Dt : 20/9/2022

case-1 : Creating object using 'java.lang.StringBuffer()'

syntax:

StringBuffer sb = new StringBuffer();

=>In this syntax the StringBuffer object is created with the default capacity 16.

=>when the length crossed the capacity then the capacity increases automatically by doubling the capacity and adding 2.

=>we perform the following important operations on StringBuffer:

- *append()*
- *insert()*
- *delete()*
- *reverse()*

Ex : DemoBuffer1.java

```
package maccess;
import java.util.*;
public class DemoBuffer1 {
      public static void main(String[] args) {
          Scanner s = new Scanner(System.in);
          StringBuffer sb = new StringBuffer();//Con_Call
          System.out.println("====Details of Buffer====");
          System.out.println("default capacity:"+sb.capacity());
          System.out.println("length:"+sb.length());
          while(true) {
            System.out.println("====Choice====");
            System.out.println
            ("1.append()\n2.insert(int,String)"
                        +
"\n3.delete(int,int)\n4.reverse()\n5.exit");
            System.out.println("Enter the Choice:");
            int choice = Integer.parseInt(s.nextLine());
            switch(choice)
```

```
            {
            case 1:
                  System.out.println("Enter the data:");
                  sb.append(s.nextLine());//Adding data to Buffer
                  System.out.println("====Details of Buffer====");
                  System.out.println("Buffer data :
"+sb.toString());
                 System.out.println("capacity:"+sb.capacity());
                 System.out.println("length:"+sb.length());
                  break;
            case 2:
                  if(sb.length()>0) {
                        System.out.println
                        ("Enter the index to perform insert
operation:");
                        int i1 = Integer.parseInt(s.nextLine());
                        if(i1<sb.length()) {
                              System.out.println("Enter the data:");
                              String dt = s.nextLine();
                              sb.insert(i1,dt);//inserting data by
index
                              System.out.println("====Details of
Buffer====");
                              System.out.println("Buffer data :
"+sb.toString());
                        }else {
                              System.out.println("Invalid index
value...");
                        }
                  }else {
                        System.out.println("StringBuffer is
empty....");
                  }
                  break;
            case 3:
                  if(sb.length()>0) {
                        System.out.println("Enter the Starting
index:");
                        int sIndex = Integer.parseInt(s.nextLine());
                        if(sIndex>=0 && sIndex<sb.length()) {
                              System.out.println("Enter the ending
index:");
                              int eIndex =
Integer.parseInt(s.nextLine());
                              if(eIndex>sIndex && eIndex<sb.length())
{
                                    sb.delete(sIndex, eIndex);
```

```
                                        System.out.println("====Details of
Buffer====");
                                        System.out.println("Buffer data :
"+sb.toString());
                                   }else {
                                        System.out.println("Invalid ending
Index...");
                                   }
                              }else {
                                   System.out.println("Invalid Starting
index..");
                              }
                         }else {
                              System.out.println("StringBuffer is
empty...");
                         }
                         break;
                    case 4:
                         if(sb.length()>0) {
                              sb.reverse();
                              System.out.println("====Details of
Buffer====");
                              System.out.println("Buffer data :
"+sb.toString());
                         }else {
                              System.out.println("StringBuffer is
empty...");
                         }
                         break;
                    case 5:
                         System.out.println("Buffer operations
Stopped...");
                         System.exit(0);//stop the program
                    default:
                         System.out.println("Invalid choice....");
               }//end of switch
          }//end of loop
     }
}
```


==========

Assigment-1:

wap to check the given String is palindrome String or not,using

pre-defined method?

Assignment-2:

wap to read a String and separate as follows:

buffer1 : vowels

buffer2 : Consonents

buffer3 : Numbers

buffer4 : others

Ex :

i/P : java18 by 2022 99% LTS product

Vowels : a a o u

Consonents : j v b y L T S p r d c t

Numbers : 1 8 2 0 2 2 9 9

others : %

==========

Case-2 : Creating object using 'java.lang.StringBuffer(int)'

syntax:

StringBuffer sb = new StringBuffer(4);

=>In this syntax the StringBuffer object is created with the default capacity equal to the value passed as parameter while Object creation process.

Ex : DemoBuffer2.java

```
package maccess;
public class DemoBuffer2 {
	public static void main(String[] args) {
		StringBuffer sb = new StringBuffer(4);//Con_Call
		System.out.println("===Details of Buffer===");
		System.out.println("default capacity:"+sb.capacity());
		System.out.println("length:"+sb.length());
		sb.append("java");
		System.out.println("===Details of Buffer===");
		System.out.println("data : "+sb.toString());
		System.out.println("capacity:"+sb.capacity());
		System.out.println("length:"+sb.length());
		sb.append("J");
		System.out.println("===Details of Buffer===");
		System.out.println("data : "+sb.toString());
		System.out.println("capacity:"+sb.capacity());
		System.out.println("length:"+sb.length());

	}
}
```

o/p:

===Details of Buffer===

default capacity:4

length:0

===Details of Buffer===

data : java

capacity:4

length:4

===Details of Buffer===

data : javaJ

capacity:10

length:5

==========

Case-3 : Creating object using 'java.lang.StringBuffer(String)'

syntax:

StringBuffer sb = new StringBuffer("NITHYD");

=>In this syntax the StringBuffer object is created with the default capacity equal to the sum of "16 + length of String passed as parameter while object creation".

Ex : DemoBuffer3.java

```
package maccess;
public class DemoBuffer3 {
	public static void main(String[] args) {
		StringBuffer sb = new StringBuffer("NITHYD");//Con_Call
		System.out.println("===Details of Buffer===");
		System.out.println("data : "+sb.toString());
		System.out.println("default capacity:"+sb.capacity());
		System.out.println("length:"+sb.length());
		sb.append("javalanguageprogram");
		System.out.println("===Details of Buffer===");
		System.out.println("data : "+sb.toString());
		System.out.println("capacity:"+sb.capacity());
		System.out.println("length:"+sb.length());

	}
}
```

o/p:

===Details of Buffer===

data : NITHYD

default capacity:22

length:6

===Details of Buffer===

data : NITHYDjavalanguageprogram

capacity:46

length:25

==========