Department of Computer Science & Engineering

IT1404 – Middleware Technologies Laboratory

Prepared by

**J. Noorul Ameen M.E.,**

(Assistant Professor/CSE)



**E.G.S Pillay Engineering College**

Nagapattinam - 611001

**MIDDLEWARE LABORATORY – LAB MANUAL**

**EX.NO:1 Create a Distributed Application to Download Various Files from Various Server Using Remote Method Invocation**

**Aim:**

To create a distributed application to download various files from various location using Remote Method Invocation.

**Algorithm:**

1. To create an integer interface program to download file throws remote exception & call function Rep ().

2. Create the implementation program using to implement the interface.

3. Create com for file impl& assign the value & call super ().

4. Read buffer length.

5. Close the buffer

6. Print the name using function rep () otherwise catch the exception

7. Create the program for file server

8. Refind the file server with ip address192.2.168.0.3 otherwise catch the exception to get message.

9. Create the program for file client

10. Check if arg length! =print if proper arg.

11. Read name file data the no of times send.

12. Call the function rep ()

13. Stop

**EX.NO:1**

**DATE :**

**CREATE A DISTRIBUTED APPLICATION TO DOWNLOAD VARIOUS Files from various serverusing REMOTE METHOD INVOCATION**

**“AddServerIntf.java”**

1st File name “AddServerIntf.java” – defines remote interface provided by the server.

importjava.rmi.\*;

public interface AddServerIntf extends Remote

{

double add(double d1,double d2)throws RemoteException;

}

**“AddServerImpl.java”**

2nd File name “AddServerImpl.java” – implements remote interface.

importjava.rmi.\*;

importjava.rmi.server.\*;

public class AddServerImpl extends UnicastRemoteObject implements AddServerIntf

{

publicAddServerImpl()throws RemoteException

{

}

public double add(double d1,double d2)throws RemoteException

{

return d1+d2;

}

}

**“ AddServer.java”**

3rd File name ” AddServer.java” – Main program for the server machine.

import java.net.\*;

importjava.rmi.\*;

public class AddServer

{

public static void main(String args[])

{

try

{

AddServerImpladdServerImpl=new AddServerImpl();

Naming.rebind("AddServer",addServerImpl);

}

catch(Exception e)

{

}

}

}

**“ AddClient”**

4th File name “ AddClient” – Client side of the distributed applications.

importjava.rmi.\*;

public class AddClient

{

public static void main(String args[])

{

try

{

String addServerURL="rmi://"+args[0]+"/AddServer";

AddServerIntfaddServerIntf=(AddServerIntf)Naming.lookup(addServerURL);

System.out.println("The First Number is:"+args[1]);

double d1=Double.valueOf(args[1]).doubleValue();

System.out.println("The Second Number is:"+args[2]);

double d2=Double.valueOf(args[2]).doubleValue();

System.out.println("The sum is "+addServerIntf.add(d1,d2));

}

catch(Exception e)

{

System.out.println("Exception: "+e);

}

}

}

**STEP 2:**

Before going to second step U must compile all the 4 programs.

Generate Stub by,

COMMAND

rmicAddServerImpl

This command generate the AddServerImpl\_Stub.class

**Step 3:**

Install files on client and server machines

copy the compiled file to client and server directory or machine

**CLIENT:**

1.AddClient.class

2.AddServerImpl\_Stub.class

3.AddServerIntf.class

**SERVER:**

1. AddServerIntf.class
2. AddServerImpl\_Stub.class
3. AddServerImpl.class
4. AddServer.class

**STEP 4:**

Start the rmi registry on the server machine by

startrmiegistry

**STEP 5:**

server side machine or folders by,

javaAddServer

**STEP 6:**

Client side machine or folder by,

javaAddClientserverip 8 9

javaAddClient 127.0.0.1 8 9

**OUTPUT IS:**

The first number is: 8

the second number is: 9

The sum is: 17

**Result:**

Thus the program for RMI can be used to download various file from various servers &output is verified.

**EX.NO:2 Create a Java Bean to Draw Various Graphical Shapes & Display it Using BDK or Without Using BDK**

**Aim:**

To create a java bean to draw various graphical shapes& display it using Bean Development Kit.

**Algorithm:**

1. Start the program.

2. Read the variable

3. Assign the default constructor to set the size(100,200) & set to false.

4. Constructor with one parameter & repaint() can be called.

5. Get shape ()is used to return the shape.

6. Check if shape() draw rectangle() &arc.

7. Stop.

**EX.NO:2**

**DATE :**

**CREATE A JAVA BEAN TO DRAW VARIOUS GRAPHICAL SHAPES AND DISPLAY IT USING OR WITHOUT USING BDK**

import java.io.\*;

importjava.awt.\*;

public class shape extends Canvas implements Serializable

{

privateboolean shape;

public shape()

{

setSize(new Dimension(100,200));

shape=false;

}

public void setShape(boolean a)

{

this.shape=a;

repaint();

}

publicbooleangetShape()

{

return shape;

}

public void paint(Graphics g)

{

Dimension d=getSize();

if(shape)

{

g.drawRect(100,100,100,100);

g.drawRect(120,70,80,100);

g.drawRoundRect(10,120,200,220,20,20);

}

else

{

g.drawOval(50,50,70,70);

g.drawOval(130,20,85,60);

g.drawArc(200,80,80,80,0,180);

}

}

}

**Execution Commands:**

F:\MWT LabPgms\JavaBean> set PATH=%PATH%.; F:\MWT LabPgms\JavaBean

F:\MWT LabPgms\JavaBean> set CLASSPATH=%CLASSPATH%.; F:\MWT LabPgms\JavaBean

F:\MWT LabPgms\JavaBean>javac shape.java

F:\MWT LabPgms\JavaBean>jar cf shape.jar \*.class

Search and Run the following run.bat file from Bean Development Kit

D:\beans\beanbox\run.bat

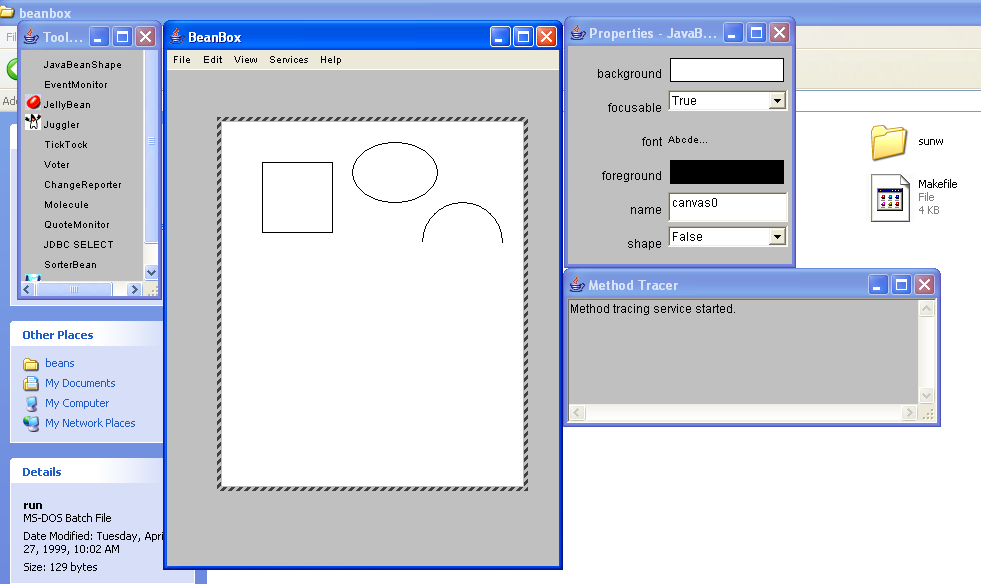
Load the jar file by using the File Menu

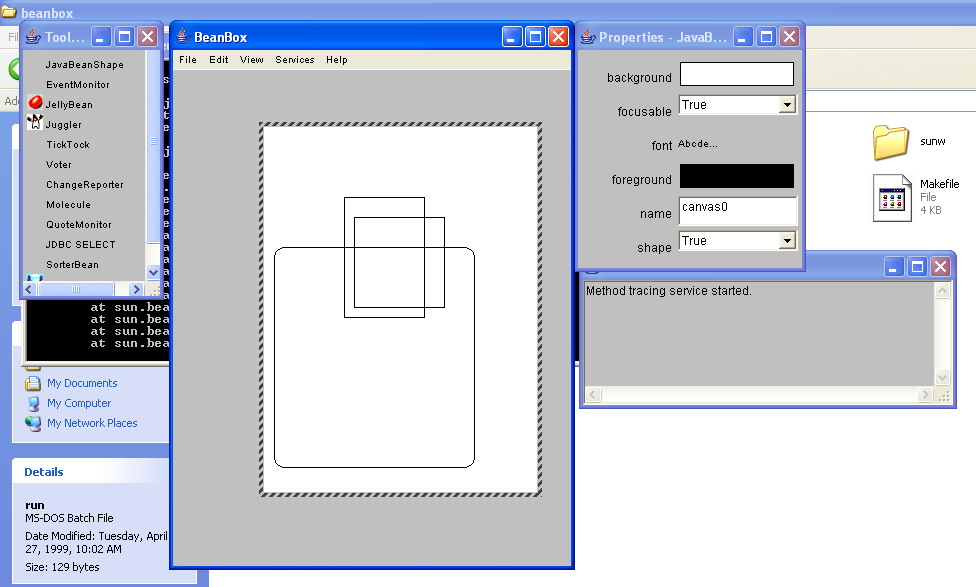
File>LoadJar> Choose shape.jar file location and load it

Now we can view the shape on ToolBox window, Draw and Drop it into BeanBox window

and Draw different shapes.

**OUTPUT:**





**Result:**

Thus the program has to be used to draw various shapes using bean box.

**EX.NO:3Develop an Enterprise Java Bean for Banking Operations**

**Aim:**

To develop an Enterprise Java Bean for Banking operation.

**Algorithm:**

1. Start the program

2. Define an interface Bankhome that contain create() of Bank Remote Interface.

3. Under Bankbean class define ejbcreate() to create connection to the database table Ace table

4. Create a class named Bankclient with choice for deposit,withdraw,Balance& exit

5.if choice=1

Call deposit function

if choice=2

Call withdraw function

if choice=3

Call setbalace function

if choice=4

exit from the program.

6.stop the program.

**EX.NO:3**

**DATE :**

**Develop an Enterprise Java Bean for Banking Operations**

**BankHome.java**

import java.rmi.RemoteException;

import javax.ejb.CreateException;

import javax.ejb.EJBHome;

public interface BankHome extends EJBHome

{

public BankRemote create() throws RemoteException, CreateException;

}

**BankRemote.java**

import java.rmi.RemoteException;

import javax.ejb.CreateException;

import javax.ejb.EJBObject;

public interface BankRemote extends EJBObject

{

public void deposit(float amt) throws RemoteException, CreateException;

public void withDraw(float amt) throws RemoteException, CreateException;

public String getBalance() throws RemoteException, CreateException;

}

**BankBean.java**

import java.rmi.RemoteException;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.ejb.EJBException;

import javax.ejb.SessionBean;

import javax.ejb.SessionContext;

import javax.security.auth.login.AccountException;

public class BankBean implements SessionBean

{

public float amount;

public String pwd, accNum, name, sql;

private Connection con;

private Statement stat;

private ResultSetrs;

SessionContextsesCtx;

public void ejbCreate()

{

System.out.println("ejbCreate Method Called...");

try {

con = new DatabaseAdaptor().getConnection();

stat = con.createStatement();

sql = "Select balance from AccTable where accountNo="+accNum+"";

System.out.println(sql);

rs = stat.executeQuery(sql);

if(rs.next())

{

amount = rs.getFloat(1);

}

}

catch (Exception e) {

e.printStackTrace();

}

}

public float getBalance()

{

return amount;

}

public void deposit(float amt)

{

amount = amount + amt;

}

public void withDraw(float amt)throws AccountException

{

if(amount <amt) throw new AccountException("Insufficient Balance!...");

if((amount - amt)<500) throw new AccountException("Insufficient Balance!...");

amount = amount - amt;

}

public void setSessionContext(SessionContextsesCtx) throws EJBException, RemoteException

{

this.sesCtx = sesCtx;

}

public void ejbRemove() throws EJBException, RemoteException

{

}

public void ejbActivate() throws EJBException, RemoteException

{

}

public void ejbPassivate() throws EJBException, RemoteException

{

}

}

**BankClient.java**

import java.io.DataInputStream;

import java.io.IOException;

import java.rmi.RemoteException;

import java.util.Hashtable;

import java.util.Properties;

import javax.ejb.CreateException;

import javax.ejb.RemoveException;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

public class BankClient

{

static String user = null;

static String password = null;

static String url = "t3://localhost:7001";

public static void main(String args[ ]) throws IOException

{

Hashtablehashtable = new Hashtable();

intbal, amt, ch;

String name, accNum, pwd;

DataInputStream dis = new DataInputStream(System.in);

try {

Context ctx = getInitialContext();

BankHomebankHome = (BankHome)ctx.lookup("beep");

BankRemotebankRemote = bankHome.create();

do {

System.out.println("1.Deposit\n 2.Withdraw\n 3.Balance\n 4.Exit");

System.out.println("Enter your choice:");

ch = Integer.parseInt(dis.readLine());

switch(ch)

{

case 1:

System.out.println("Enter Deposit Amount:");

bankRemote.deposit(java.lang.Float.parseFloat(dis.readLine()));

System.out.println("Your Money has been Deposited Successfully:");

break;

case 2:

System.out.println("Enter Withdraw Amount:");

try {

bankRemote.withDraw(java.lang.Float.parseFloat(dis.readLine()));

}

catch (Exception e) {

e.printStackTrace();

}

System.out.println("Your Money has been Debited Successfully:");

break;

case 3:

String curBal = bankRemote.getBalance();

System.out.println("Current Balance is:"+curBal);

break;

case 4:

bankRemote.remove();

System.exit(0);

default:

System.out.println("Wrong Choice:");

}

}while(ch!=4);

}

catch (RemoteException re) {

re.printStackTrace();

}

catch (CreateExceptionce) {

ce.printStackTrace();

}

catch (RemoveException rem){

rem.printStackTrace();

}

catch (NamingException ne) {

ne.printStackTrace();

}

catch (Exception e) {

e.printStackTrace();

}

}

static public Context getInitialContext()

throws Exception {

Properties properties = new Properties();

properties.put(Context.INITIAL\_CONTEXT\_FACTORY,"weblogic.jndi.T3InitialContextFactory");

properties.put(Context.PROVIDER\_URL,url);

if(user!=null)

{

System.out.println("user:"+user);

properties.put(Context.SECURITY\_PRINCIPAL,user);

if(password==null)password="";

properties.put(Context.SECURITY\_CREDENTIALS,password);

}

return new InitialContext(properties);

}

}

**DatabaseAdaptor.java**

import java.sql.Connection;

import java.sql.DriverManager;

public class DatabaseAdaptor

{

private static String DRIVER, URL\_DSN;

private static booleanisDriverLoaded = false;

public DatabaseAdaptor()

{

DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver";

URL\_DSN = "jdbc:BankDSN";

}

public Connection getConnection()

{

Connection con = null;

try {

if(!isDriverLoaded)

{

Class.forName(DRIVER);

isDriverLoaded = true;

}

con = DriverManager.getConnection(URL\_DSN);

}

catch (Exception e) {

e.printStackTrace();

}

return con;

}

}

**Compilation and Execution Commands:**

**Server Path:**

E:\Ejb\Bank >path= C:\Program Files\Java\jdk1.5.0\_02\bin;

E:\Ejb\Bank >set classpath= C:\Program Files\Java\jdk1.5.0\_02\bin;

E:\Ejb\Bank >set classpath=%classpath%;

E:\Ejb\Bank >javac \*.java

E:\Ejb\Bank >C:\bea\weblogic81\server\bin\setWLSEnv;

**Jar File Creation:**

E:\Ejb\Bank >jar –cvf bankoper.jar \*.class

**Client Path:**

E:\Ejb\Bank > path= C:\Program Files\Java\jdk1.5.0\_02\bin;

E:\Ejb\Bank >set classpath= C:\Program Files\Java\jdk1.5.0\_02\lib\jconsole.jar;

E:\Ejb\Bank > set classpath=%classpath%;

E:\Ejb\Bank > set classpath=%classpath%;C:\bea\weblogic81\server\lib\weblogic.jar;

E:\Ejb\Bank > C:\bea\weblogic81\server\bin\setWLSEnv;

E:\Ejb\Bank >java BankClient

**OUTPUT:**

1.Deposit

2.Withdraw

3.Balance

4.Exit

Enter Your Choice:

1

Enter Deposit Amount

5000.00

Successfully Deposited Thank You…..

1.Deposit

2.Withdraw

3.Balance

4.Exit

Enter Your Choice:

3

Your Current Balance is: 5000.00

1.Deposit

2.Withdraw

3.Balance

4.Exit

Enter Your Choice:

2

Enter Withdraw Amount

2000.00

Successfully Withdraw Thank You….

1.Deposit

2.Withdraw

3.Balance

4.Exit

Enter Your Choice:

3

Your Current Balance is: 3000.00

1.Deposit

2.Withdraw

3.Balance

4.Exit

Enter Your Choice:

2

Enter Withdraw Amount

4000.00

AccountException: Insufficient Balance!!!.....

1.Deposit

2.Withdraw

3.Balance

4.Exit

Enter Your Choice:

4

Exit

**Result:**

Thus the program for EJB program has been executed successfully.

**EX.NO:4Develop an Enterprise Java Bean for Library Operations**

**Aim:**

To develop an Enterprise Java Bean for Library operations.

**Algorithm:**

1. Start the program

2. Define an interface Libraryhome which extend EJB home & contain create() method.

3. Create the interface Library remote which extend EJB object & contain issue,receive,copy methods.

4. Create the class Library which implements session bean & gives definition for create issue & receive method.

5.In library client class, through buffered reader get the input values & declare variable for Library home & library remote.

6. Get theoptions & check the cases namely issue,receive& exit.

7. Build & execute by setting appropriate paths.

**EX.NO:4**

**DATE :**

**Develop an Enterprise Java Bean for Library Operations**

**LibHome.java**

import java.rmi.RemoteException;

import javax.ejb.CreateException;

import javax.ejb.EJBHome;

public interface LibHome extends EJBHome

{

public LibRemote create(int id, String title, String author, intnc) throws

RemoteException, CreateException;

}

**LibRemote.java**

import java.rmi.RemoteException;

import javax.ejb.EJBObject;

public interface LibRemote extends EJBObject

{

public boolean issue(int id, String title, String author, intnc)throws

RemoteException;

public boolean receive(int id, String title, String author, intnc)throws

RemoteException;

public intncpy() throws RemoteException;

}

**LibraryBean.java**

import java.rmi.RemoteException;

import javax.ejb.EJBException;

import javax.ejb.SessionBean;

import javax.ejb.SessionContext;

public class LibraryBean implements SessionBean

{

intbkid;

String tit;

String auth;

int nc1;

boolean status=false;

public void ejbCreate(intid,Stringtitle,Stringauthor,intnc)

{

bkid=id;

tit=title;

auth=author;

nc1=nc;

}

public intncpy()

{

return nc1;

}

public boolean issue(intid,Stringtit,Stringauth,intnc)

{

if(bkid==id)

{

nc1--;

status=true;

}

else

status=false;

return(status);

}

public boolean receive(intid,Stringtit,Stringauth,intnc)

{

if(bkid==id)

{

nc1++;

status=true;

}

else

status=false;

return(status);

}

public void ejbActivate() throws EJBException, RemoteException

{

}

public void ejbPassivate() throws EJBException, RemoteException

{

}

public void ejbRemove() throws EJBException, RemoteException

{

}

public void setSessionContext(SessionContextsesCxt) throws EJBException,

RemoteException

{

}

}

**LibClient.java**

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.Properties;

import javax.naming.InitialContext;

import javax.rmi.PortableRemoteObject;

public class LibClient

{

public static void main(String[ ]args) throws Exception

{

Properties props = new Properties();

props.setProperty(InitialContext.INITIAL\_CONTEXT\_FACTORY, "weblogic.jndi.WLInitialContextFactory");

props.setProperty(InitialContext.PROVIDER\_URL, "t3://localhost:7001");

props.setProperty(InitialContext.SECURITY\_PRINCIPAL, "");

props.setProperty(InitialContext.SECURITY\_CREDENTIALS, "");

InitialContextinitialContext = new InitialContext(props);

Object objRef = initialContext.lookup("library2");

LibHomelibHome = (LibHome) PortableRemoteObject.narrow

(objRef, LibHome.class);

BufferedReaderbr = new BufferedReader(new InputStreamReader(System.in));

intch;

String tit, auth;

int id, nc;

System.out.println("Enter the Details:");

System.out.println("Enter the Account Number:");

id = Integer.parseInt(br.readLine());

System.out.println("Enter the Book Title:");

tit = br.readLine();

System.out.println("Enter the Author:");

auth = br.readLine();

nc = Integer.parseInt(br.readLine());

int temp = nc;

do {

System.out.println("\t\t LIBRARY OPERATIONS:");

System.out.println("\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("");

System.out.println("\t\t 1.ISSUE");

System.out.println("\t\t 2.RECEIVE");

System.out.println("\t\t 3.EXIT");

System.out.println("\t\t 4.ENTER UR OPTION:");

ch = Integer.parseInt(br.readLine());

LibRemotelibRemote = libHome.create(id, tit, auth, nc);

switch(ch)

{

case 1:

System.out.println("Entering:");

nc = libRemote.ncpy();

if(nc>0)

{

if(libRemote.issue(id, tit, auth, nc))

{

System.out.println("BOOK ID IS:"+id);

System.out.println("BOOK TITLE IS:"+tit);

System.out.println("BOOK AUTHOR IS:"+auth);

System.out.println("NO.OFCOPIES:"+libRemote.ncpy());

nc = libRemote.ncpy();

System.out.println("Success:");

break;

}

}

else

System.out.println("Book is not available:");

break;

case 2:

System.out.println("Entering:");

if(temp>nc)

{

System.out.println("Temp:"+temp);

}

if(libRemote.receive(id, tit, auth, nc))

{

System.out.println("BOOK ID IS:"+id);

System.out.println("BOOK TITLE IS:"+tit);

System.out.println("BOOK AUTHOR IS:"+auth);

System.out.println("NO.OF COPIES:"+libRemote.ncpy());

nc = libRemote.ncpy();

System.out.println("Success:");

break;

}

else

System.out.println("Invalid Transaction:");

break;

case 3:

System.exit(0);

}

}

while (ch<=3 &&ch>0);

}

}

**Compilation and Execution Commands:**

**Server Path:**

E:\Ejb\Lib>path=C:\Program Files\Java\jdk1.5.0\_02\bin;

E:\Ejb\Lib>set classpath=C:\Program Files\Java\jdk1.5.0\_02\bin;

E:\Ejb\Lib>set classpath=%classpath%;

E:\Ejb\Lib>javac \*.java

E:\Ejb\Lib>C:\bea\weblogic81\server\bin\setWLSEnv;

**Jar File Creation:**

E:\Ejb\Lib>jar –cvf liboper.jar \*.class

**Client Path:**

E:\Ejb\Lib> path=C:\Program Files\Java\jdk1.5.0\_02\bin;

E:\Ejb\Lib>set classpath= C:\Program Files\Java\jdk1.5.0\_02\lib\jconsole.jar;

E:\Ejb\Lib>set classpath=%classpath%;

E:\Ejb\Lib>set classpath=%classpath%;C:\bea\weblogic81\server\lib\weblogic.jar;

E:\Ejb\Lib>C:\bea\weblogic81\server\bin\setWLSEnv;

E:\Ejb\Lib>java LibClient

**OUTPUT:**

Enter the Details

Enter the Account Number: 101

Enter the Book Title: Core Java

Enter the Author: BalaguruSamy

Enter the no.of.copies: 2

**LIBRARY OPERATIONS**

1.ISSUE

2.RECEIVE

3.EXIT

ENTER YOUR OPTION: 2

Entering Invalid Transaction

**LIBRARY OPERATIONS**

1.ISSUE

2.RECEIVE

3.EXIT

ENTER YOUR OPTION: 1

Entering valid Transaction

BOOK ID IS: 101

BOOK TITLE IS: Core Java

BOOK AUTHOR IS: BalaguruSamy

NO.OF.COPIES: 1

Success

**LIBRARY OPERATIONS**

1.ISSUE

2.RECEIVE

3.EXIT

ENTER YOUR OPTION: 2

BOOK ID IS: 101

BOOK TITLE IS: Core Java

BOOK AUTHOR IS: BalaguruSamy

NO.OF.COPIES: 2

Success

**LIBRARY OPERATIONS**

1.ISSUE

2.RECEIVE

3.EXIT

ENTER YOUR OPTION: 3

E:\Ejb\Lib>

**Result:**

Thus the EJB program for Library operation is executed& the output is verified

**EX.NO: 5 Active-X Control for File Operations**

**Aim:**

To create an Active-x control for file operation.

**Algorithm:**

1. Start the program

2.In net workspace select window class library then control design with appear.

3. Select menuship& place it in the user control

4. Include the file menu containing new,open,createfile,deletefile,create directory & delete directory.

5.Add rich text boe to menux trips the place to user control in the form

6. Write the appropriate coding for each menu

7. Build & execute by setting appropriate paths.

8. Stop the program.

**EX.NO:5**

**DATE :**

**Create an Active-X control for File Operations**

Dim f As New FileSystemObject

Dim t As TextStream

Dim s1 As String

Public Sub filecreation(s As String)

On Error GoTo X

s1 = s

If (s1 = "") Then

MsgBox ("Please enter the filename")

Else

If (f.FileExists(s1) = False) Then

f.CreateTextFile (s1)

MsgBox ("File Created")

Else

MsgBox ("File Already created")

End If

End If

Exit Sub

X:

MsgBox ("Error in File Creation")

End Sub

Public Sub filewrite(a As String)

On Error GoTo X

If (s1 = "") Then

MsgBox ("Please enter the filename")

Else

If (f.FileExists(s1) = True) Then

Set t = f.OpenTextFile(s1, ForWriting, True)

t.WriteLine (a)

MsgBox ("File Written")

t.Close

Else

MsgBox ("File Doesnt Exist")

End If

End If

Exit Sub

X:

MsgBox ("Error in File Writing")

End Sub

Public Sub fileappend(a As String)

On Error GoTo X

If (s1 = "") Then

MsgBox ("Please enter the filename")

Else

If (f.FileExists(s1) = True) Then

Set t = f.OpenTextFile(s1, ForAppending, True)

t.WriteLine (a)

MsgBox ("File Appended")

t.Close

Else

MsgBox ("File not Exist")

End If

End If

Exit Sub

X:

MsgBox ("Error in File Appending")

End Sub

Public Function fileread(s) As String

On Error GoTo X

Dim k As String

If (s = "") Then

MsgBox ("Please enter the filename")

Else

If (f.FileExists(s) = True) Then

Set t = f.OpenTextFile(s, ForReading, True)

While (t.AtEndOfStream = False)

k = k & " " &t.ReadLine

Wend

fileread = k

t.Close

Else

MsgBox ("File Not Exist")

End If

End If

Exit Function

X:

MsgBox ("Error in Reading")

End Function

Public Sub filedelete(s)

On Error GoTo X

If (s = "") Then

MsgBox ("Please enter the filename")

Else

If (f.FileExists(s) = True) Then

f.DeleteFile (s)

MsgBox ("File Deleted")

Else

MsgBox ("File Doesnt Exist")

End If

End If

Exit Sub

X:

MsgBox ("Error in File Deletion")

End Sub

**// Standard Project (with Three textboxes and six command buttons)**

Dim t As New FileActivex.Class1

Private Sub Command1\_Click()

t.filecreation (Trim(Text3.Text))

End Sub

Private Sub Command2\_Click()

t.filewrite (Trim(Text1.Text))

End Sub

Private Sub Command3\_Click()

t.fileappend (Trim(Text1.Text))

End Sub

Private Sub Command4\_Click()

Text2.Text = t.fileread(Trim(Text3.Text))

End Sub

Private Sub Command5\_Click()

t.filedelete (Trim(Text3.Text))

End Sub

Private Sub Command6\_Click()

End

End Sub

**EXECUTION:**

**// ActivexDLL Project**

1. Open "activexDll" project in VB

2. Give the reference to "c:\windows\system32\scrrun.dll” (for FileSystemObject)

Project 🡪Reference 🡪 (check) Microsoft scripting Runtime

3. Type the coding for File operation

4. Make the dll file

File 🡪"Make Project1.dll"

**// Standard Project**

1. Open a new standard application project in VB

2. Give the reference to the dll file (Project1.dll)

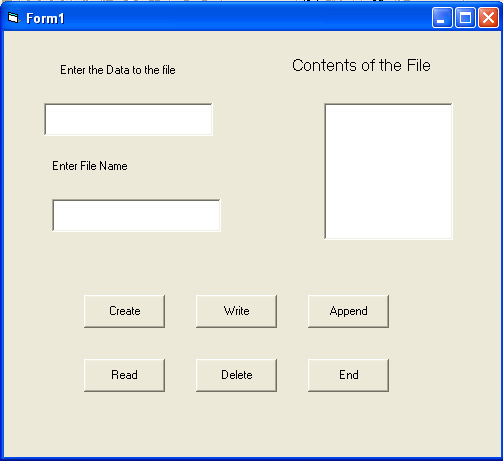
Project 🡪Reference 🡪Browse 🡪Select the project1.dll by navigate to the file location

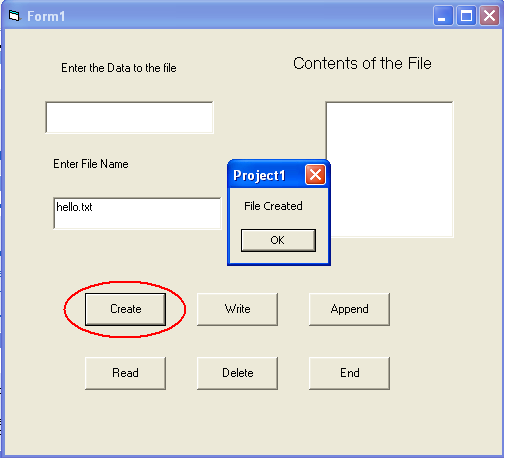
3. Design the VB Form

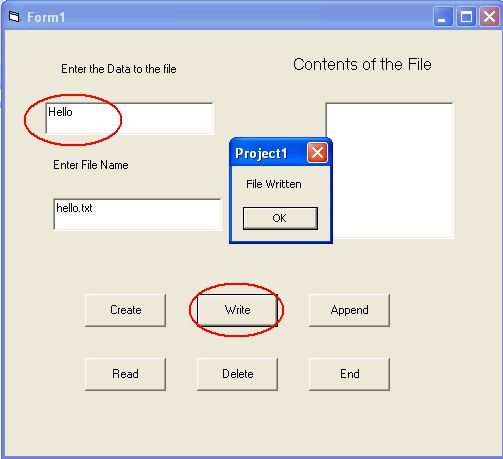
4. Type the coding

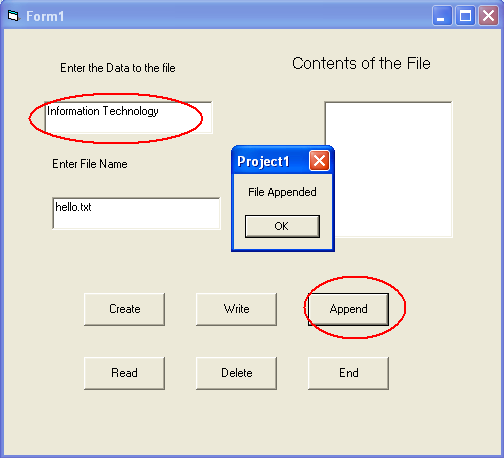
5. Execute the application

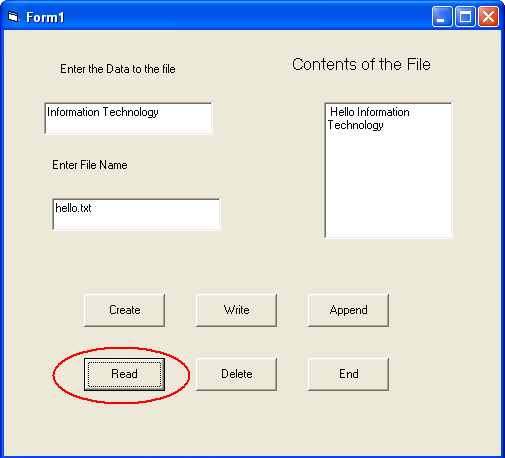
**OUTPUT:**

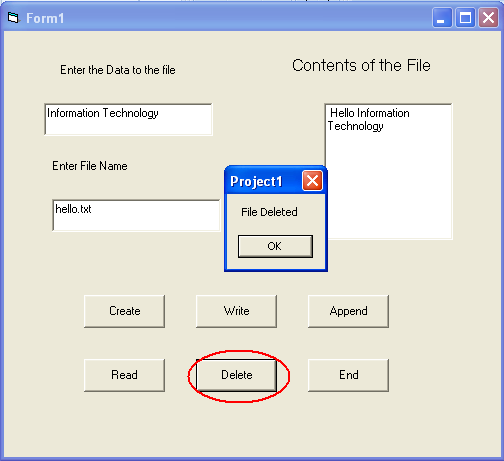
****

****

****







**Result:**

Thus the Active-x program for file operation is executed& the output is verified.

**EX.NO:6 Develop a component for converting the currency values using COM / .NET**

**Aim:**

To develop a component for converting the currency values using DCOM/ .Net.

**Algorithm:**

1. Start the program

2. Createform1 call button1 click.

i.if combobox1, selected item is ’Dollar’ & other combobox item is ‘Dollar’ assign textbox1.text with textbox2.text.

ii.if combobox1, selected item is ’Dollar’ & other combobox item is ‘Euro’ assign textbox1.text

iii.if combobox1, selected item is ’Dollar’,Euro,Yen& rupees other combobox 2 contains one among the above four write the code for calculation.

3. Stop the program.

**EX.NO:6**

**DATE :**

**Develop a component for Converting the Currency Values using COM/.NET**

Public Class Form1

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

Private Sub Label3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label3.Click

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

If ComboBox1.SelectedItem = "Dollar" And ComboBox2.SelectedItem = "Dollar" Then

TextBox2.Text = "$" & TextBox1.Text

ElseIf ComboBox1.SelectedItem = "Dollar" And ComboBox2.SelectedItem = "Euro" Then

TextBox2.Text = "E" & Val(TextBox1.Text) / 1.5

ElseIf ComboBox1.SelectedItem = "Dollar" And ComboBox2.selectedItem = "yen" Then

TextBox2.Text = "Y" & Val(TextBox1.Text) \* 135

ElseIf ComboBox1.SelectedItem = "Dollar" And ComboBox2.selectedItem = "Rupees" Then

TextBox2.Text = "Rs." & Val(TextBox1.Text) \* 45

End If

If ComboBox1.SelectedItem = "Euro" And ComboBox2.selectedItem = "Dollar" Then

TextBox2.Text = "$" & Val(TextBox1.Text) \* 1.5

ElseIf ComboBox1.SelectedItem = "Euro" And ComboBox2.SelectedItem = "Euro" Then

TextBox2.Text = "E" & TextBox1.Text

ElseIf ComboBox1.SelectedItem = "Euro" And ComboBox2.SelectedItem = "Yen" Then

TextBox2.Text = "Y" & Val(TextBox1.Text) \* 165

ElseIf ComboBox1.SelectedItem = "Euro" And ComboBox2.SelectedItem = "Rupees" Then

TextBox2.Text = "Rs." & Val(TextBox1.Text) \* 50

End If

If ComboBox1.SelectedItem = "Yen" And ComboBox2.SelectedItem = "Dollar" Then

TextBox2.Text = "$" & (TextBox1.Text) / 135

ElseIf ComboBox1.SelectedItem = "Yen" And ComboBox2.SelectedItem = "Euro" Then

TextBox2.Text = "E" & Val(TextBox1.Text) / 165

ElseIf ComboBox1.SelectedItem = "Yen" And ComboBox2.SelectedItem = "Yen" Then

TextBox2.Text = "Y" & TextBox1.Text

ElseIf ComboBox1.SelectedItem = "Yen" And ComboBox2.SelectedItem = "Rupees" Then

TextBox2.Text = "Rs." & Val(TextBox1.Text) / 30

End If

If ComboBox1.SelectedItem = "Rupees" And ComboBox2.SelectedItem = "Dollar" Then

TextBox2.Text = "$" & Val(TextBox1.Text) / 45

ElseIf ComboBox1.SelectedItem = "Rupees" And ComboBox2.SelectedItem = "Euro" Then

TextBox2.Text = "E" & Val(TextBox1.Text) / 50

ElseIf ComboBox1.SelectedItem = "Rupees" And ComboBox2.SelectedItem = "Yen" Then

TextBox2.Text = "Y" & Val(TextBox1.Text) \* 30

ElseIf ComboBox1.SelectedItem = "Rupees" And ComboBox2.SelectedItem = "Rupees" Then

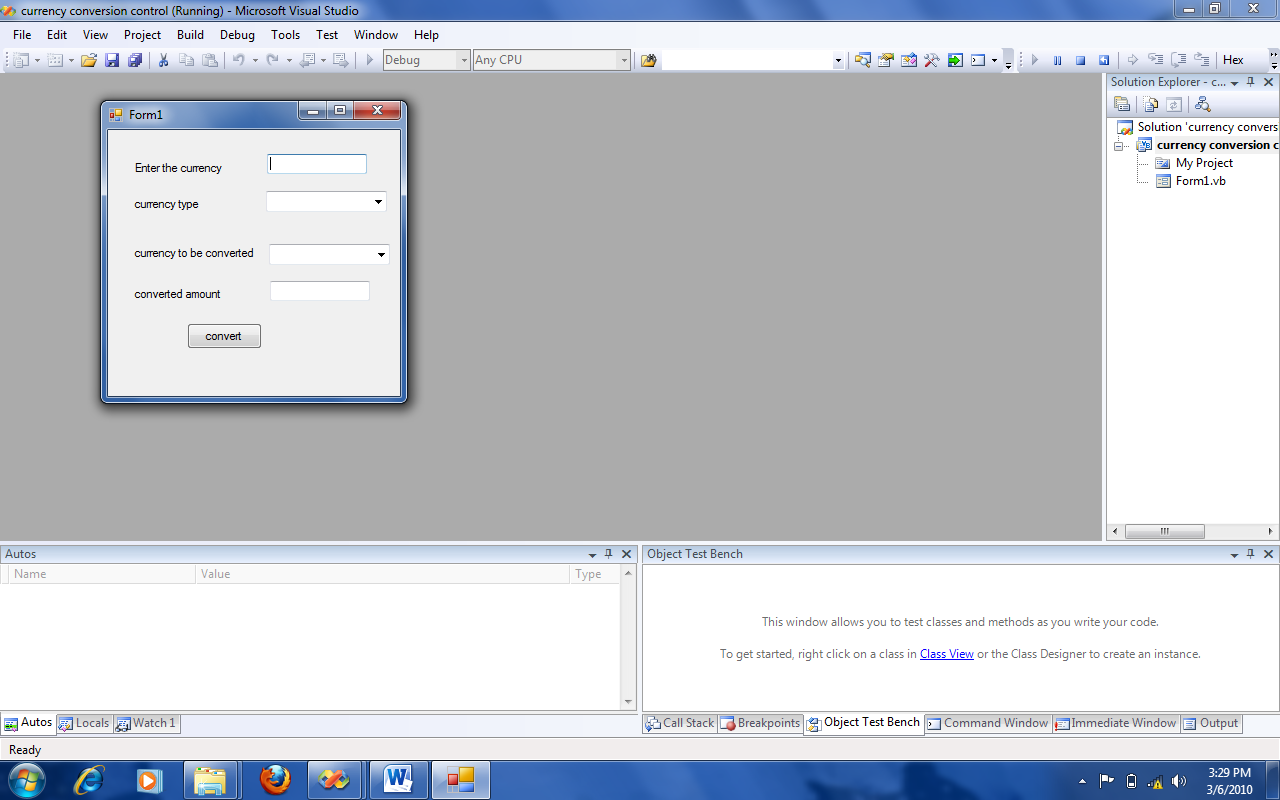
TextBox2.Text = "Rs." & TextBox1.Text

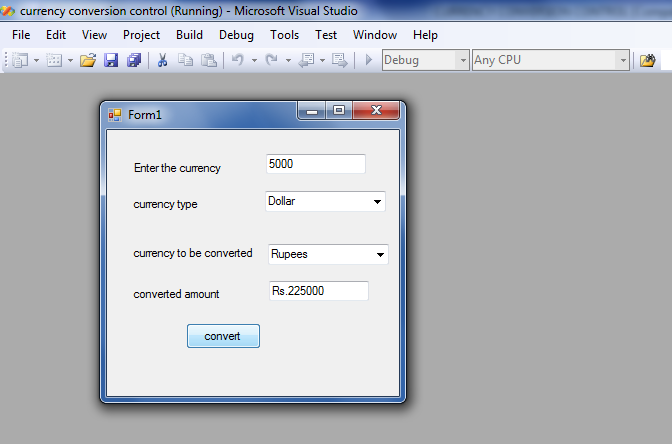
End If

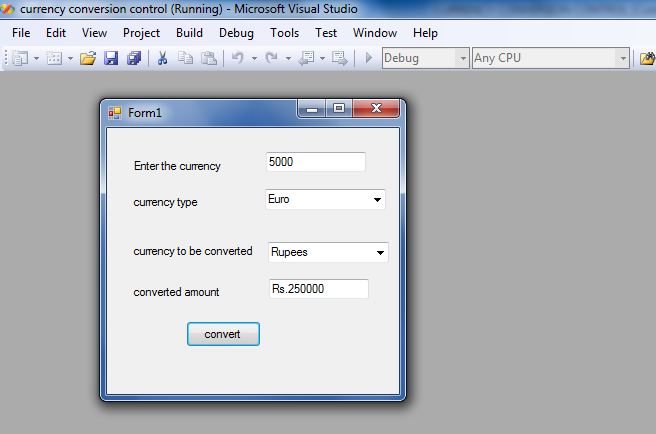
End Sub

End Class

**OUTPUT:**







**Result:**

Thus the program for currency conversion using .Net is executed& the output is verified.

**EX.NO:7 Develop a component for encryption and decryption using COM / .NET**

**Aim:**

Todevelop a component for encryption and decryption using COM/.Net.

**Algorithm:**

1. Start the program

2. Create form1 using ‘encrypr’ that convert text to string,assign to label1.

3. We encrypt the message using shared key shared iv

4. Using ‘decrypt’ function convert encrypt message to original plain text.

5. Using button-2 click that & assign as decrypt message to label1.

6. Stop the program.

**EX.NO:7**

**DATE :**

**Develop a component for Encryption and Decryption using COM/.NET**

Public Class Form1

Dim a As Char()

Dim otemp As Char

Dim i As Integer

Dim tem As Integer

Dim st As String

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

st = ""

a = TextBox2.Text.ToCharArray

For i = 0 To a.Length - 1

tem = Asc(a(i))

tem = tem - 5

otemp = Chr(tem)

st&= otemp

Next

TextBox3.Text = st

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

st = " "

a = TextBox1.Text.ToCharArray

i = a.length

For i = 0 To a.Length - 1

tem = Asc(a(i))

tem = tem + 5

otemp = Chr(tem)

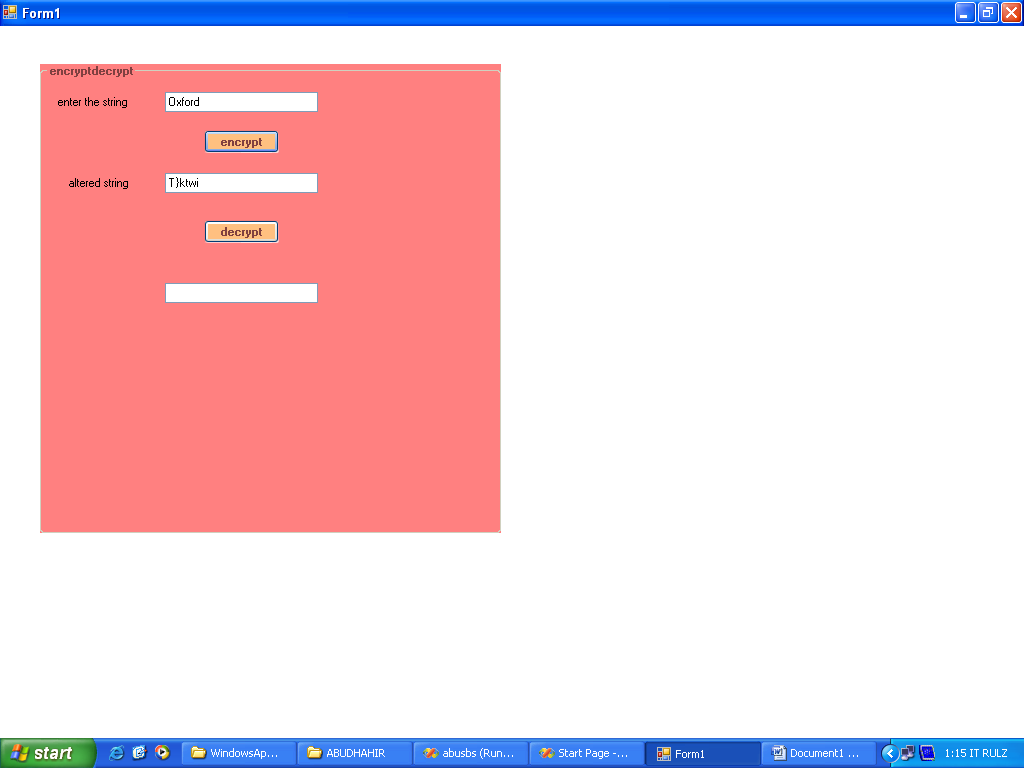
st&= otemp

Next

TextBox2.Text = st

End Sub

End Class



**Result:**

Thus the program for Encryption & Decryption is executed& the output is verified.

**EX.NO:8Develop a component for retrieving information from message box using DCOM / .NET**

**Aim:**

To develop a component for retrieving information from messagebox using DCOM/.Net.

**Algorithm:**

1. Start the program

2. Declare ‘g’ as string.

3.if ‘g’ is 6,then msgbox display yes.

4. if ‘g’ is 7,then msgbox display no.

5. if ‘g’ is 2,then msgbox display cancel.

6. Stop the program.

**EX.NO:8**

**DATE :**

**Develop a Component for Retrieving Information from MessageBox using DCOM/.NET**

Public Class Form1

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim a As String = MessageBox.Show("The calculations are complete","My Application",MessageBoxButtons.YesNoCancel,MessageBoxIcon.Asterisk)

If a = "6" Then

MsgBox("yes")

ElseIf a = "7" Then

MsgBox("no")

ElseIf a = "2" Then

MsgBox("Hai")

End If

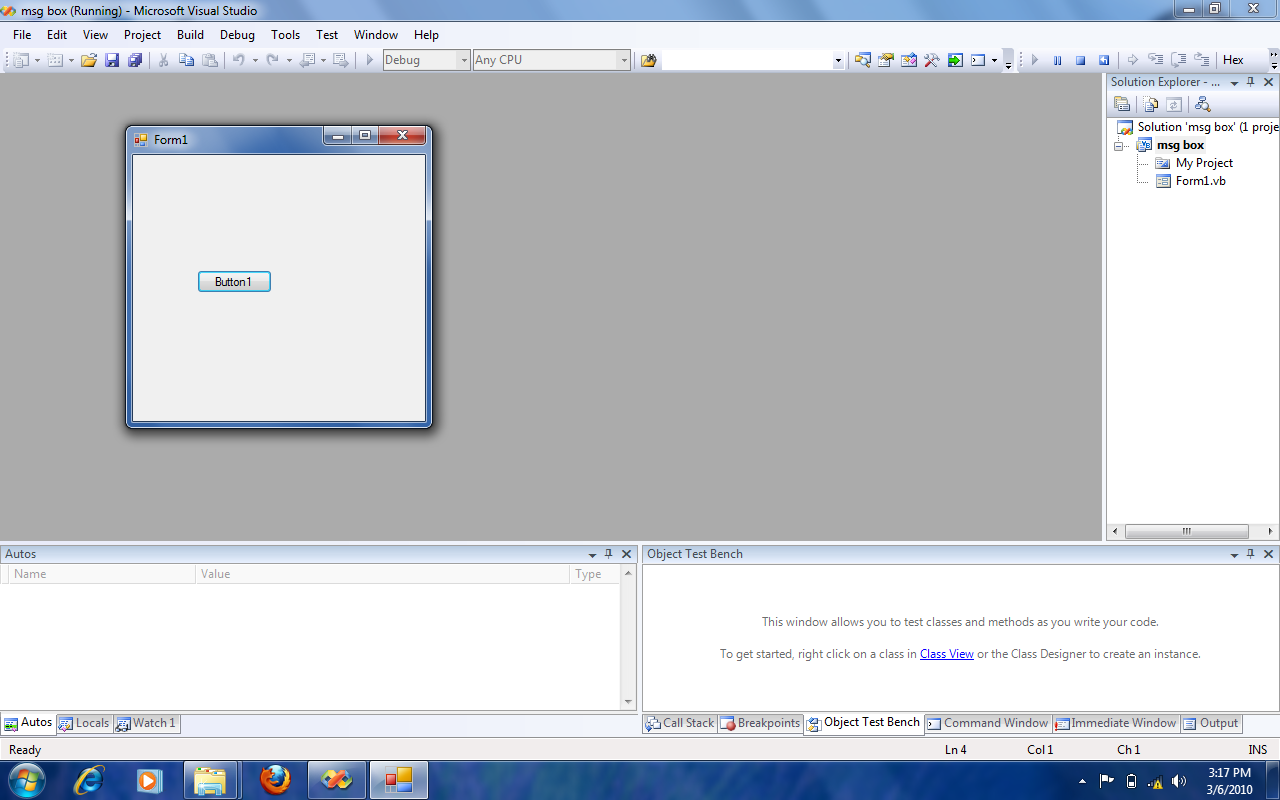
End Sub

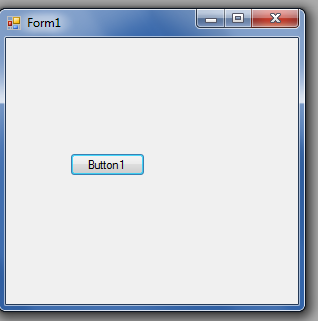
Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

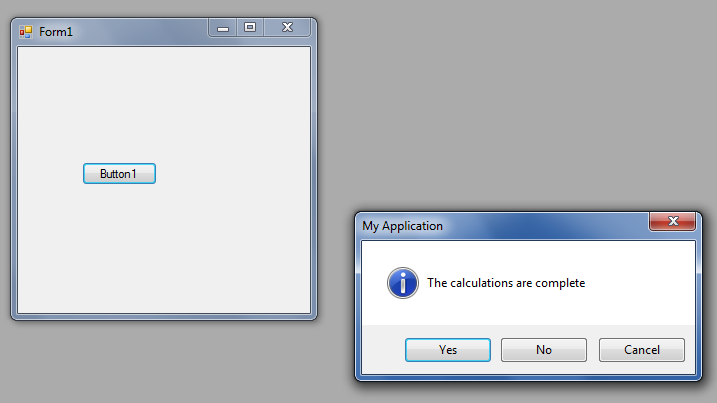
End Sub

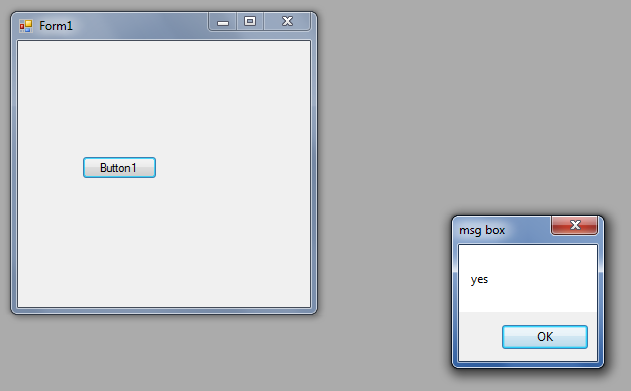
End Class

**OUTPUT:**









**Result:**

Thus the program for Message box using .Net is executed& the output is verified.

**EX.NO:9Develop a middleware component for retrieving Stock Market Exchange information usingCORBA**

**Aim:**

To develop a middleware component for retrieving stock market exchange using CORBA.

**Algorithm:**

1. Start the program

2. Create a class stock server & activate the specified function.

3. Create a class stock client & read choice such as buy,sell& display.

4. Perform appropriate operations based on the choice.

5. In stock interface,declare the input function.

6. Stop the program.

**EX.NO:9**

**DATE :**

**Develop a Middleware Component for Retrieving Stock Market Exchange Information using CORBA**

**Stock Server.java**

import StockApp.\*;

import org.omg.CosNaming.\*;

import org.omg.CORBA.\*;

import org.omg.CORBA.Object;

import org.omg.CosNaming.NamingContextPackage.\*;

import org.omg.PortableServer.\*;

import org.omg.PortableServer.POA;

class StockImpl extends StockPOA

{

private ORB orb;

public void setORB(ORB j)

{

orb = j;

}

public double input(double unit, double no)

{

return (unit \* no);

}

}

public class StockServer

{

public static void main(String[ ]args)

{

try {

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

POA rootPOA = POAHelper.narrow(orb.resolve\_initial\_references("RootPOA"));

rootPOA.the\_POAManager().activate();

StockImplstockImpl = new StockImpl();

stockImpl.setORB(orb);

Object objectRef1 = rootPOA.servant\_to\_reference(stockImpl);

Stock stockRef = StockHelper.narrow(objectRef1);

org.omg.CORBA.Object objectRef2 = orb.resolve\_initial\_references("NameService");

NamingContextExtncRef = NamingContextExtHelper.narrow(objectRef2);

String name = "P1";

NameComponentncPath[ ] = ncRef.to\_name(name);

ncRef.rebind(ncPath, stockRef);

System.out.println("Server is Ready & Waiting.....");

orb.run();

}

catch (Exception e)

{

System.out.println("ERROR:"+e);

e.printStackTrace();

}

}

}

**Stock Client.java**

import StockApp.\*;

import org.omg.CosNaming.\*;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import org.omg.CORBA.ORB;

import org.omg.CosNaming.NamingContextExt;

import org.omg.CosNaming.NamingContextExtHelper;

public class StockClient

{

static Stock stock;

public static void main(String args[ ])

{

String[ ] uid1 = { "M124", "p1190", "m890", "k345"};

String[ ] p = {"HUTCH", "Infosys","Hutch","Infosys"};

double [ ] unit = {5,6,5,6};

double [ ]noshares = {10,4,7,3};

double [ ] total = {0,0,0,0};

intch;

try {

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

org.omg.CORBA.ObjectobjectRef = orb.resolve\_initial\_references("NamingService");

NamingContextExtnamingContextRef = NamingContextExtHelper.narrow(objectRef);

String name = "p1";

stock = StockHelper.narrow(namingContextRef.resolve\_str(name));

for (int i=0; i<4; i++)

total[i] = stock.input(unit[i], noshares[i]);

System.out.println("STOCK MARKER EXCHANGE:");

System.out.println("----------------------");

System.out.println("STATUS OF NSE:");

System.out.println("------------");

System.out.println("Uid Company name \t unit\t No.of.Shares\t total:");

for (int i=0; i<4; i++)

{

System.out.println("\t"+uid1[i]+"\t"+p[i]+"\t"+unit[i]+"\t\t"+noshares[i]+"\t\t"+total[i]);

}

BufferedReaderbufferedReader = new BufferedReader(new InputStreamReader(System.in));

String choice = "y";

Do

{

System.out.println("\t\t Menu Option:");

System.out.println("\n 1.Buy, \n2.Sell, \n 3.Display");

System.out.println("Enter ur Choice:");

ch = Integer.parseInt(bufferedReader.readLine());

switch(ch)

{

case 1:

System.out.println("Enter Buyer ID:");

String str1 = bufferedReader.readLine();

System.out.println("Enter Company Name:");

String str2 = bufferedReader.readLine();

System.out.println("No of Shares u want to buy:");

double doub1 = Double.parseDouble(bufferedReader.readLine());

System.out.println("Enter the Seller id:");

String str3 = bufferedReader.readLine();

for (int i=0; i<4; i++)

{

if (uid1 [i].equals(str1))

{

if(p [i].equals(str2))

{

noshares[i] = noshares[i] + doub1;

total[i] = unit[i] \* noshares[i];

}

}

}

for (int i=0; i<4; i++)

{

if (uid1 [i].equals(str3))

{

if(p [i].equals(str2))

{

noshares[i] = noshares[i] - doub1;

total[i] = unit[i] \* noshares[i];

}

}

}

break;

case 2:

System.out.println("Enter Seller ID:");

String str4 = bufferedReader.readLine();

System.out.println("Enter Company Name:");

String str5 = bufferedReader.readLine();

System.out.println("No of Shares u want to buy:");

double doub2 = Double.parseDouble(bufferedReader.readLine());

System.out.println("Enter the Seller id:");

String str6 = bufferedReader.readLine();

for (int i=0; i<4; i++)

{

if (uid1 [i].equals(str4))

{

if(p [i].equals(str5))

{

noshares[i] = noshares[i] - doub2;

total[i] = unit[i] \* noshares[i];

}

}

}

for (int i=0; i<4; i++)

{

if (uid1 [i].equals(str6))

{

if(p [i].equals(str5))

{

noshares[i] = noshares[i] + doub2;

total[i] = unit[i] \* noshares[i];

}

}

}

break;

case 3:

System.out.println("\t\t Status of NSE:");

System.out.println("\t uid \t Companyname \t\t unit \t No.of.Share\t total:");

for (int i=0; i<4; i++)

{

System.out.println("\t"+uid1[i]+"\t"+p[i]+"\t"+unit[i]+"\t\t"+noshares[i]+"\t\t"+total[i]);

}

break;

}

System.out.println("Do you want to continue Y/N:");

choice = bufferedReader.readLine();

}

while (choice.equals("y"));

}

catch (Exception e) {

System.out.println("ERROR:"+e);

e.printStackTrace();

}

}

}

**Stock.idl**

module StockApp

{

interface Stock

{

double input (in double unit, in double rate);

};

};

**OUTPUT:**

E:\Corba>set PATH=%PATH%.; E:\Corba;

E:\Corba>set CLASSPATH=%CLASSPATH%.; E:\Corba;

E:\Corba>idlj Stock.idl

E:\Corba>idlj -fall Stock.idl

E:\Corba>javac \*.java StockApp/\*.java

Note: StockApp/StockPOA.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

E:\Corba>Start orbd -ORBInitialPort 1050

E:\Corba>Start java StockServer -ORBInitialPort 1050 -ORBInitialHostlocalhost

E:\Corba>Start java StockClient -ORBInitialPort 1050 -ORBInitialHostlocalhost

**STOCK MARKET EXCHANGE**

**STATUS OF NSE**

-------------------------------------------------------------------------------------

uid Company name unit No.of.shares total

-------------------------------------------------------------------------------------

M124 HUTCH 5.0 10.0 50.0

Pl190 Infosys 6.0 4.0 24.0

M890 Hutch 5.0 7.0 35.0

K345 Infosys 6.0 3.0 18.0

-------------------------------------------------------------------------------------

Menu Options

1. Buy
2. Sell
3. Display

Enter your choice: 1

Enter Buyer ID: M124

Enter Company Name: Hutch

No.of.shares you want to buy: 5

Enter the Seller ID: 1

Do you want to continue y/n: y

Menu Options

1. Buy
2. Sell
3. Display

Enter your choice: 3

**STATUS OF NSE**

-------------------------------------------------------------------------------------

uid Company name unit No.of.shares total

-------------------------------------------------------------------------------------

M124 HUTCH 5.0 15.0 75.0

Pl190 Infosys 6.0 4.0 24.0

M890 Hutch 5.0 7.0 35.0

K345 Infosys 6.0 3.0 18.0

-------------------------------------------------------------------------------------

Do you want to continue y/n: y

Menu Options

1. Buy
2. Sell
3. Display

Enter your choice: 2

Enter Seller ID: 2

Enter Company Name: Infosys

No.of.shares you want to buy: 5

Enter the Buyer ID: k345

Do you want to continue y/n: y

Menu Options

1. Buy
2. Sell
3. Display

Enter your choice: 3

**STATUS OF NSE**

-------------------------------------------------------------------------------------

uid Company name unit No.of.shares total

-------------------------------------------------------------------------------------

M124 HUTCH 5.0 15.0 75.0

Pl190 Infosys 6.0 4.0 24.0

M890 Hutch 5.0 7.0 35.0

K345 Infosys 6.0 8.0 48.0

-------------------------------------------------------------------------------------

Do you want to continue y/n: n

**Result:**

Thus the program for stock market exchange using CORBA is executed& the output is verified.

**EX.NO:10 Develop a middleware component for retrieving Weather Forecast information using CORBA**

**Aim:**

To develop a middleware component for weather forecast information using CORBA.

**Algorithm:**

1. Start the program

2. Create interface file ‘weather’.

3. Create server file ‘weatherserver’.

4.i. Create ORB object that call init method.

ii. Create object for weather impl& using that call set ORB ()

5. Define adjustemp(),getmax(),getmin(),call(),getcurrent(),shutdown().

6. Create weather client file.

7. Stop the program.

**EX.NO:10**

**DATE :**

**Develop a Middleware Component for Retrieving Weather Forecast Information using CORBA**

**WeatherServer.java**

import org.omg.CORBA.ORB;

import org.omg.CORBA.Object;

import org.omg.CosNaming.NameComponent;

import org.omg.CosNaming.NamingContextExt;

import org.omg.CosNaming.NamingContextExtHelper;

import org.omg.PortableServer.POA;

import org.omg.PortableServer.POAHelper;

public class WeatherServer

{

public static void main(String[ ] args)

{

try {

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

POA rootPOA = POAHelper.narrow(orb.resolve\_initial\_references("RootPOA"));

rootPOA.the\_POAManager().activate();

WeatherImplweatherImpl = new WeatherImpl();

weatherImpl.setORB(orb);

Object objectRef1 = rootPOA.servant\_to\_reference(stockImpl);

Weather weatherRef = WeatherHelper.narrow(ref);

org.omg.CORBA.Object objectRef2 = orb.resolve\_initial\_references("NameService");

NamingContextExtncRef = NamingContextExtHelper.narrow(objectRef2);

String name = "Hello";

NameComponentncPath[ ] = ncRef.to\_name(name);

ncRef.rebind(ncPath, weatherRef);

System.out.println("Server is Ready & Waiting.....");

orb.run();

}

catch (Exception e)

{

System.out.println("ERROR:"+e);

e.printStackTrace();

}

System.out.println("Server Exsisting:.....");

}

}

**WeatherImpl.java**

import org.omg.CORBA.ORB;

public class WeatherImpl extends WeatherPOA

{

private ORB orb;

private double current;

private double min;

private double max;

public voidsetORB(ORB orb\_val)

{

orb = orb\_val;

}

public voidadjustTemp(double g)

{

current+=g;

if(current> max)

max =current;

else if(current<min)

min = current;

}

public double getMax()

{

return max;

}

public double getMin()

{

return min;

}

public double getCurrent()

{

return current;

}

public void call1(double doub)

{

min = max = current = doub;

}

public void shutdown()

{

orb.shutdown(false);

}

}

**WeatherClient.java**

import org.omg.CORBA.ORB;

import org.omg.CosNaming.NamingContextExt;

import org.omg.CosNaming.NamingContextExtHelper;

public class WeatherClient

{

static Weather weather;

public static void main(String[ ] args)

{

double[ ][ ] d1 = {{-1.5,4.2,5.8,-4.5},{-2.3,-7.8,6.2,-4.7},

{4.3,-2.5,3.5,-6.7},{-5.2,4.3,8.2,6.2}};

String [ ] str = {"Chennai","Mumbai","Delhi","Calcutta"};

try {

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

org.omg.CORBA.ObjectobjectRef = orb.resolve\_initial\_references("NamingService");

NamingContextExtnamingContextRef = NamingContextExtHelper.narrow(objectRef);

String name = "Hello";

weather = WeatherHelper.narrow(namingContextRef.resolve\_str(name));

weather.call1(20.0);

System.out.println("\n\n");

System.out.println("\t\t WEATHER FORECASTING");

System.out.println("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for (int j=0; j<4; j++)

{

System.out.println("\n\n");

System.out.println("City:"+str[j]);

for (int i=0; i<3; i++)

{

weather.adjustTemp(d1[j][i]);

}

System.out.println("\t\t Maximum Temperature:"+weather.getMax());

System.out.println("\t\t Minimum Temperature:"+weather.getMin());

}

weather.shutdown();

}

catch (Exception e)

{

System.out.println("ERROR:"+e);

e.printStackTrace();

}

}

}

**Weather.idl**

module WeatherApp

{

interface Weather

{

void adjustTemp(in double r);

double getCurrent();

double getMax();

double getMin();

void call1(in double k);

oneway void shutdown();

};

};

**OUTPUT:**

E:\Corba>set PATH=%PATH%.; E:\Corba;

E:\Corba>set CLASSPATH=%CLASSPATH%.; E:\Corba;

E:\Corba>idlj Weather.idl

E:\Corba>idlj -fall Weather.idl

E:\Corba>javac \*.java WeatherApp/\*.java

Note: WeatherApp/WeatherPOA.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

E:\Corba>Start orbd -ORBInitialPort 1050

E:\Corba>Start java WeatherServer -ORBInitialPort 1050 -ORBInitialHostlocalhost

E:\Corba>Start java WeatherClient -ORBInitialPort 1050 -ORBInitialHostlocalhost

**WEATHER FORECASTING**

City: CHENNAI

Maximum Temperature: 28.5

Minimum Temperature: 18.5

City: MUMBAI

Maximum Temperature: 28.5

Minimum Temperature: 18.4

City: DELHI

Maximum Temperature: 29.9

Minimum Temperature: 18.4

City: KOLKATTA

Maximum Temperature: 37.2

Minimum Temperature: 18.4

**Result:**

Thus the program for weather forecast using CORBA is executed& the output is verified.