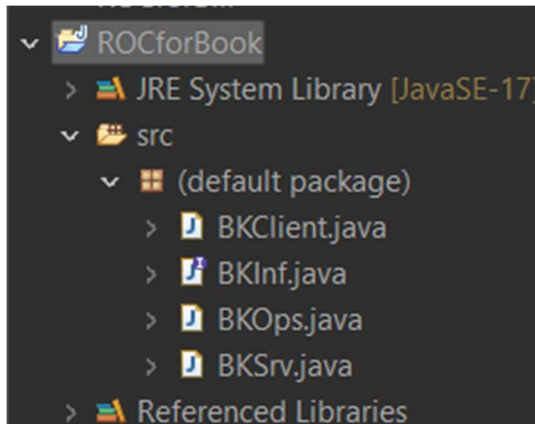


## Practical No. 04

### Remote Object Communication

**Q.1 Using MySQL create Library database. Create table Book (Book\_id, Book\_name, Book\_author) and retrieve the Book information from Library database using Remote Object Communication concept.**



**Code:**

#### **BKSrv.java**

```
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class BKSrv {
    public BKSrv() {
        super();
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            BKInf skeleton = new BKOps();
            LocateRegistry.createRegistry(1900);
            Naming.rebind("rmi://localhost:1900/ROCforBook", skeleton);
            System.out.println("Server Registered");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
}

```

### **BKClient.java**

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.rmi.Naming;

public class BKClient {

    public BKClient() {

        super();

    }

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        String sql = "", ch = "";

        try {

            BKInf stub = (BKInf) Naming.lookup("rmi://localhost:1900/ROCforBook");

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

            while(true) {

                System.out.println("Select an Option");

                System.out.println("1. Retrieve Book Info : ");

                System.out.println("2. Insert Book Info : ");

                System.out.println("3. Exit ");

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

                ch = br.readLine();

                if(ch.equals("1")) {

                    sql = "select * from libData";

                    sql = stub.getData(sql);

                }

            }

        }

    }

}

```



```

import java.rmi.server.UnicastRemoteObject;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

public class BKOps extends UnicastRemoteObject implements BKInf{

    private static final long serialVersionUID = 1L;

    Connection con;

    Statement stmt;

    ResultSet rs;

    ResultSetMetaData rsmd;

    String colStr, resultStr;

    public BKOps() throws RemoteException{

        super();

        con = null;

        stmt = null;

        rs = null;

        rsmd = null;

        colStr = "";

        resultStr = "";

    }

    public void setDBCon() {

        try {

            String URL = "jdbc:mysql://localhost:3306/librarysys";

            Class.forName("com.mysql.jdbc.Driver");

            con = DriverManager.getConnection(URL,"root","");

        }catch(Exception e) {

```

```

        e.printStackTrace();
    }
}

public String getData(String strQry) throws RemoteException {
    try {
        setDBCon();
        System.out.println("Server Registered");
        stmt = con.createStatement();
        rs = stmt.executeQuery(strQry);
        rsmd = rs.getMetaData();
        for (int i = 1; i <= rsmd.getColumnCount(); i++) {
            colStr = colStr + rsmd.getColumnName(i) + "\t";
        }
        while(rs.next()) {
            for (int i = 1; i <= rsmd.getColumnCount(); i++) {
                resultStr = resultStr + rs.getString(i) + "\t";
            }
            resultStr = resultStr + "\n";
        }
    }
    catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    }
    return colStr + "\n\n" + resultStr;
}

public String insertData(String strQry) throws RemoteException {
    try {

```

```
        setDBCon();

        System.out.println("Server Registered");

        stmt = con.createStatement();

        int recordInserted = stmt.executeUpdate(strQry);

        if(recordInserted != 0) {

            resultStr = "\nRecord inserted Successfully!";

        }else {

            resultStr = "\nRecord not inserted Successfully!";

        }

    }catch (Exception e) {

        // TODO: handle exception

        e.printStackTrace();

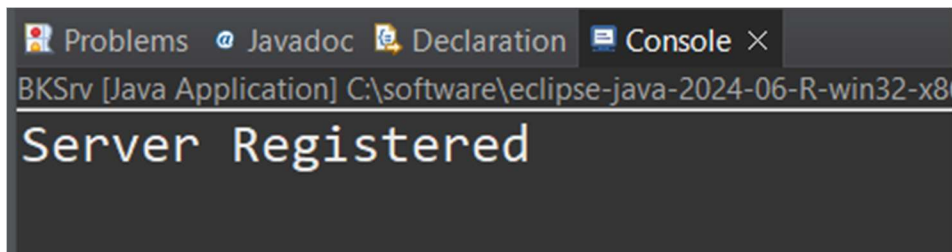
    }

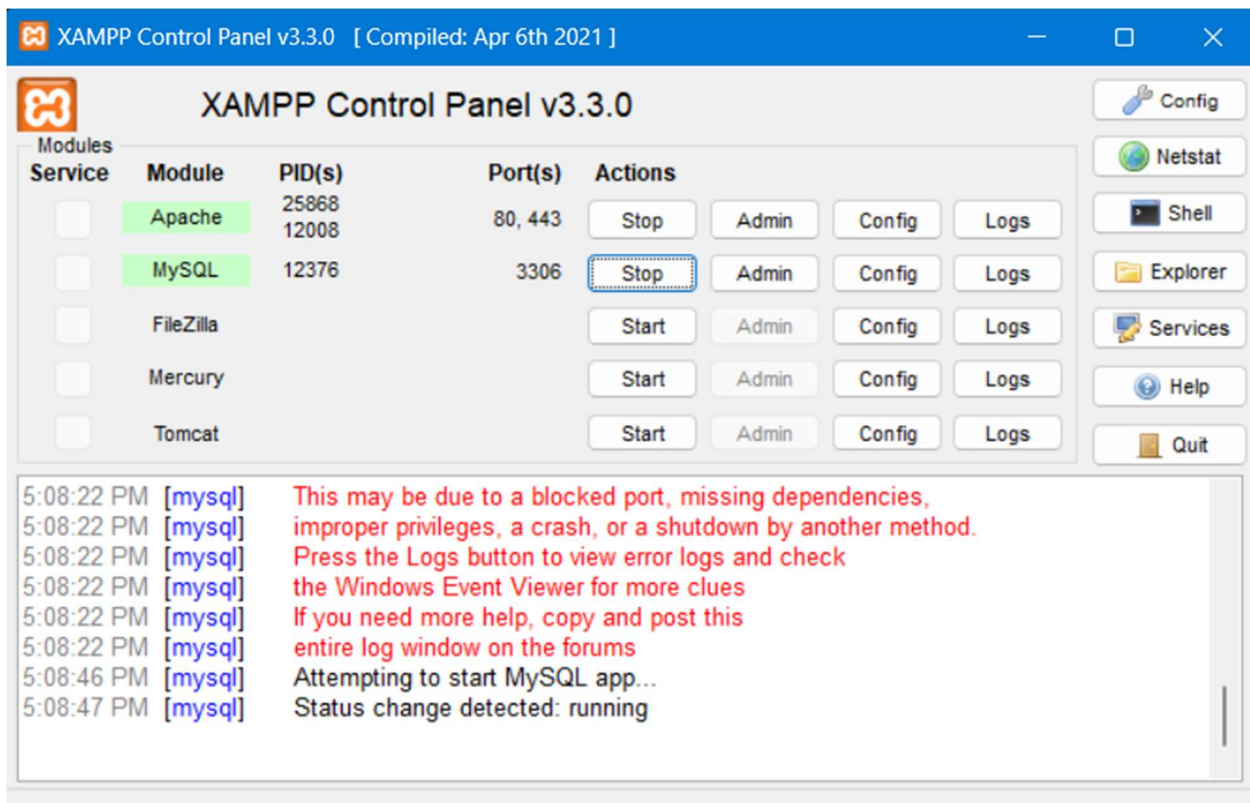
    return resultStr;

}

}
```

**Output:**





```

Select an Option
1. Retrieve Book Info :
2. Insert Book Info :
3. Exit
Enter Your Choice :
1
bookId  bookName          bookAuthor

1       The Great Gatsby    F. Scott Fitzgerald
2       To Kill a Mockingbird Harper Lee
3       1984               George Orwell

```

```
Select an Option
1. Retrieve Book Info :
2. Insert Book Info :
3. Exit
Enter Your Choice :
2

Record inserted Successfully!
```

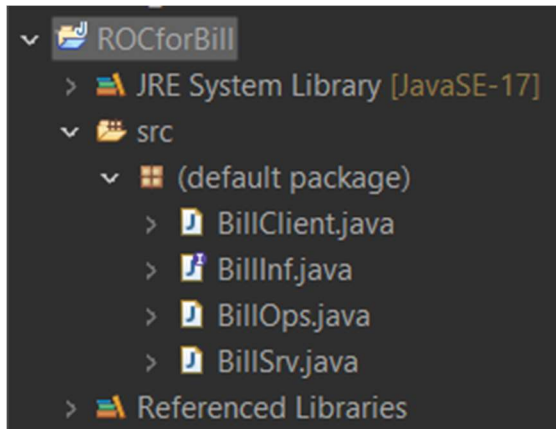
```
Select an Option
1. Retrieve Book Info :
2. Insert Book Info :
3. Exit
Enter Your Choice :
1
```

| bookId                        | bookName              | bookAuthor       | bookId | bookName            | bookAuthor |
|-------------------------------|-----------------------|------------------|--------|---------------------|------------|
| Record inserted Successfully! | 1                     | The Great Gatsby |        | F. Scott Fitzgerald |            |
| 2                             | To Kill a Mockingbird | Harper Lee       |        |                     |            |
| 3                             | 1984                  | George Orwell    |        |                     |            |
| 101                           | The Alamack N R       | Tim Drewes       |        |                     |            |

| bookId | bookName              | bookAuthor          |
|--------|-----------------------|---------------------|
| 1      | The Great Gatsby      | F. Scott Fitzgerald |
| 2      | To Kill a Mockingbird | Harper Lee          |
| 3      | 1984                  | George Orwell       |
| 101    | The Alamack N R       | Tim Drewes          |



**Q.2 Using MySQL create Electric\_Bill database. Create table Bill (consumer\_name, bill\_due\_date, bill\_amount) and retrieve the bill information from the Electric\_Bill database using Remote Object Communication concept.**



**Code:**

**BillSrv.java**

```
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class BillSrv {
    public BillSrv() {
        super();
    }

    public static void main(String[] args) {
        try {
            BillInf skeleton = new BillOps();
            LocateRegistry.createRegistry(1900);
            Naming.rebind("rmi://localhost:1900/ROCforBill", skeleton);
            System.out.println("Server Registered");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

### **BillClient.java**

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.rmi.Naming;

public class BillClient {

    public BillClient() {

        super();

    }

    public static void main(String[] args) {

        String sql = "", ch = "";

        try {

            BillInf stub = (BillInf) Naming.lookup("rmi://localhost:1900/ROCforBill");

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

            while(true) {

                System.out.println("Select an Option");

                System.out.println("1. Retrieve Bill Info : ");

                System.out.println("2. Insert Bill Info : ");

                System.out.println("3. Exit ");

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

                ch = br.readLine();

                if(ch.equals("1")) {

                    sql = "select * from billData";

                    sql = stub.getData(sql);

                }

                else if(ch.equals("2")) {

                    sql = "insert into billData(consumeNm, billDt, billAmt)
values ('Onkar', '10-11-2024', 500)";

                    sql = stub.insertData(sql);
```

```

        }
        else if(ch.equals("3")) {
            System.exit(0);
        }
        else {
            sql = "Please enter valid Input";
        }
        System.out.println(sql);
    }
}
}catch (Exception e) {
    e.printStackTrace();
}
}
}

```

#### **BillInf.java**

```

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface BillInf extends Remote{
    public String getData(String strQry) throws RemoteException;
    public String insertData(String strQry) throws RemoteException;
}

```

#### **BillOps.java**

```

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;

```

```

import java.sql.ResultSetMetaData;

import java.sql.Statement;

public class BillOps extends UnicastRemoteObject implements BillInf{

    private static final long serialVersionUID = 1L;

    Connection con;

    Statement stmt;

    ResultSet rs;

    ResultSetMetaData rsmd;

    String colStr, resultStr;

    public BillOps() throws RemoteException{

        super();

        con = null;

        stmt = null;

        rs = null;

        rsmd = null;

        colStr = "";

        resultStr = "";

    }

    public void setDBCon() {

        try {

            String URL = "jdbc:mysql://localhost:3306/billsys";

            Class.forName("com.mysql.jdbc.Driver");

            con = DriverManager.getConnection(URL,"root","");

        }catch(Exception e) {

            e.printStackTrace();

        }

    }

    public String getData(String strQry) throws RemoteException {

```

```

try {
    setDBCon();
    System.out.println("Server Registered");
    stmt = con.createStatement();
    rs = stmt.executeQuery(strQry);
    rsmd = rs.getMetaData();
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
        colStr = colStr + rsmd.getColumnName(i) + "\t";
    }
    while(rs.next()) {
        for (int i = 1; i <= rsmd.getColumnCount(); i++) {
            resultStr = resultStr + rs.getString(i) + "\t";
        }
        resultStr = resultStr + "\n";
    }
}
catch (Exception e) {
    e.printStackTrace();
}
return colStr + "\n\n" + resultStr;
}

public String insertData(String strQry) throws RemoteException {
    try {
        setDBCon();
        System.out.println("Server Registered");
        stmt = con.createStatement();
        int recordInserted = stmt.executeUpdate(strQry);
        if(recordInserted != 0) {

```

```

        resultStr = "\nRecord inserted Successfully!";
    }else {
        resultStr = "\nRecord not inserted Successfully!";
    }
}catch (Exception e) {
    e.printStackTrace();
}
return resultStr;
}
}

```

**Output:**

BillSrv [Java Application] C:\software\eclipse-java-2

**Server Registered**

Select an Option

1. Retrieve Bill Info :

2. Insert Bill Info :

3. Exit

Enter Your Choice :

1

| consumeNM | billDt | billAmt |
|-----------|--------|---------|
|-----------|--------|---------|

|       |            |     |
|-------|------------|-----|
| Onkar | 10-11-2024 | 500 |
|-------|------------|-----|

Select an Option

1. Retrieve Bill Info :

2. Insert Bill Info :

3. Exit

Enter Your Choice :

2

Record inserted Successfully!

Select an Option

1. Retrieve Bill Info :

2. Insert Bill Info :

3. Exit

Enter Your Choice :

1

| consumeNM | billDt | billAmt | consumeNM | billDt | billAmt |
|-----------|--------|---------|-----------|--------|---------|
|-----------|--------|---------|-----------|--------|---------|

Record inserted Successfully!Onkar 10-11-2024 500

|       |            |     |
|-------|------------|-----|
| Onkar | 10-11-2024 | 500 |
|-------|------------|-----|

| consumeNM | billDt     | billAmt |
|-----------|------------|---------|
| Onkar     | 10-11-2024 | 500     |
| Onkar     | 10-11-2024 | 500     |