

5 October 2020

Aayush Shah

19BCE245

# Practical 7

## OOP Lab

### Practical 7 A

Write a program to perform following operations on string  
“Nirma University”

- Reverse the string
- Replace character Ni with Ab
- Check whether strings “rma” and “Uni” present in original string or not
- Compare this program implementation using String and StringBuffer methods.

### CODE

**FILE : Practical7a.java**

```
class Practical7a {
    public static void main(String[] args) {
        String s1 = new String("Nirma University");
//7a part 1
        //method 1
        String rev1 = "";
        for (int i=s1.length()-1;i>=0;i--) {
            rev1 = rev1 + s1.charAt(i);
        }
        System.out.println("Reversed String : " + rev1);

        //method 2
        StringBuilder rev2 = new StringBuilder(s1);
        System.out.println(rev2.reverse());

        //method 3
        char[] charString = new char[s1.length()];
```

```

charString = s1.toCharArray();
System.out.print("char array : ");
for (int i=0;i<s1.length();i++) {
    System.out.print(charString[i]);
}
System.out.println();
for(int i=0;i<s1.length()/2;i++){
    char temp = charString[i];
    charString[i] = charString[s1.length()-1-i];
    charString[s1.length()-1-i] = temp;
}
System.out.print("reversed char array : ");
for (int i=0;i<s1.length();i++) {
    System.out.print(charString[i]);
}
System.out.println();

```

//7a part 2

```

//method 1
String replacedString = s1.replace("Ni", "Ab");
System.out.println(replacedString);
//method 2    [same as above just storing it in dummy string]
System.out.println(s1.replace("Ni", "Ab"));

```

//7a part 3

```

//method 1
if(s1.indexOf("rma")>=0)
    System.out.println("rma" + " is present in " + s1);
else    //returns -1
    System.out.println("rma" + " is not present in " + s1);
if(s1.indexOf("Uni")>=0)
    System.out.println("Uni" + " is present in " + s1);
else    //returns -1
    System.out.println("Uni" + " is not present in " + s1);

//method 2
System.out.print("rma is in the " + s1 + " : ");
System.out.println(s1.contains("rma"));
System.out.print("Uni is in the " + s1 + " : ");
System.out.println(s1.contains("Uni"));

```

//7a part 4

```

    }
}

```

**INPUT :**

-

**OUTPUT :**

```
Reversed String : ytisrevinU amriN
ytisrevinU amriN
char array : Nirma University
reversed char array : ytisrevinU amriN
Abrma University
Abrma University
rma is present in Nirma University
Uni is present in Nirma University
rma is in the Nirma University : true
Uni is in the Nirma University : true
```



Run Succeeded

Time 141 ms



Practical7a ↕

Tabs: 4 ↕

66 lines, 1831 characters

**CONCLUSION :**

From the practical 7 A, We learned about the concepts of String and its predefined methods and their uses. Which includes replace, reverse, charAt, length etc. also the difference between String , StringBuffer and StringBuilder.

## Practical 7 B

Write a program to find number of vowels, consonants and digits from an entered string using switch case.

### CODE

#### **FILE : Practical7b.java**

```
import java.util.Scanner;
class Practical7b {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = new String();
        System.out.print("Enter String : ");
        s = sc.nextLine();
        s = s.toLowerCase();
        int countVowels = 0;
        int countConsonents = 0;
        int countDigits = 0;
        int countSymbols = 0;
        for(int i=0;i<s.length();i++){
            switch (s.charAt(i)) {
                case 'a','e','i','o','u':
                    countVowels++;
                    break;
                case '1','2','3','4','5','6','7','8','9','0':
                    countDigits++;
                    break;
                case 'b','c','d','f','g','h','j','k','l','m','n','p','q','r','s','t','v','w','x','y','z':
                    countConsonents++;
                    break;
                default:
                    countSymbols++;
                    break;
            }
        }
        System.out.println("In the String " + s + " : ");
        System.out.println("Digits : " + countDigits);
        System.out.println("Vowels : " + countVowels);
        System.out.println("Consonents : " + countConsonents);
        System.out.println("Symbols : " + countSymbols);
    }
}
```

```
}
```

**INPUT :**

*aayush3490*

**OUTPUT :**

```
Enter String : aayush3490
In the String aayush3490 :
Digits : 4
Vowels : 3
Consonents : 3
Symbols : 0
```

✓ Run Succeeded

Time 181 ms

C Practical7b ↕

Tabs: 4 ↕

Line 35, Column 2

**CONCLUSION :**

From the practical 7 B, We again got a hands-on String and its uses. Here we used the `charAt()`, Which is useful for extracting the specific char.

## Practical 6 C

Write a program to reverse words in a string.  
For example, if input is “Welcome to Nirma”. Output should be “emocleW ot amriN”.

### CODE :

#### **FILE : Practical7c.java**

```
import java.util.Scanner;
class Practical7c {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = new String();
        System.out.print("Enter String : ");
        s = sc.nextLine();

        //method 1
        StringBuilder s2 = new StringBuilder();
        for(int i=0;i<=s.length();i++){
            if(i==s.length() || s.charAt(i) == ' '){
                System.out.print(s2.reverse() + " ");
                s2 = new StringBuilder();
            }
            else{
                s2.append(s.charAt(i));
            }
        }
    }
}
```

**INPUT :**

*I am Aayush Shah*

**OUTPUT :**

```
Enter String : I am Aayush Shah
I ma hsuyaA hahS
```



Run Succeeded

Time 145 ms

Symbol ↕

Tabs: 4 ↕

21 lines, 481 characters

**CONCLUSION :**

For the practical 7 C, We used the stringBuilder as String is immutable and we have to make changes in it so we have to take a string object which is mutable so that we taken stringBuidler which provides functionalities such as reverse, appends etc..

## Practical 7 D

Accept a paragraph of text consisting of sentences that are terminated by either '.' (full stop), '!' (exclamation mark) or a '?' (question mark). Assume that there can be maximum 10 sentences in a paragraph. Write a program to arrange the sentences in increasing order of their number of words.

For Example :

INPUT: Please come and attend the party. Hello! How are you?

OUTPUT :

Hello = 1

How are you = 3

Please come and attend the party = 6

### CODE :

#### **FILE : Practical7d.java**

```
import java.util.Scanner;
public class Practical7d {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        StringAndWords[] Paragraph = new StringAndWords[10];
        String original = new String();
        StringBuilder sentence = new StringBuilder();
        int count = 0;
        System.out.print("Enter String : ");
        original = sc.nextLine();
        int countIndex = 0;
        int end = 0;
        for(int i=0;i<original.length();i++){
            count = 0;
            if(original.charAt(i)=='.' || original.charAt(i)=='!' ||
original.charAt(i)=='?'){
                for(int j=end;j<i;j++){
                    if(original.charAt(j)==' ' && (j!=0 && original.charAt(j-1)!='.'
&& original.charAt(j-1)!='!' && original.charAt(j-1)!='?') )
                        count++;
                }
                Paragraph[countIndex] = new StringAndWords(original.substring(end,i), count);
                countIndex++;
                end = i;
            }
        }
        // Sorting the array based on word count
        for(int i=0;i<Paragraph.length;i++){
            for(int j=i+1;j<Paragraph.length;j++){
                if(Paragraph[i].count > Paragraph[j].count){
                    // Swap
                    StringAndWords temp = Paragraph[i];
                    Paragraph[i] = Paragraph[j];
                    Paragraph[j] = temp;
                }
            }
        }

        // Printing the sorted sentences
        for(int i=0;i<Paragraph.length;i++){
            System.out.println(Paragraph[i].sentence + " = " + Paragraph[i].count);
        }
    }
}
```



```

        sentence.append(original.charAt(j));
    }
    Paragraph[countIndex++] = new
StringAndWords(count+1,sentence);
    sentence = new StringBuilder();
    end = i+1;
    count=0;
}
}
for(int pick=0;pick<countIndex;pick++){
    for(int comp=pick+1;comp<countIndex;comp++){
        if(Paragraph[pick].count>Paragraph[comp].count){
            int temp = Paragraph[pick].count;
            Paragraph[pick].count = Paragraph[comp].count;
            Paragraph[comp].count = temp;
            StringBuilder tempStr = new
StringBuilder(Paragraph[pick].str);
            Paragraph[pick].str = Paragraph[comp].str;
            Paragraph[comp].str = tempStr;
        }
    }
}

for(int i=0;i<countIndex;i++)
    System.out.println(Paragraph[i].str + " = " + Paragraph[i].count);
}
}
class StringAndWords{
    int count;
    StringBuilder str = new StringBuilder();
    StringAndWords(int count,StringBuilder sentence){
        this.count = count;
        this.str = sentence;
    }
}
}

```

## INPUT :

Please come and attend the party. Hello! How are you?

## OUTPUT :

```
Enter String : Please come and attend the party. Hello! How are you?  
Hello = 1  
How are you = 3  
Please come and attend the party = 6
```

✓ Run Succeeded | Time 187 ms | Symbol ↕ | Tabs: 4 ↕ | 47 lines, 1548 characters

## CONCLUSION :

For the practical 7 D, We made use of `StringBuilder` and for loops to extract words from it, then we count the words and append it through string builder's method `append` to `StringBuilder` 'sentence' and then sorted these array of strings as their word count number.