Dt : 17/5/2022

Ex_Program : DBCon17.java

```java
package test;

import java.util.*;

import javax.sql.rowset.*;

public class DBCon17 {

    public static void main(String[] args) {

    try {

        Scanner s = new Scanner(System.in);

        Class.forName("oracle.jdbc.driver.OracleDriver");

        RowSetFactory rsf = RowSetProvider.newFactory();

            //Implemented Object of RowSetFactory

        System.out.println("===Choice====");

        System.out.println("1.JdbcRowSet\n2.CachedRRowSet");

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

        int choice = s.nextInt();

        switch(choice)

        {

        case 1:

            JdbcRowSet jrs = rsf.createJdbcRowSet();

            jrs.setUrl("jdbc:oracle:thin:@localhost:1521:xe");

            jrs.setUsername("system");

            jrs.setPassword("manager");

            jrs.setCommand("select * from Product45");
```

```java
        jrs.execute();

        while(jrs.next()) {

                System.out.println(jrs.getString(1)+"\t"+

            jrs.getString(2)+"\t"+jrs.getFloat(3)+"\t"+

                        jrs.getInt(4));

        }//end of loop

        break;

case 2:

        CachedRowSet crs = rsf.createCachedRowSet();

        crs.setUrl("jdbc:oracle:thin:@localhost:1521:xe");

        crs.setUsername("system");

        crs.setPassword("manager");

        crs.setCommand("select * from Bank45");

        crs.execute();

        while(crs.next()) {

                System.out.println(crs.getLong(1)+"\t"+

            crs.getString(2)+"\t"+crs.getFloat(3)+"\t"+

                        crs.getString(4));

        }//end of loop

        break;

default:

        System.out.println("Invalid Choice...");

}//end of switch;

s.close();
```

```
        }catch(Exception e) {e.printStackTrace();}

        }

}

o/p:

===Choice====

1.JdbcRowSet

2.CachedRRowSet

Enter the Choice:

2
```

6123456   Raj   9000.0 savings

313131   Ram   3500.0 Savings

56565   Alex   234.0   savings

456   Alex   234.0   savings

==

Summary of Objects generated from CoreJava:

1.User defined Class Object

2.String Objects

3.WrapperClass Objects

4.Array objects

5.Collection<E> objects

6.Map<K,V> objects

7.Enum<E> object

Summary of Objects generated from JDBC:

1.Connection Object

2.Statement Object

3.PreparedStatement Object

4.CallableStatement Object

5.NonScrollable ResultSet Object

6.Scrollable ResultSet Object

7.RowSet Object

  (i)JdbcRowSet Object

  (ii)CachedRowSet Object

    =>WebRowSet Object

      (i)FiltertedRowSet Object

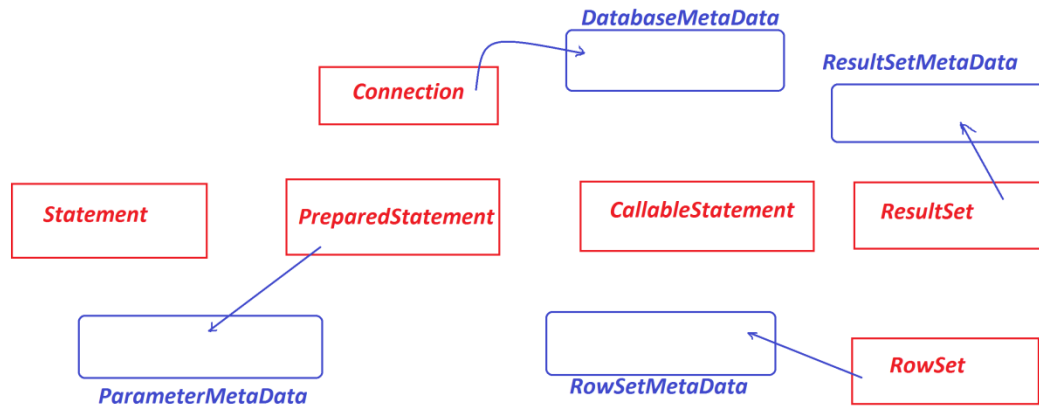   *(ii)JoinRowSet Object*

8.Metadata Objects

  (i)DatabaseMetaData object

  (ii)ParameterMetaData Object

  (iii)ResultSetMetaData Object

  (iv)RowSetMetaData Object

Diagram:

==

define Metadata in JDBC?

=>Metadata means the data holding information about another data

is known as Metadata.

=>According to Object Oriented proggramming,Metadata means one object

holding information about another object.

=>The following are the metadata components from JDBC:

1.DatabaseMetaData

2.ParameterMetaData

3.ResultSetMetaData

4.RowSetMetaData

1.DatabaseMetaData:

=>DatabaseMetaData is an interface from java.sql package and

which is used to hold information about Connection Object.

syntax:

DatabaseMetaData dmd = con.getMetaData();

2.ParameterMetaData:

=>ParameterMetaData is an interface from java.sql package and

which is used to hold information about PreparedStatement object.

syntax:

ParameterMetaData pmd = ps.getParameterMetaData();

3.ResultSetMetaData :

=>ResultSetMetaData is an interface from java.sql package and

which is used to hold the information about ResultSet Object.

syntax:

ResultSetMetaData rsmd = rs.getMetaData();

4.RowSetMetaData:

=>RowSetMetaData is also an interface from java.sql package and

which is used to hold information about RowSet Object.

syntax:

RowSetMetaData rsmd = (RowSetMetaData)rs.getMetaData();

Ex:

package test;

import java.sql.;*

```java
import javax.sql.rowset.*;

import javax.sql.*;

public class DBCon18 {

    public static void main(String[] args) {

    try {

        Class.forName("oracle.jdbc.driver.OracleDriver");

            Connection con = DriverManager.getConnection

                    ("jdbc:oracle:thin:@localhost:1521:XE","system","manager");

            DatabaseMetaData dmd = con.getMetaData();

            System.out.println("DriverVersion:"+dmd.getDriverMajorVersion());

            PreparedStatement ps = con.prepareStatement

                    ("select * from Book45 where bcode=?");

            ps.setString(1,"");

            ParameterMetaData pmd = ps.getParameterMetaData();

            System.out.println("Count:"+pmd.getParameterCount());

            ResultSet rs = ps.executeQuery();

            ResultSetMetaData rsmd = rs.getMetaData();

            System.out.println("Column count:"+rsmd.getColumnCount());


            JdbcRowSet jrs = RowSetProvider.newFactory().createJdbcRowSet();

            jrs.setUrl("jdbc:oracle:thin:@localhost:1521:XE");

            jrs.setUsername("system");

            jrs.setPassword("manager");

            jrs.setCommand("select * from Product45");
```

```
        jrs.execute();

        RowSetMetaData rsmd2 = (RowSetMetaData)jrs.getMetaData();

        System.out.println(rsmd2.getColumnCount());

        con.close();

        jrs.close();

    }catch(Exception e) {System.out.print(e.getMessage());}

    }

}
```

==

faq:

define Stream?

  =>The Continuous flow of data is known as Stream.

Types of Streams:

   =>Streams in Java are categorized into two types:

      1.Byte Stream(Binary Stream)

      2.Character Stream

1.Byte Stream(Binary Stream):

   =>The continuous flow of 8-bit data is known as Byte Stream or Binary

Stream.

   =>Byte Stream supports all multi-media data files like Text,Audio,Video,

Image and Animation.

2.Character Stream:

 =>The Continuous flow of 16-bit data is known as Character Stream.

 =>Character Stream is preferable for Text data and not preferable for

Audio,Video,Image and Animation files.

 ---

faq:

wt is the diff b/w

 (i)input stream

 (ii)output stream

(i)input stream:

 =>The stream into JavaProgram is known as input stream.

   Ex:

    JavaProgram reading stream from file

(ii)output stream:

 =>The stream outof JavaProgram is known as output stream.

   Ex:

    JavaProgram writing stream onto file

===

Note:

 =>we use the following SQL-types to store streams:

   1.BLOB

   2.CLOB

1.BLOB:

  =>BLOB stands for 'Binary Large OBjects' and which is used to store binary

Stream data.(Multi-Media data)

2.CLOB:

  =>CLOB stands for 'Character Large OBjects' and which is used to store

Character Stream data.(Text data)

 ===
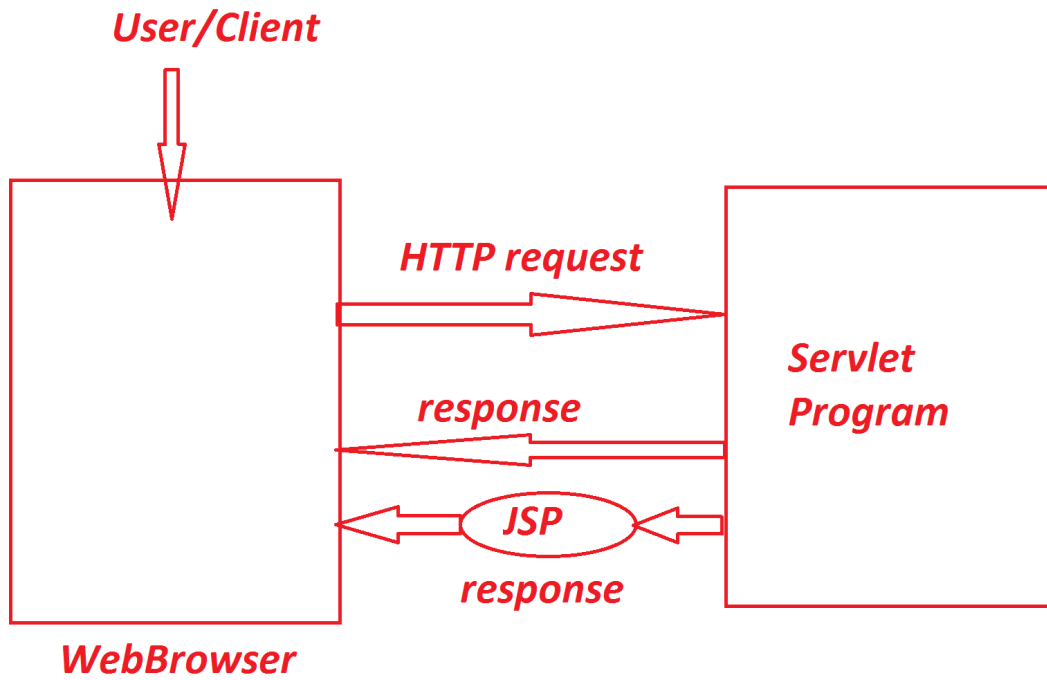
*imp

Servlet Programming:(Unit-2)

define Servlet Program?

  =>The PlatForm independent JavaProgram which is executed in

Server environment is known as Servlet Program or Server Program.

  =>Servlet Program will accept HTTP request from user through

WebBrowser and provides response.

Diagram:

User/Client

HTTP request

Servlet Program

response

JSP

response

WebBrowser

===