**Department of Software Engineering**

**Course Code: CS332**

**Class: BESE6AB**

**Lab 08: Java RMI**

**Date: October 30, 2018**

**Time: 14:00 – 17:00 pm**

**Instructor: Dr. Asad Waqar Malik**

**Name: Ahmad Amjad Mughal**

**Reg No: 121672**

**Class: BSCS6C**

**Introduction**

When two JVMs need to communicate, Java RMI is one option we have to make that happen. In this article, we’ll bootstrap a simple example showcasing Java RMI technology.

**Lab Solution**

**RMI\_Interface**

**import** java.rmi.Remote;

**import** java.rmi.RemoteException;

**public** **interface** RMI\_Interface **extends** Remote

{

**void** HotelClient(**int** type) **throws** RemoteException;

**void** HotelClientList(**int** type) **throws** RemoteException;

**void** HotelClientBook(String name, **int** type) **throws** RemoteException;

**void** HotelClientGuests(String name) **throws** RemoteException;

}

**RMI\_Client**

**import** java.rmi.registry.\*;

**import** java.util.\*;

**public** **class** RMI\_Client

{

**static** Scanner *scan* = **new** Scanner(System.***in***);

**public** RMI\_Client() {}

**public** **static** **void** main(String args[])

{

String name;

**int** type;

RMI\_Client client = **new** RMI\_Client();

type = client.RoomReservation();

System.***out***.println("Enter Your Name");

name = *scan*.next();

client.connectServer(name, type);

}

**private** **int** RoomReservation()

{

**int** type;

System.***out***.println("There are Total 5 Room Types are available at different prices");

System.***out***.println("We have 10 rooms of type 0 which are single rooms that costs 55 Euros a night");

System.***out***.println("We have 20 rooms of type 1 which are single rooms that costs 75 Euros a night");

System.***out***.println("We have 5 rooms of type 2 which are single rooms that costs 80 Euros a night");

System.***out***.println("We have 3 rooms of type 3 which are single rooms that costs 150 Euros a night");

System.***out***.println("We have 2 rooms of type 4 which are single rooms that costs 230 Euros a night");

System.***out***.println("Enter The Option");

type = *scan*.nextInt();

**return** type;

}

**private** **void** connectServer(String name, **int** type)

{

**try**

{

Registry register = LocateRegistry.*getRegistry*("127.0.0.1", 1099);

RMI\_Interface stub = (RMI\_Interface)register.lookup("RMI\_Interface");

System.***out***.println("----We are Checking----");

stub.HotelClient(type);

stub.HotelClientBook(name, type);

stub.HotelClientList(type);

stub.HotelClientGuests(name);

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

**RMI\_Server**

**import** java.rmi.registry.\*;

**import** java.rmi.RemoteException;

**import** java.rmi.server.UnicastRemoteObject;

**public** **class** RMI\_Server **extends** UnicastRemoteObject **implements** RMI\_Interface

{

**int** countType0 = 10, countType1 = 20, countType2 = 5, countType3 = 3, countType4 = 2;

**int** countGuests = 0;

String guests[] = **new** String[40];

**public** RMI\_Server() **throws** RemoteException

{

**super**();

}

**public** **void** HotelClient(**int** type) **throws** RemoteException{

**switch**(type)

{

**case** 0:

countType0--;

System.***out***.println(countType0 + " Rooms are available of type 0");

**break**;

**case** 1:

countType1--;

System.***out***.println(countType1 + " Rooms are avaliable of type 1");

**break**;

**case** 2:

countType2--;

System.***out***.println(countType2 + " Rooms are available of type 2");

**break**;

**case** 3:

countType3--;

System.***out***.println(countType3 + " Rooms are avaliable of type 3");

**break**;

**case** 4:

countType4--;

System.***out***.println(countType4 + " Rooms are available of type 4");

**break**;

**default**:

**break**;

}

}

**public** **void** HotelClientList(**int** type) **throws** RemoteException

{

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

System.***out***.println("Following Rooms are available for booking");

System.***out***.println(countType0 + " of Type 0 are available for 55 Euros per night ");

System.***out***.println(countType1 + " of Type 1 are available for 75 Euros per night ");

System.***out***.println(countType2 + " of Type 2 are available for 80 Euros per night ");

System.***out***.println(countType3 + " of Type 3 are available for 150 Euros per night ");

System.***out***.println(countType4 + " of Type 4 are available for 230 Euros per night ");

}

**public** **void** HotelClientBook(String name, **int** type) **throws** RemoteException

{

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

System.***out***.println(name + " is residing in room of type " + type);

}

**public** **void** HotelClientGuests(String name) **throws** RemoteException

{

System.***out***.println("Current Guests are");

guests[countGuests] = name;

**for** (**int** i = 0; i <= countGuests; i++)

System.***out***.println( guests[i]);

countGuests++;

}

**public** **static** **void** main(String args[])

{

**try**

{

Registry registry = LocateRegistry.*createRegistry*(1099);

registry.bind("RMI\_Interface", **new** RMI\_Server());

System.***out***.println("Alloting Rooms");

}

**catch**(Exception e)

{

System.***err***.println("This is Exception of type " + e.toString());

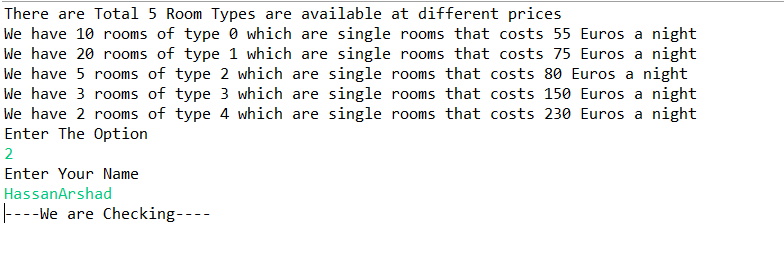
e.printStackTrace();

}

}

}

**Screenshot**

****

****