**package** com.dxc.training.dbcon;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** DBConnection {

**public** DBConnection() {

// **TODO** Auto-generated constructor stub

}

**public** **static** Connection getConnection() {

**try** {

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

}

**catch**(ClassNotFoundException e) {

e.printStackTrace();

}

Connection connection = **null**;

**try** {

connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/dxc","root","root");

}**catch**(SQLException e) {

e.printStackTrace();

}

**return** connection;

}

}

**package** com.dxc.training;

/\*\*

\* **@author** akumar2078

\*

\*/

**public** **class** users {

**private** String username;

**private** String password;

**public** users() {

// **TODO** Auto-generated constructor stub

}

**public** users(String username, String password) {

**super**();

**this**.username = username;

**this**.password = password;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

@Override

**public** **int** hashCode() {

**final** **int** prime = 31;

**int** result = 1;

result = prime \* result + ((password == **null**) ? 0 : password.hashCode());

result = prime \* result + ((username == **null**) ? 0 : username.hashCode());

**return** result;

}

@Override

**public** **boolean** equals(Object obj) {

**if** (**this** == obj)

**return** **true**;

**if** (obj == **null**)

**return** **false**;

**if** (getClass() != obj.getClass())

**return** **false**;

users other = (users) obj;

**if** (password == **null**) {

**if** (other.password != **null**)

**return** **false**;

} **else** **if** (!password.equals(other.password))

**return** **false**;

**if** (username == **null**) {

**if** (other.username != **null**)

**return** **false**;

} **else** **if** (!username.equals(other.username))

**return** **false**;

**return** **true**;

}

@Override

**public** String toString() {

**return** "users [username=" + username + ", password=" + password + "]";

}

}

**package** com.dxc.trainig.dao;

**import** java.sql.SQLException;

**import** com.dxc.training.users;

**public** **interface** usersDAO {

**public** **boolean** validateUsers(String username,String password) **throws** SQLException;

}

**package** com.dxc.trainig.dao;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** com.dxc.training.users;

**import** com.dxc.training.dbcon.DBConnection;

**public** **class** usersDAOImpl **implements** usersDAO {

Connection connection = DBConnection.*getConnection*();

**public** **boolean** status = **false**;

**public** usersDAOImpl() {

// **TODO** Auto-generated constructor stub

}

@Override

**public** **boolean** validateUsers(String username, String password) **throws** SQLException {

Statement st = connection.createStatement();

String queryCheck = "SELECT \* from users WHERE Username = Abhijit AND Passowrd = Kumar";

ResultSet rs = st.executeQuery(queryCheck);

**if** (rs.next()) {

status = **true**;

}

**return** status;

}

}

**package** client;

**import** java.sql.SQLException;

**import** java.util.Scanner;

**import** com.dxc.trainig.dao.trainingDAOImpl;

**import** com.dxc.trainig.dao.usersDAOImpl;

**import** com.dxc.training.users;

**public** **class** userapp {

**public** userapp() {

usersDAOImpl user = **new** usersDAOImpl();

}

**public** **void** launch() **throws** SQLException {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Please enter username :");

String uname = scanner.next();

System.***out***.println("Please enter password :");

String pass = scanner.next();

usersDAOImpl user = **new** usersDAOImpl();

**if** (user.validateUsers(uname, pass)) {

**int** choice;

choice = scanner.nextInt();

**switch**(choice) {

**case** 1:

trainingDAOImpl data = **new** trainingDAOImpl();

data.getAllData();

**break**;

**case** 2:

data.getonebyone();

**case** 3:

}

} **else** {

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

}

}

}

**package** client;

**import** java.sql.SQLException;

**public** **class** Main {

**public** Main() {

// **TODO** Auto-generated constructor stub

}

**public** **static** **void** main(String[] args) **throws** SQLException {

userapp app = **new** userapp();

app.launch();

}

}

**package** com.dxc.training;

**public** **class** training {

**private** **int** Sap\_Id;

**private** String Employee\_name;

**private** String Stream;

**private** **int** Percentage;

**public** training() {

// **TODO** Auto-generated constructor stub

}

**public** training(**int** sap\_Id, String employee\_name, String stream, **int** percentage) {

**super**();

Sap\_Id = sap\_Id;

Employee\_name = employee\_name;

Stream = stream;

Percentage = percentage;

}

**public** **int** getSap\_Id() {

**return** Sap\_Id;

}

**public** **void** setSap\_Id(**int** sap\_Id) {

Sap\_Id = sap\_Id;

}

**public** String getEmployee\_name() {

**return** Employee\_name;

}

**public** **void** setEmployee\_name(String employee\_name) {

Employee\_name = employee\_name;

}

**public** String getStream() {

**return** Stream;

}

**public** **void** setStream(String stream) {

Stream = stream;

}

**public** **int** getPercentage() {

**return** Percentage;

}

**public** **void** setPercentage(**int** percentage) {

Percentage = percentage;

}

@Override

**public** **int** hashCode() {

**final** **int** prime = 31;

**int** result = 1;

result = prime \* result + ((Employee\_name == **null**) ? 0 : Employee\_name.hashCode());

result = prime \* result + Percentage;

result = prime \* result + Sap\_Id;

result = prime \* result + ((Stream == **null**) ? 0 : Stream.hashCode());

**return** result;

}

@Override

**public** **boolean** equals(Object obj) {

**if** (**this** == obj)

**return** **true**;

**if** (obj == **null**)

**return** **false**;

**if** (getClass() != obj.getClass())

**return** **false**;

training other = (training) obj;

**if** (Employee\_name == **null**) {

**if** (other.Employee\_name != **null**)

**return** **false**;

} **else** **if** (!Employee\_name.equals(other.Employee\_name))

**return** **false**;

**if** (Percentage != other.Percentage)

**return** **false**;

**if** (Sap\_Id != other.Sap\_Id)

**return** **false**;

**if** (Stream == **null**) {

**if** (other.Stream != **null**)

**return** **false**;

} **else** **if** (!Stream.equals(other.Stream))

**return** **false**;

**return** **true**;

}

@Override

**public** String toString() {

**return** "training [Sap\_Id=" + Sap\_Id + ", Employee\_name=" + Employee\_name + ", Stream=" + Stream

+ ", Percentage=" + Percentage + "]";

}

}

package com.dxc.trainig.dao;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.List;

import com.dxc.training.training;

import com.dxc.training.dbcon.DBConnection;

public class trainingDAOImpl implements trainingDAO {

Connection connection = DBConnection.getConnection();

private static final String FETCH\_TABLE\_ALL= "select \* from training";

private static final String TYPE\_SCROLL\_INSENSITIVE;

private static final String CONCUR\_UPDATABLE;

public trainingDAOImpl() {

// TODO Auto-generated constructor stub

}

@Override

public List<training> getAllData() throws SQLException {

Statement stat= connection.createStatement();

ResultSet res = stat.executeQuery(FETCH\_TABLE\_ALL);

return null;

}

public void getonebyone() {

try {

Statement stat =

connection.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

ResultSet res = stat.executeQuery(FETCH\_TABLE\_ALL);

ResultSetMetaData rsmd = res.getMetaData();

while(res.next()) {

for (int i = 1; i <= rsmd.getColumnCount(); i++) {

System.out.print(res.getString(i) + " ");

}

System.out.println("\nEnter the percentage you want to update:-");

percentage= scanner.nextInt();

res.updateInt(3, percentage);

res.updateRow();

}

}