Dt : 25/9/2023

=>we can perform the following operations on StringBuffer Object:

(i)insert()

(ii)delete()

(iii)reverse()

Program : DemoBuffer5.java

```
package maccess;
public class DemoBuffer5
{
    public static void main(String[] args)
    {
       StringBuffer sb = new StringBuffer();
       sb.append("java language program");
       System.out.println("data in sb :
"+sb.toString());
       System.out.println("*****insert()*****");
       sb.insert(5, "HYD-NIT ");
       System.out.println("data in sb :
"+sb.toString());
       System.out.println("****delete()*****");
       sb.delete(5, 13);
       System.out.println("data in sb :
"+sb.toString());
       sb.deleteCharAt(7);
       System.out.println("data in sb :
"+sb.toString());
       System.out.println("****reverse()*****");
       sb.reverse();
       System.out.println("data in sb :
"+sb.toString());
    }
}
```

o/p:

```
data in sb : java language program
*****insert()*****
data in sb : java HYD-NIT language program
****delete()*****
data in sb : java language program
data in sb : java laguage program
****reverse()*****
data in sb : margorp egaugal avaj
```

===

Assignment:

wap to read a String and check the String is Palindrome String or not?

(Using reverse() method)

==

Note:

=>StringBuffer class is known as "Synchronized Class".

faq:

define Synchronized Class?

=>The class which is declared with "synchronized methods" is known as synchronized class.

faq:

define synchronized methods?

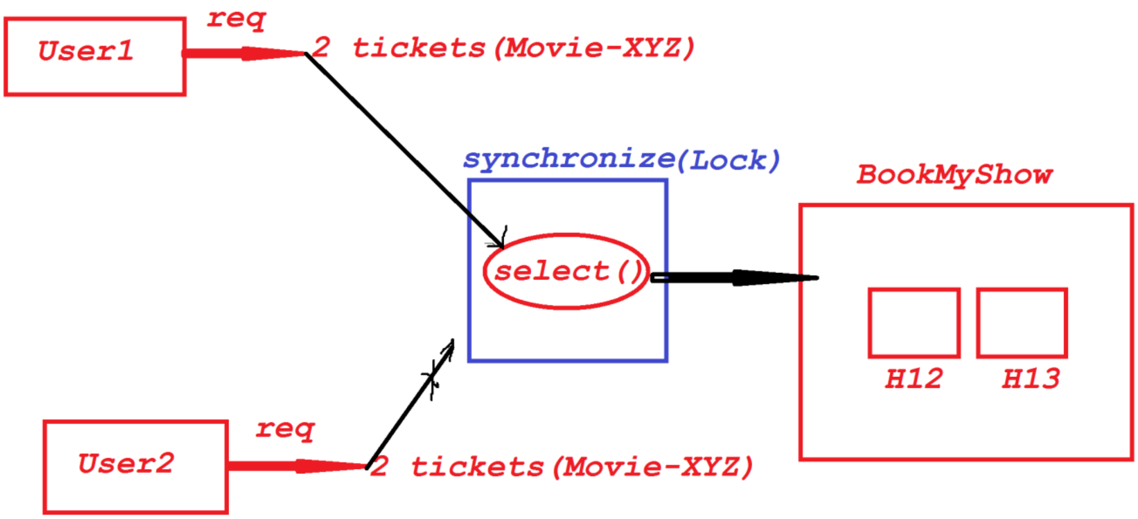
=>The methods which are declared with "synchronized" keyword are known as synchronized methods.

faq:

wt is the behaviour of synchronized method?

=>These synchronized methods will be under the synchronized-lock and the methods are used by one user at-a-time.

Diagram:



===

(c)StringBuilder:

=>StringBuilder class is from java.lang package and which also generate Mutable Objects.

=>The following are some important constructors from the StringBuilder:

```
public java.lang.StringBuilder();
public java.lang.StringBuilder(int);
public java.lang.StringBuilder(java.lang.String);
public java.lang.StringBuilder(java.lang.CharSequence);
```

=>StringBuilder is having same behaviour like StringBuffer,but StringBuilder is NonSynchronized class.

=>NonSynchronized class means the class is holding NonSynchronized methods.

Note:

=>In realtime,StringBuffer is used in Multi-User Applications and StringBuilder is used in Single-User Applications.

==

faq:

define Utility Classes?

=>The Classes which perform operations on other Objects are known as Utility Classes.

=>The following are two Utility classes related to String-Objects:

(a)StringTokenizer class

(b)StringJoiner class(Java8 - new Component)

(a)StringTokenizer class:

=>StringTokenizer class is from java.util package and which is used to break the given string into pieces(tokens) based on "delimiter"

(delimiter - means break specification)

=>The following are some important methods of StringTokenizer:

public java.util.StringTokenizer

(java.lang.String, java.lang.String);

public boolean hasMoreTokens();

public java.lang.String nextToken();

public java.lang.String nextToken(java.lang.String);

public boolean hasMoreElements();

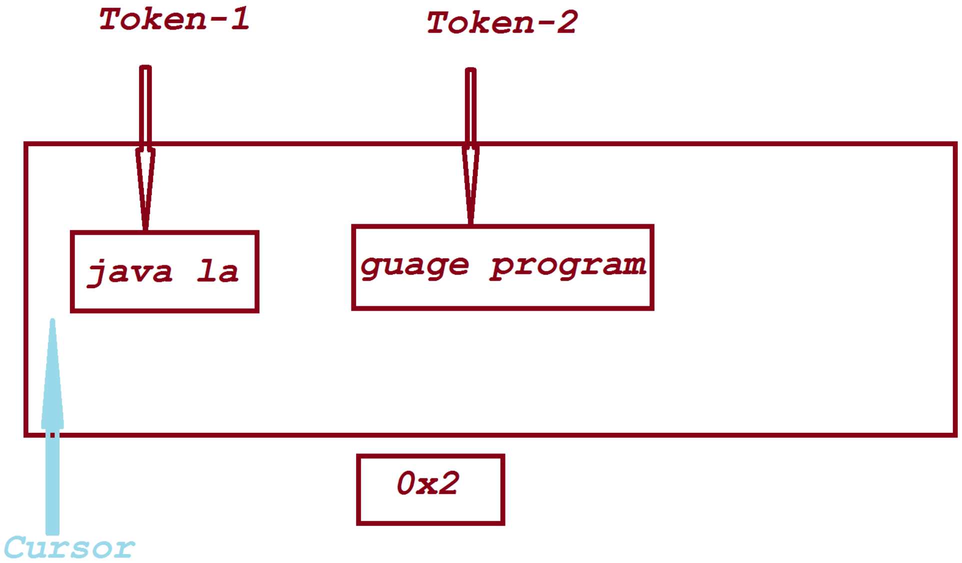
public java.lang.Object nextElement();

public int countTokens();

Program : DemoStringTokenizer1.java

```
package maccess;
import java.util.*;
public class DemoStringTokenizer1 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try(s;){
        System.out.println("Enter the String:");
        String str = s.nextLine();
        System.out.println("Enter the delimiter:");
        String dl = s.nextLine();
        StringTokenizer ob1 = new
StringTokenizer(str,dl);
        StringTokenizer ob2 = new
StringTokenizer(str,dl);
        System.out.println("Str : "+str.toString());
        System.out.println("Count of
Tokens:"+ob1.countTokens());
        System.out.println("****Tokens-
hasMoreElements()*****");
        while(ob1.hasMoreElements())
        {
            String ele = (String)ob1.nextElement();
            System.out.println("Token :
"+ele.toString());
        }
        System.out.println("Count of
Tokens:"+ob1.countTokens());
        System.out.println("****Tokens-
hasMoreTokens()****");
        while(ob2.hasMoreTokens())
        {
         String ele = ob2.nextToken();
         System.out.println("Token : "+ele.toString());
        }
        System.out.println("Count of
Tokens:"+ob2.countTokens());

        }//end of try with resource
    }
}
```

```
o/p:

Enter the String:

java language program

Enter the delimiter:

n

Str : java language program

Count of Tokens:2

****Tokens-hasMoreElements()*****

Token : java la

Token : guage program

Count of Tokens:0

****Tokens-hasMoreTokens()****

Token : java la

Token : guage program

Count of Tokens:0

==============================================================
str = "java language program"        dl="n"
```



Token-1

Token-2

java la

guage program

0x2

Cursor

```
StringTokenizer ob = new StringTokenizer(str,dl);
```