Name: **Dhadge Yash Kailas**

Roll No: **9019**

Class: **BE-IT**

Course: **Lab Practice – V**

**(Distributed Systems)**

**Assignment No: 02**

Problem Statement:

**Develop any distributed application using CORBA to demonstrate object brokering. (Calculator or String operations).**

# Program:

**1) StringOperation.idl**

|  |
| --- |
| module SimpleString  {  interface StringOperation  {  string get\_String(in string symbol);  };  }; |

**2) StringOperationImpl.java**

|  |
| --- |
| import org.omg.CosNaming.\*;  import org.omg.CORBA.\*;  import SimpleString.\*;  import java.util.\*;  import java.io.\*;  import java.lang.\*;  import java.io.DataInputStream;  public class StringOperationImpl extends \_StringOperationImplBase  {  public String get\_String(String symbol)  {  Scanner sc = new Scanner(System.in);  DataInputStream din=new DataInputStream(System.in);  System.out.println("1.Get Length");  System.out.println("2.Join Two Java Strings");  System.out.println("3.Compare two Strings");  System.out.println("4.Replace the character");  System.out.println("5.Reverse the String");  System.out.println("\nEnter your choice : ");  int ch = sc.nextInt();  switch(ch)  {  case 1:  {  int length = symbol.length();  String len = String.valueOf(length);  return len;  }  case 2:  {  try{  System.out.println("\nEnter the String to concat : ");  String str =din.readLine();  String joinedString = symbol.concat(str);  return joinedString;  }  catch(Exception e)  {  e.printStackTrace();  }  }  case 3:  {  try{  System.out.println("\nEnter the String to compare : ");  String str2 =din.readLine();  boolean result1 = symbol.equals(str2);  return String.valueOf(result1);  }  catch(Exception e)  {  e.printStackTrace();  }  }  case 4:  {  System.out.print("Input a character to be replaced : ");  char c = sc.next().charAt(0);  System.out.print("Input a character to be replaced with : ");  char c1 = sc.next().charAt(0);  return symbol.replace(c, c1);  }  case 5:  {  StringBuilder input1 = new StringBuilder();  input1.append(symbol);  input1.reverse();  return input1.toString();  }  default:  {  System.out.println("Invalid choice");  return "0";  }  }  }  public StringOperationImpl()  {  super();  }  } |

**3) StringOperationServer.java**

import org.omg.CosNaming.\*;

import org.omg.CORBA.\*;

import SimpleString.\*;

public class StringOperationServer

{

public static void main(String args[])

{

try

{

ORB orb=ORB.init(args,null);

StringOperationImpl stringOperationImpl = new StringOperationImpl();

orb.connect(stringOperationImpl);

org.omg.CORBA.Object objRef=orb.resolve\_initial\_references("NameService");

NamingContext ncRef=NamingContextHelper.narrow(objRef);

NameComponent nc=new NameComponent("NASDAQ","");

NameComponent path[]={nc};

ncRef.rebind(path,stringOperationImpl);

System.out.println("Server is ready");

Thread.currentThread().join();

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

**4) StringOperationClient.java**

import org.omg.CosNaming.\*;

import org.omg.CORBA.\*;

import SimpleString.\*;

import java.io.DataInputStream;

public class StringOperationClient

{

public static void main(String args[])

{

try

{

DataInputStream din=new DataInputStream(System.in);

ORB orb=ORB.init(args,null);

NamingContext ncRef=NamingContextHelper.narrow(orb.resolve\_initial\_references("NameService"));

NameComponent path[]={new NameComponent("NASDAQ","")};

StringOperation market=StringOperationHelper.narrow(ncRef.resolve(path));

System.out.println("\nEnter the String : ");

String str =din.readLine();

System.out.println("\nResult : " +market.get\_String(str));

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

OUTPUT:







