**1.\*\*\*\*\*\*\*\*\*\*\*\*\*Invoice Details:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DBConnection {

public static Connection getConnection() throws SQLException, InstantiationException, IllegalAccessException, ClassNotFoundException{

//fill your code

ResourceBundle rb = ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

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

Connection connection = DriverManager.getConnection(url,username,password);

return connection;

}

}

---------------------------------

public class InvalidInvoiceNumberException extends Exception {

InvalidInvoiceNumberException(String s){

super(s);

}

}

-------------------------------------------

public class Invoice {

Integer id;

String customerName;

String invoiceNumber;

Double amount;

Double totalAmount;

public Invoice(){}

public Invoice(Integer id, String customerName, String invoiceNumber,

Double amount,Double totalAmount){

this.id = id;

this.customerName = customerName;

this.invoiceNumber = invoiceNumber;

this.amount = amount;

this.totalAmount = totalAmount ;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getCustomerName() {

return customerName;

}

public void setCustomerName(String customerName) {

this.customerName = customerName;

}

public String getInvoiceNumber() {

return invoiceNumber;

}

public void setInvoiceNumber(String invoiceNumber) {

this.invoiceNumber = invoiceNumber;

}

public Double getAmount() {

return amount;

}

public void setAmount(Double amount) {

this.amount = amount;

}

public Double getTotalAmount() {

return totalAmount;

}

public void setTotalAmount(Double totalAmount) {

this.totalAmount = totalAmount;

}

}

------------------------------------------------

import java.util.List;

import java.io.\*;

import java.sql.SQLException;

public class Main {

public static void main(String agrs[]) throws IOException,InvalidInvoiceNumberException, InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

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

Integer invoiceId,vat,cst,travel;

SalesInvoiceBO salesInvoiceBO = new SalesInvoiceBO();

ServiceInvoiceBO serviceInvoiceBO = new ServiceInvoiceBO();

System.out.println("1).Sales Tax");

System.out.println("2).Service Tax");

Integer ch = Integer.parseInt(br.readLine());

if(ch==1){

List<SalesInvoice> salesList = salesInvoiceBO.listSalesInvoice();

System.out.format("%s %8s %8s %8s\n","Id","VatTax","CstTax","Invoice Number");

for(SalesInvoice salesListIns:salesList){

System.out.format("%s %7s %7s %7s\n",salesListIns.getId(),salesListIns.getVatTaxPercentage(),salesListIns.getCstTaxPercentage(),salesListIns.getInvoiceNumber());

}

System.out.println("Enter the invoice number");

String salesInvoiceNumber = br.readLine();

try{

if(!salesInvoiceNumber.startsWith("INV")){

throw new InvalidInvoiceNumberException("Invalid invoice number");

}

else{

System.out.println("Enter the vat tax");

vat = Integer.parseInt(br.readLine());

System.out.println("Enter the cst tax");

cst = Integer.parseInt(br.readLine());

//fill the code

SalesInvoice salesinvoice = salesInvoiceBO.getSalesInvoice(salesInvoiceNumber);

salesinvoice.setCstTaxPercentage(cst);

salesinvoice.setVatTaxPercentage(vat);

salesInvoiceBO.updateTaxPercentage(salesinvoice);

System.out.println("TotalAmount "+salesinvoice.computeTax());

}}catch(InvalidInvoiceNumberException e){

//System.out.println(e.getMessage());

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

}

}

else{

List<ServiceInvoice> serviceList = serviceInvoiceBO.listServiceInvoice();

System.out.format("%s %7s %7s\n","Id","TravelTax","Invoice Number");

for(ServiceInvoice serviceListIns:serviceList){

System.out.format("%s %7s %7s\n",serviceListIns.getId(),serviceListIns.getTravelTaxPercentage(),serviceListIns.getInvoiceNumber());

}

System.out.println("Enter the invoice number");

String serviceInvoiceNumber = br.readLine();

try{

if(!serviceInvoiceNumber.startsWith("INV")){

throw new InvalidInvoiceNumberException("Invalid invoice number");

}

else{

System.out.println("Enter the travel tax");

travel = Integer.parseInt(br.readLine());

//fill the code

ServiceInvoice serviceinvoice = null;

try {

serviceinvoice = serviceInvoiceBO.getServiceInvoice(serviceInvoiceNumber);

} catch (InstantiationException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IllegalAccessException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

serviceinvoice.setTravelTaxPercentage(travel);

serviceInvoiceBO.updateTaxPercentage(serviceinvoice);

System.out.println("TotalAmount "+serviceinvoice.computeTax());

} }catch(InvalidInvoiceNumberException e){

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

//System.out.println(e.getMessage());

}

}

}

}

----------------------------------------------

public class SalesInvoice extends Invoice{

Integer vatTaxPercentage;

Integer cstTaxPercentage;

public SalesInvoice(){

super();

}

public SalesInvoice(Integer id, String customerName, String invoiceNumber,

Double amount,Double balanceAmount,Integer vatTaxPercentage,Integer cstTaxPercentage) {

super(id,customerName,invoiceNumber,amount,balanceAmount);

this.vatTaxPercentage = vatTaxPercentage;

this.cstTaxPercentage = cstTaxPercentage;

}

public Integer getVatTaxPercentage() {

return vatTaxPercentage;

}

public void setVatTaxPercentage(Integer vatTaxPercentage) {

this.vatTaxPercentage = vatTaxPercentage;

}

public Integer getCstTaxPercentage() {

return cstTaxPercentage;

}

public void setCstTaxPercentage(Integer cstTaxPercentage) {

this.cstTaxPercentage = cstTaxPercentage;

}

public Double computeTax(){

//fill your code

double finalamount = this.amount;

finalamount = finalamount\*(1+(cstTaxPercentage+vatTaxPercentage)\*0.01);

return finalamount;

}

}

--------------------------------------------------

import java.sql.SQLException;

import java.util.List;

public class SalesInvoiceBO {

SalesInvoiceDAO salesInvoiceDAO = new SalesInvoiceDAO();

public void updateTaxPercentage(SalesInvoice salesInvoice) throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

//fill your code

salesInvoiceDAO.updateTaxPercentage(salesInvoice);

}

public List<SalesInvoice> listSalesInvoice() throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

return salesInvoiceDAO.listSalesInvoiceDetails();

}

public SalesInvoice getSalesInvoice(String SalesInvoiceNumber) throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

return salesInvoiceDAO.getSalesInvoice(SalesInvoiceNumber);

}

}

----------------------------------------------

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class SalesInvoiceDAO {

public List<SalesInvoice> listSalesInvoiceDetails() throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException {

List<SalesInvoice> salesinvoices = new ArrayList<SalesInvoice>();

Connection connection = DBConnection.getConnection();

Statement stmt=connection.createStatement();

ResultSet rs = stmt.executeQuery("select \*from sales\_invoice,invoice where sales\_invoice.id = invoice.id");

while(rs.next()){

salesinvoices.add(new SalesInvoice(rs.getInt("id"), rs.getString("customer\_name"), rs.getString("invoice\_number"), rs.getDouble("amount"),rs.getDouble("total\_amount"),rs.getInt("vat\_Tax\_Percentage"),rs.getInt("cst\_Tax\_Percentage")));

}

return salesinvoices;

//fill your code

}

public SalesInvoice getSalesInvoice(String salesInvoiceNumber) throws SQLException, InstantiationException, IllegalAccessException, ClassNotFoundException{

Connection connection = DBConnection.getConnection();

Statement stmt=connection.createStatement();

ResultSet rs = stmt.executeQuery("select \*from sales\_invoice,invoice where invoice.invoice\_number = '"+salesInvoiceNumber+

"' and sales\_invoice.id = invoice.id");

rs.next();

return (new SalesInvoice(rs.getInt("id"), rs.getString("customer\_name"), rs.getString("invoice\_number"), rs.getDouble("amount"),rs.getDouble("total\_amount"),rs.getInt("vat\_Tax\_Percentage"),rs.getInt("cst\_Tax\_Percentage")));

//fill your code

}

public void updateTaxPercentage(SalesInvoice salesInvoice) throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

//fill your code

Connection connection = DBConnection.getConnection();

Statement stmt=connection.createStatement();

stmt.execute("update sales\_invoice set vat\_Tax\_Percentage="+salesInvoice.getVatTaxPercentage()+",cst\_Tax\_Percentage="+ salesInvoice.getCstTaxPercentage()+

" where id="+salesInvoice.getId());

}

}

--------------------------------------------------

public class ServiceInvoice extends Invoice {

Integer travelTaxPercentage;

ServiceInvoice(){

super();

}

ServiceInvoice(Integer id, String customerName, String invoiceNumber,

Double amount,Double balanceAmount,Integer travelTaxPercentage){

super(id,customerName,invoiceNumber,amount,balanceAmount);

this.travelTaxPercentage = travelTaxPercentage;

}

public Integer getTravelTaxPercentage() {

return travelTaxPercentage;

}

public void setTravelTaxPercentage(Integer travelTaxPercentage) {

this.travelTaxPercentage = travelTaxPercentage;

}

public Double computeTax(){

//fill your code

double finalamount = this.amount;

finalamount = finalamount\*(1+(travelTaxPercentage\*0.01));

return finalamount;

}

}

--------------------------------------------

import java.sql.SQLException;

import java.util.List;

public class ServiceInvoiceBO {

ServiceInvoiceDAO serviceInvoiceDAO = new ServiceInvoiceDAO();

public void updateTaxPercentage(ServiceInvoice serviceInvoice) throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

//fill your code

serviceInvoiceDAO.updateTaxPercentage(serviceInvoice);

}

public List<ServiceInvoice> listServiceInvoice() throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

//fill your code

return serviceInvoiceDAO.listServiceInvoiceDetails();

}

public ServiceInvoice getServiceInvoice(String serviceInvoiceNumber) throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

return serviceInvoiceDAO.getServiceInvoice(serviceInvoiceNumber);

//fill your code

}

}

-------------------------------------------

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class ServiceInvoiceDAO {

public List<ServiceInvoice> listServiceInvoiceDetails() throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException {

List<ServiceInvoice> serviceInvoices = new ArrayList<ServiceInvoice>();

Connection connection = DBConnection.getConnection();

Statement stmt=connection.createStatement();

ResultSet rs = stmt.executeQuery("select \*from service\_invoice,invoice where service\_invoice.id = invoice.id");

while(rs.next()){

serviceInvoices.add(new ServiceInvoice(rs.getInt("id"), rs.getString("customer\_name"), rs.getString("invoice\_number"), rs.getDouble("amount"),rs.getDouble("total\_amount"),rs.getInt("travel\_Tax\_Percentage")));

}

return serviceInvoices;

//fill your code

}

public ServiceInvoice getServiceInvoice(String serviceInvoiceNumber) throws InstantiationException, IllegalAccessException, ClassNotFoundException, SQLException{

Connection connection = DBConnection.getConnection();

Statement stmt=connection.createStatement();

ResultSet rs = stmt.executeQuery("select \*from service\_invoice,invoice where invoice.invoice\_number = '"+serviceInvoiceNumber+

"' and service\_invoice.id = invoice.id");

rs.next();

return (new ServiceInvoice(rs.getInt("id"), rs.getString("customer\_name"), rs.getString("invoice\_number"), rs.getDouble("amount"),rs.getDouble("total\_amount"),rs.getInt("travel\_Tax\_Percentage")));

//fill your code

}

public void updateTaxPercentage(ServiceInvoice serviceInvoice) throws SQLException, InstantiationException, IllegalAccessException, ClassNotFoundException{

//fill your code

Connection connection = DBConnection.getConnection();

Statement stmt=connection.createStatement();

stmt.execute("update service\_invoice set travel\_Tax\_Percentage="+serviceInvoice.getTravelTaxPercentage()+" where id="+serviceInvoice.getId());

}

}

**2.\*\*\*\*\*\*\*\*\*\*\*\*ShipmentDetails\*\*\*\*\*\*\*\*\*\*:**

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import static org.hamcrest.collection.IsIterableContainingInAnyOrder.containsInAnyOrder;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class ShipmentJunit {

ShipmentBO shipmentBO;

SimpleDateFormat sdf;

List<Shipment> shipmentlist = new ArrayList<Shipment>();

@Before

public void shipmentObjectCreation()

{

//fill code here.

shipmentBO = new ShipmentBO();

sdf = new SimpleDateFormat("dd/MM/yyyy");

}

@Test

public void testUpdateArrivalDate() throws Exception

{

//fill code here.

shipmentBO.updateArrivalDate(45,"12/10/2017", ShipmentDAO.shipmentList);

List<Shipment> shipmentlist = ShipmentDAO.shipmentList;

for(int i=0;i<shipmentlist.size();i++){

if(shipmentlist.get(i).getId()==45){

assertEquals(0,(shipmentlist.get(i).getArrivalDate()).compareTo(sdf.parse("12/10/2017")));

}

}

}

@Test

public void testFilterByArrivalDate() throws Exception

{

//fill code here.

/\* List<String> expnames = null;

expnames.add("Laptop");

expnames.add("Television");

expnames.add("Washing Machine");

expnames.add("Air Conditioner");

expnames.add("Air Cooler");

expnames.add("Personal Computer");

expnames.add("Samsung Mobile");

expnames.add("Home Theater");\*/

List<String> names = shipmentBO.filterByArrivalDate(ShipmentDAO.shipmentList, sdf.parse("11/11/2015"));

assertThat(names, containsInAnyOrder("Laptop","Television","Washing Machine","Air Conditioner","Air Cooler","Personal Computer",

"Samsung Mobile","Home Theater"));

}

}

**3.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*InvoicePayment\*\*\*\*\*\*\*\*:**

import java.util.regex.Pattern;

public class CreditCardPayment extends Payment// fill the code

{

private String cardNumber;

private String cvv;

private String cardName;

public CreditCardPayment() {

}

public CreditCardPayment(String name, Double amount, String cardNumber, String cvv, String cardName) {

super(name, amount);

this.cardNumber = cardNumber;

this.cvv = cvv;

this.cardName = cardName;

}

public String getCardNumber() {

return cardNumber;

}

public void setCardNumber(String cardNumber) {

this.cardNumber = cardNumber;

}

public String getCvv() {

return cvv;

}

public void setCvv(String cvv) {

this.cvv = cvv;

}

public String getCardName() {

return cardName;

}

public void setCardName(String cardName) {

this.cardName = cardName;

}

@Override

public Double calculateTotalAmount() throws InvalidPaymentException{

if(!Pattern.matches("^[0-9]{16}$",cardNumber)||!Pattern.matches("^[0-9]{3}$",cvv))

{

throw new InvalidPaymentException("Invalid Card Details");

}

else

{

Double d=super.amount+(0.1\*super.amount);

return d;

}

}

}

-----------------------------------

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DbConnection {

public static Connection getConnection() throws ClassNotFoundException, SQLException {

ResourceBundle rb= ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

//fill the code

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

Connection con=DriverManager.getConnection(

url,username,password);

return con;

}

}

--------------------------------

public class InvalidPaymentException extends Exception {

// fill the code

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public InvalidPaymentException(String message)

{

super(message);

}

}

------------------------------

public class Invoice {

private Integer id;

private String customerName;

private Integer paymentAttempts;

private Double totalAmount;

private Double balance;

private String status;

public Invoice() {

}

public Invoice(Integer id, String customerName, Integer paymentAttempts,Double totalAmount, Double balance, String status) {

super();

this.id = id;

this.customerName = customerName;

this.paymentAttempts = paymentAttempts;

this.totalAmount = totalAmount;

this.balance = balance;

this.status = status;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getCustomerName() {

return customerName;

}

public void setCustomerName(String customerName) {

this.customerName = customerName;

}

public Integer getPaymentAttempts() {

return paymentAttempts;

}

public void setPaymentAttempts(Integer paymentAttempts) {

this.paymentAttempts = paymentAttempts;

}

public Double getTotalAmount() {

return totalAmount;

}

public void setTotalAmount(Double totalAmount) {

this.totalAmount = totalAmount;

}

public Double getBalance() {

return balance;

}

public void setBalance(Double balance) {

this.balance = balance;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

@Override

public String toString() {

// TODO Auto-generated method stub

return String.format("%-5s %-15s %-20s %-15s %-10s %s\n",id, customerName, paymentAttempts, totalAmount, balance, status);

}

}

------------------------------

import java.sql.SQLException;

import java.util.List;

public class InvoiceBO {

InvoiceDAO invoiceDAO=new InvoiceDAO();

public List<Invoice> getAllInvoice() throws ClassNotFoundException, SQLException {

//fill the code

return invoiceDAO.getAllInvoices();

}

public void updateInvoiceDetails(int id,Double amount) throws ClassNotFoundException, SQLException {

//fill the code

invoiceDAO.updateInvoiceDetails(id, amount);

}

}

------------------------

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class InvoiceDAO {

public List<Invoice> getAllInvoices() throws ClassNotFoundException, SQLException{

// fill the code

Connection con=DbConnection.getConnection();

List<Invoice> listOfInvoices = new ArrayList<>();

try {

Statement stmt=con.createStatement();

ResultSet resultSet=stmt.executeQuery("select \* from invoice");

while (resultSet.next()) {

listOfInvoices.add(new Invoice(resultSet.getInt("id"),resultSet.getString("customer\_name"),resultSet.getInt("payment\_attempts"),resultSet.getDouble("total\_amount"),resultSet.getDouble("balance"),resultSet.getString("status")));

}

}catch(Exception e){

con.close();

System.out.println(e);

}

return listOfInvoices;

}

public void updateInvoiceDetails(Integer invoiceId, Double amount) throws ClassNotFoundException, SQLException{

// fill the code

Connection con=DbConnection.getConnection();

Connection con1=DbConnection.getConnection();

int count=0;

try {

Statement stmt=con.createStatement();

ResultSet resultSet=stmt.executeQuery("select \* from invoice where id="+invoiceId);

while (resultSet.next()) {

count=resultSet.getInt("payment\_attempts");

}

count=count+1;

Statement stmt1=con1.createStatement();

stmt1.executeUpdate("update invoice set payment\_attempts="+count+",balance="+amount+" where id="+invoiceId);

}catch(Exception e){

con.close();

System.out.println(e);

}

}

}

------------------------

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.sql.SQLException;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.List;

public class Main {

public static void main(String[] args) throws IOException, ClassNotFoundException, SQLException, ParseException, InvalidPaymentException {

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

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

System.out.println("Invoice Details:");

System.out.format("%-5s %-15s %-20s %-15s %-10s %s\n","ID","Customer Name","Payment Attempts","Total Amount","Balance","Status");

try {

//fill the code

InvoiceBO invoiceBO=new InvoiceBO();

List<Invoice> listOfInvoices=invoiceBO.getAllInvoice();

for(Invoice invoice:listOfInvoices)

{

System.out.format("%-5s %-15s %-20s %-15s %-10s %s\n",invoice.getId(),invoice.getCustomerName(),invoice.getPaymentAttempts(),invoice.getTotalAmount(),invoice.getBalance(),invoice.getStatus());

}

System.out.println("Enter the invoice id to pay :");

Integer id = Integer.parseInt(buff.readLine());

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

String name = buff.readLine();

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

Double amount = Double.parseDouble(buff.readLine());

System.out.println("Enter the card number :");

String cardNo = buff.readLine();

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

String cvv = (buff.readLine());

System.out.println("Enter the card name :");

String cardName = buff.readLine();

CreditCardPayment creditCardPayment=new CreditCardPayment(name, amount, cardNo, cvv, cardName);

Double totalAmount=creditCardPayment.calculateTotalAmount();

// fill the code

Double balanceAmount=0.0;

System.out.printf("Total Amount to be paid is %.2f\n",totalAmount);

//fill the code

for(Invoice invoice:listOfInvoices)

{

if(invoice.getId()==id)

{

balanceAmount=invoice.getBalance()-amount;

}

}

invoiceBO.updateInvoiceDetails(id, balanceAmount);

System.out.printf("The balance amount is %.2f",balanceAmount);

}

catch(InvalidPaymentException ipe)

{

System.out.println(ipe);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

---------------------------

public abstract class Payment {

protected String accountHolderName;

protected Double amount;

public Payment() {

}

public Payment(String accountHolderName, Double amount) {

super();

this.accountHolderName = accountHolderName;

this.amount = amount;

}

public String getAccountHolderName() {

return accountHolderName;

}

public void setAccountHolderName(String accountHolderName) {

this.accountHolderName = accountHolderName;

}

public Double getAmount() {

return amount;

}

public void setAmount(Double amount) {

this.amount = amount;

}

// fill the code

public abstract Double calculateTotalAmount() throws InvalidPaymentException;

}

**4.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SkillWar\*\*\*\*\*\*\*\*\*\*\***

**Contact.java**

package jdbc.cc.section1;

public class Contact {

private Integer id;

private String orgName;

private String street;

private String city;

private String state;

Contact(String orgName, String street, String city, String state) {

this.orgName = orgName;

this.street = street;

this.city = city;

this.state = state;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getOrgName() {

return orgName;

}

public void setOrgName(String orgName) {

this.orgName = orgName;

}

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

}

**DBConnection.java**

package jdbc.cc.section1;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DBConnection {

public static Connection getConnection()

throws SQLException, InstantiationException, IllegalAccessException, ClassNotFoundException {

ResourceBundle rb = ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

Connection connect = null;

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

connect = DriverManager.getConnection(url, username, password);

return connect;

}

}

**Main.java**

package jdbc.cc.section1;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.\*;

public class Main {

public static void main(String ags[]) throws Exception {

DateFormat format = new SimpleDateFormat("yyyy-MM-dd");

UserDAO userDao = new UserDAO();

List<User> userList = userDao.getAllUserSortByLastLogin();

System.out.println("List of all user details based on last login time");

System.out.format("%-15s %-15s %-15s %-25s %-15s %-15s\n", "Name", "LastLogin Time", "Organization", "Street",

"City", "State");

for (User user : userList) {

System.out.format("%-15s %-15s %-15s %-25s %-15s %-15s\n", user.getName(),

format.format(user.getLastLoginTime()), user.getContact().getOrgName(),

user.getContact().getStreet(), user.getContact().getCity(), user.getContact().getState());

}

}

}

**User.java**

package jdbc.cc.section1;

import java.sql.Date;

public class User {

private Integer id;

private String name;

private Date lastLoginTime;

private Contact contact;

User(String name, Date lastLoginTime, Contact contact) {

this.name = name;

this.lastLoginTime = lastLoginTime;

this.contact = contact;

}

public Contact getContact() {

return contact;

}

public void setContact(Contact contact) {

this.contact = contact;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Date getLastLoginTime() {

return lastLoginTime;

}

public void setLastLoginTime(Date lastLoginTime) {

this.lastLoginTime = lastLoginTime;

}

}

**UserDAO.java**

package jdbc.cc.section1;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import java.sql.Date;

public class UserDAO {

public List<User> getAllUserSortByLastLogin()

throws ClassNotFoundException, SQLException, InstantiationException, IllegalAccessException {

List<User> userList = new ArrayList<User>();

Connection connection = DBConnection.getConnection();

// execute query;

Statement statement = connection.createStatement();

ResultSet userRs = statement.executeQuery("SELECT \* FROM user\_details");

while (userRs.next()) {

String userName = userRs.getString("username");

Date date = userRs.getDate("last\_login\_time");

int contactId = userRs.getInt("contact\_id");

String orgName = null;

String street = null;

String city = null;

String state = null;

Statement statement1 = connection.createStatement();

ResultSet contactRs = statement1.executeQuery("SELECT \* FROM contact WHERE id=" + contactId);

if (contactRs.next()) {

orgName = contactRs.getString("org\_name");

street = contactRs.getString("street");

city = contactRs.getString("city");

state = contactRs.getString("state");

}

Contact contact = new Contact(orgName, street, city, state);

User user = new User(userName, date, contact);

userList.add(user);

}

Collections.sort(userList, new Comparator<User>() {

@Override

public int compare(User o1, User o2) {

return o1.getLastLoginTime().compareTo(o2.getLastLoginTime());

}

});

return userList;

}

}

=====================================================================================