Day 11
Assignment } 07.05.2023

STRING :- String is a non-primitive datatype. Act as a class and also a datatype.

- String is a sequence of characters But in java string is an object that represents a sequence of characters

- String enclosed with double Quotes and works on the principles of index basis (starts from 0-n-1)

- String is a class and it is in java.lang package. Additionally most of predefined methods and classes in java from java.lang hence explicit import does not happen.

- The java.lang.String class is used to create a string object.

- String values are stored in String constant pool (SCP). SCP is a type of memory which is shared by heap memory.

- String is literal and immutable.

There are two ways to create String object.

[- By String literal
- By new keyword]

String literal:

- Java string literal is created by using double quotes.

For example:

String s = "welcome";

Each time you create a string literal, the JVM checks the "string constant pool", a reference to the pooled instance is returned if string is doesn't exists in the pool. a referenced to the pooled instance is returned. If the string is doesn't exist in the pool a new string instance is created and placed in the pool.

for example:

String s = "welcome";
String s2 = "welcome"; // It does not create new instance.

By new keyword:

String s = new String ("welcome");

The string can also be declared using new operator i.e. dynamically allocated.

In case of string are dynamically allocated they are assigned a new memory location in heap. The string will not be added to string constant pool.

Example program for string literal:

```
package org.string.methods;
public class StringMethods{
    public static void main(String
[] args)
```

```
{ String s = "welcome";
  String s1 = "welcome";
  string s2 = "Java";
  string s3 = "Java";
System.out.println (s.hashcode());
System.out.println (s1.hashcode());
System.out.println (s3.hashCode());
System.out.println (s2.hashCode());
}}
```

STRING METHODS:-

toUpperCase(), toLowerCase(), StartsWith()
endswith(), indexof(), lastIndexOf()
equals(), equalsIgnoreCase(), length()
contains(), concat(), charAt(),
toCharArray(), replace(), isEmpty(),
isBlank(), split(), trim(), subString()

```
package org.string;
public class StringConcepts {
    public static void main (String [] args)
    { String s = "march Batch Three";
      String s1 = "Students";
      String s3 = " ";

      int length = s.length();
      System.out.println(length);

      String upper = s.toUppercase();
      System.out.println (upper);

      String lower = s.toLowerCase();
      System.out.println (lower);
      System.out.println (s.startswith("m"));
      System.out.println (s.endswith("e"));
```

```
System.out.println(s.indexOf('a'));
                        // left to right

System.out.println(s.lastIndexOf('a'));
                        // right to left

System.out.println(s.lastIndexOf('a'));
System.out.println(s.equals(s1));
System.out.println(s.equalsIgnoreCase(s1));
                        // will not considered
                           upper and lower case

System.out.println(s.contains("march"));
String concat = s.concat(s1);
System.out.println(concat);

char charAt = s.charAt(7);
System.out.println(charAt);

System.out.println(s.replace("march", "may"));
System.out.println(s3.isEmpty()); // consider space
System.out.println(s3.isBlank()); // doesnot
                           consider space.

System.out.println(s);
System.out.println(s.trim());
System.out.println(s.substring(3));
System.out.println(s.substring(3,10));
```

Immutable String :

- A string is an immutable object which means we cannot change them after creating the objects. Whenever we change any string, a new instance is created.

Immutable refers to something that cannot be changed or modified. Hence, immutable objects are ones that cannot be modified after they have been created.

String Builder and StringBuffer (Immutable to mutable String)

```
package org.string;
public class MutableString {
 public void immutableString()
  {
    System.out.println("Immutable String");
    String s = "March Batch";
    String s1 = "Three";
    System.out.println(s.hashCode());
    String concat = s.concat(s1);
    System.out.println(concat.hashCode());

}
public void mutableString()
 {  System.out.println("Mutable String");
    StringBuffer sb = new StringBuffer("March
                                  Batch");
    System.out.println(sb.hashCode());
    System.out.println(sb.append("Three"));
    System.out.println(sb.hashCode());
    System.out.println(sb.insert(2, 'r');
    System.out.println(sb.deleteCharAt(2));
    System.out.println(sb.replace(0,4,"May"));

}
public static void main(String[] args)
 { MutableString ms = new MutableString();
   ms.immutableString();
   ms.mutableString();
}
```

Points to remember:-

- String is immutable whereas StringBuffer and StringBuilder are mutable classes.

- StringBuffer is threadsafe and Synchronized whereas StringBuilder is not, that is why StringBuilder is faster than StringBuffer

- String concatenation operator (+) internally uses StringBuffer or StringBuilder class.

- for String manipulation in a non multi-threaded environment, we use StringBuilder else use StringBuffer class.

NOTES TAKING :- - 07.05.2023

Day - 11

STRING :

Object refer the sequence of characters.

index based.

String s = "academy" → String literal

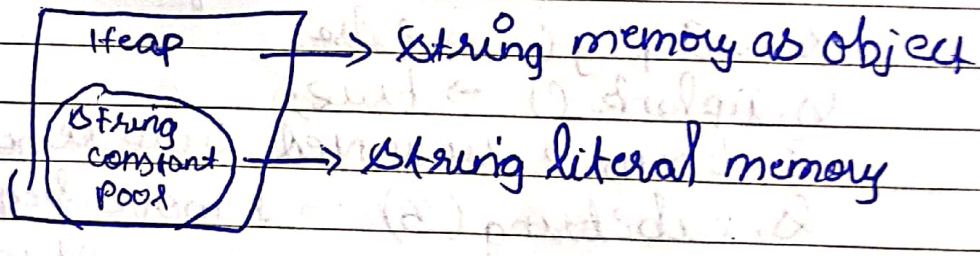
s	a	c	a	d	e	m	y
	0	1	2	3	4	5	6

0 to n-1

java.lang

String is literal and immutable.

```
String s = new String("March Batch Three");
Sysо(s);
Sysо(s1);
```



hashCode(); → method :.

s.hashCode(); → retervie address of variable.

Uppercase :.

```
String u = s.toUpperCase();
Sysо(u);
```

Small case - String methods return type

→ case sensitive

S.startsWith("m"); → String starts with.
S.endswith("e"); → Strng ends with
S.hascode(); → ~~find length~~ find address
S.length(); → findlength
S.toUppercase(); → lower to upper
S.tolowercase(); → upper to lower.
S.indexof('b'); → index of B
S.lastIndexof('a'); → last index of a
S.equals(S1) → check if two strngs are equals.
S.equalsIgnoreCase(S)); → case sensitive -no
S.contains("saite"); → check if word is found
~~S.concat~~ S.concat(S1); → concat.
S.charAt(4); → return 4th character in String.
S.replace("March", "May");
S.isEmpty() → false;
S.isBlank() → true;
S.trim → unwanted space remove.
S.subString(3) → remove begin and print from 3rd index.
S.subString(3,n); → starts with 3 and ends with n-1

immutable :-

StringBuffer
StringBuilder

StringBuffer sb = new StringBuffer("March Gated");
sb.append("three");
Syso(sb);
Syso(sb.hashcode);
Syso(sb.deleteCharAt(2));
Syso(sb.insert(2,'n'));

Buffer

String ~~Builder~~ - Synchronized thread safe

String Builde - Asynchronized Non-thread Safe