package day10;

import java.sql.\*;

import java.util.Scanner;

public class task1 {

static Connection connect()

{

try {

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

Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3309/login","root","dbms");

if(con!=null)

{

return con;

}

}catch(Exception e) {

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

}

return null;

}

static void CreateTable(){

Connection con= *connect*();

String query="create table signup(id int,username varchar(30),password varchar(30))";

try {

PreparedStatement ps = con.prepareStatement(query);

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

ps.executeUpdate();

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

con.close();

ps.close();

}catch(SQLException e) {

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

}

}

static void InsertTable() {

int id;

String username,password;

Connection con = *connect*();

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

System.***out***.println("enter ID :");

id = s.nextInt();

System.***out***.println("enter Name :");

username = s.next();

System.***out***.println("enter Password :");

password = s.next();

String query = "insert into signup (id,username,password) values(?,?,?)";

try{

PreparedStatement ps = con.prepareStatement(query);

ps.setInt(1,id);

ps.setString(2, username);

ps.setString(3, password);

ps.executeUpdate();

con.close();

ps.close();}

catch(SQLException e)

{

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

}

}

static void DeleteTable() {

int id;

Connection con = *connect*();

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

System.***out***.println("enter ID :");

id = s.nextInt();

String query = "delete from signup where id = ?";

try{

PreparedStatement ps = con.prepareStatement(query);

ps.setInt(1, id);

ps.executeUpdate();

con.close();

ps.close();}

catch(SQLException e)

{

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

}

}

static void UpdateTable() {

int id;

String username,password;

Connection con = *connect*();

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

System.***out***.println("enter ID :");

id = s.nextInt();

System.***out***.println("enter Name :");

username = s.next();

System.***out***.println("enter Password :");

password = s.next();

String query = "update signup set username = ?,password =? where id=id";

try{

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, username);

ps.setString(2, password);

ps.setInt(3, id);

ps.executeUpdate();

con.close();

ps.close();}

catch(SQLException e)

{

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

}

}

static void View() {

Connection con = *connect*();

String query="select \* from signup" ;

try {

PreparedStatement ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery();

while(rs.next())

{

System.***out***.println("Id:"+rs.getInt(1)+"Name:"+rs.getString(2)+"Password:"+rs.getString(3));

}

con.close();

ps.close();

}

catch(SQLException e) {

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

}

}

public static void main(String args[])

{

int n;

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

do {

System.***out***.println("1.Create Tables\n2.Insert\n3.Modify\n4.Display\n5.Delete\n6.Exit\n");

System.***out***.println("Enter your option:\n");

n = s.nextInt();

switch(n)

{

case 1:

*CreateTable*();

break;

case 2:

*InsertTable*();

break;

case 3:

//ModifyTable();

break;

case 4:

*View*();

break;

case 5:

*DeleteTable*();

break;

case 6:

System.*exit*(0);

break;

default:

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

}

}while(n!=6);

}

}

package day10;

import java.sql.\*;

import java.util.Scanner;

public class task2 {

static Connection connect() {

try {

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

Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3309/userlogin", "root", "dbms");

return con;

} catch (Exception e) {

System.***out***.println("Connection error: " + e);

}

return null;

}

static void CreateTable() {

Connection con = *connect*();

if (con == null) return;

String query = "CREATE TABLE users(id INT, username VARCHAR(30), password VARCHAR(30))";

try (PreparedStatement ps = con.prepareStatement(query)) {

ps.executeUpdate();

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

} catch (SQLException e) {

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

}

}

static boolean authenticateUser(Connection con, String username, String password) {

String sql = "SELECT \* FROM users WHERE username = ? AND password = ?";

try {

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, username);

pstmt.setString(2, password);

ResultSet rs = pstmt.executeQuery();

return rs.next();

} catch (SQLException e) {

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

}

return false;

}

static boolean updatePassword(Connection con, String username, String newPassword) {

String sql = "UPDATE users SET password = ? WHERE username = ?";

try {

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, newPassword);

pstmt.setString(2, username);

return pstmt.executeUpdate() > 0;

} catch (SQLException e) {

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

}

return false;

}

public static void main(String[] args) {

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

*CreateTable*();

System.***out***.print("Enter username: ");

String username = s.next();

System.***out***.print("Enter password: ");

String password = s.next();

Connection con = *connect*();

if (con == null) {

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

return;

}

if (*authenticateUser*(con, username, password)) {

System.***out***.println("Login successful! Welcome " + username);

} else {

System.***out***.println("Incorrect password. Please reset your password.");

System.***out***.print("Enter new password: ");

String newPassword = s.next();

System.***out***.print("Confirm new password: ");

String confirmPassword = s.next();

if (newPassword.equals(confirmPassword)) {

if (*updatePassword*(con, username, newPassword)) {

System.***out***.println("Password reset successfully. Please log in again.");

System.***out***.print("Enter username: ");

username = s.next();

System.***out***.print("Enter password: ");

password = s.next();

if (*authenticateUser*(con, username, password)) {

System.***out***.println("Login successful! Welcome " + username);

} else {

System.***out***.println("Login failed. Exiting...");

}

} else {

System.***out***.println("User not found. Exiting...");

}

} else {

System.***out***.println("Passwords do not match. Exiting...");

}

}

try {

con.close();

}

catch (SQLException e) {

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

}

s.close();

}

}

Enter username: hari

Enter password: 123456

Incorrect password. Please reset your password.

Enter new password: 123

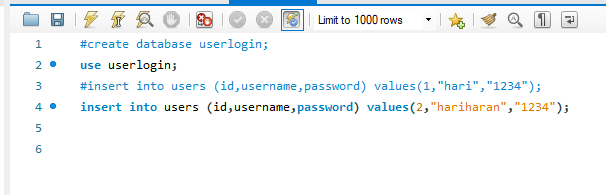
Confirm new password: 123

Password reset successfully. Please log in again.

Enter username: hari

Enter password: 123

Login successful! Welcome hari



The default method and abstracted method cannot be executed since the interface cannot be instatiated . only the static method runs because a static method in interface can be called by using the interface name itself .

package day10;

interface Second{

abstract void aclass(); // this is abstract class

default void fun() { // this is a default method

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

}

static void fun1() {

System.***out***.println("static method of the error");

}

}

class Summa implements Second{

public void aclass()

{

System.***out***.println("over rided abstract method of inteface");

}

}

public class task1 {

public static void main()

{

Summa s = new Summa();

s.aclass();

s.fun();

}

}

In order to access the default abstract method we will have to create a new class that implements the interface and then use its object for accessing ….

***ANONYMOUS INNER CLASS :***  
to over come the above problem for creating the new class we create the anonymous inner class .

**FUNCTIONAL INTERFACE :**

An interface which has only one abstract method is called functional interface ….

The F.I we do not worry about existing of default or static method ..

Legal ::

Interface interface\_name {

Public void fun(); }

ilegal ::

Interface interface\_name{

Public void fun();

Public void fun1();

}

Interface A

{ Default() 🡪 ilegal

Static() }

The functional interface always denoted as -- @ --annotation at functional interface .

example @FuntionalInteface

package lamda;

interface Second1{

abstract void aclass(); // this is abstract class

default void fun() { // this is a default method

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

}

}

public class task1 {

public static void main()

{

Second1 s = new Second1() {

public void aclass() {

System.***out***.println("abstract class... from anonymous inner class...");

}

};

s.aclass();

s.fun();

}

}

**LAMBDA :**

package lamda;

interface Second1{

abstract void aclass(); // this is abstract class

default void fun() { // this is a default method

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

}

}

public class task1 {

public static void main()

{

Second1 s = ()->System.***out***.println("abstract class");

}

}