolean endWith(String)**:- Tests if this string ends with the specified suffix.

Methods of String class

- **public int lastIndexOf(int)**:- Returns the index within this string of the last occurrence of the specified character.
- **public int lastIndexOf(String)**:- Returns the index within this string of the rightmost occurrence of the specified substring.
- **public String substring(int, int)**:- Returns a new string that is a substring of this string. The first argument is starting index for substring and second argument is end index-1 of the substring.
- **public String substring(int)**:- Returns the substring from index passed as argument till the last index of the string.

Methods of String class

- **public String toUpperCase( )**:- Converts all the characters of the String to upper case.

- **public String toLowerCase( )**:- Coverts all the characters of the String lower case.

➢ *There won't be any change in the calling String object, just a copy of that String will be returned.*

- **public static String valueOf(any primitive data type)**:- Returns the string representation of the passed data type argument.

Class StringBuffer

- The objects of class StringBuffer in java are mutable i.e. content of an object can be changed without creating a new object.
- StringBuffer is used when data of a class may change in future. Example, Salary of an employee.
- StringBuffer also has same methods as that of the class String except some of them.
- StringBuffer is also present in the package java.lang.

Constructors of StringBuffer

- **public StringBuffer( )**:- Creates an object with size 16 characters initialized with '\0'.

- **public StringBuffer(int)**:- Creates a string buffer with specified capacity in the argument and initialized with null character s.

- **public StringBuffer(String)**:- The object is created and initialized with the string passed in the argument and is appended with 16 null characters('\0').

Methods of Class StringBuffer

- **public int capacity( )**:- This method returns the current capacity. Using this method we can confirm the extra 16 characters reserved by java.

- **public void ensureCapacity(int)**:- Increases capacity to the argument passed.

- **public StringBuffer append(String)**:- An overloaded function and can append any data type.

  StringBuffer s=new StringBuffer(“India”);

  s.append(“is my country”);

  System.out.println(s);

Methods of Class StringBuffer

- **public StringBuffer reverse( )**:- As the name suggests it reverses the original string.

start index, end index+1, new string

- **public StringBuffer replace(int, int, String)**:- This method replaces the characters in a substring of this sequence with characters in the specified String.

```
StringBuffer s=new StringBuffer("Hello World");
s.replace(6, 11, "India");
System.out.println(s);
```

⟶ **Hello India**

End Of Lecture 30

Thank You

For any queries mail us @: scalive4u@gmail.com
Call us @ : 0755-4271659, 7879165533

Agenda for Next Lecture:

1. **Graphical User Interface in Java**