**정보시스템이행**

**문제해결시나리오**

**평가자료**

**- 은행 데이터베이스 -**

수업명 :java,python 기반의 AI 활용 응용

소프트웨어 개발자 양성과정

제출일 : 2020. 12. 16

작성자 : 유승화

□프로젝트 개요

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **비고** |
| **제목** | Sql을 활용한 은행 프로그램 |  |
| **프로젝트**  **작성기간** | ~2020.12.16 |  |
| **조원 구성** | 유승화 |  |
| **프로젝트 내용** | 데이터베이스를 이용하여 model1형태의 은행 프로그램 제작 |  |
| **주요기능** | 은행 추가/삭제 |  |
| 계좌 추가/삭제 |  |
| 계좌 정보 수정 |  |
| **학습내용** | 데이터베이스(오라클) 사용법 |  |
| Try/catch |  |
| 데이터베이스 설계 |  |
| 패키지 |  |

□최종 보고서

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **비고** |
| **제목** | Sql을 활용한 은행 프로그램 |  |
| **프로젝트 내용** | 데이터베이스를 이용하여 model1형태의 은행 프로그램 제작 |  |
| **완료내용** | 은행 추가/삭제 | 프로젝트  개요와 동일 |
| 계좌 추가/삭제 | 프로젝트  개요와 동일 |
| 계좌 정보 수정 | 프로젝트  개요와 동일 |
| **프로젝트 활동 후기** | | |
| **좋았던 점** | 데이터베이스의 기본적인 사용법을 알 수 있었다 |  |
| **아쉬웠던 점 및**  **어려웠던 점** | 서로 종속 관계에 있는 자료의 구조를 수정하는 게 헷갈렸다 |  |
| 평소 하는 방식과 다르게 데이터 입출력. 데이터를 담을 객체 등을 나누어 하다 보니 생소했다 |  |
| 여러 개의 은행을 구현하기 어려웠다 |  |
| **학습 효과** | 유지보수가 용이하도록 model1형태로 프로그램을 개발하는 법을 학습 |  |

□프로젝트 세부 사항 분석 및 구조 설계

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **주요 기능/구문** | **내용** | **입출력** |
| **1** | Try/catch | exception오류를 제어한다 |  |
| **2** | 패키지 | 기능에 따라 구분하여 저장한다 |  |
| **3** | Singleton(instance) | 데이터를 관리할 때 객체를 하나만 생성하여 관리한다 |  |
| **4** |  |  |  |

□작성 과정, 실행결과

|  |  |  |
| --- | --- | --- |
| **구현 내용** | **결과** | **비고** |
| 계정 추가, 제거, 로그인, 입금, 출금, 조회 | 완료 |  |
| 여러 개의 은행 구현 | 미완 |  |

□실행결과-1

텍스트이(가) 표시된 사진

자동 생성된 설명

계정 추가

텍스트이(가) 표시된 사진

자동 생성된 설명

로그인

텍스트이(가) 표시된 사진

자동 생성된 설명

입금

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

출금

텍스트이(가) 표시된 사진

자동 생성된 설명

잔액조회

텍스트이(가) 표시된 사진

자동 생성된 설명

종료화면

텍스트이(가) 표시된 사진

자동 생성된 설명

데이터가 저장된 모습

□소스코드-1

DAO

계좌 DAO-------------------------------------------------------------------------------

package dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import dto.AccountDTO;

import util.DBcon;

public class AccountDAO {

public int insertAcc(AccountDTO dto) {

int returnValue=0;

DBcon.getInstance();

String sql="insert into account values(%d, '%s', '%s', %d)";

sql=String.format(sql, dto.getAccID(),dto.getName(),dto.getPw(),dto.getBalance());

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

System.out.println("계정 추가됨");

DBcon.dbClose();

return returnValue;

}

public int updateName(Long accID,String newName) {

int returnValue=0;

DBcon.getInstance();

String sql="update account set name='"+newName+"' where accID='"+accID+"'";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int updatePW(Long accID,String newPW) {

int returnValue=0;

DBcon.getInstance();

String sql="update account set pw='"+newPW+"' where accID='"+accID+"'";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int updateBal(Long accID,int newBal) {

int returnValue=0;

DBcon.getInstance();

String sql="update account set balance='"+newBal+"' where accID='"+accID+"'";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int deleteAcc(Long accID) {

int returnValue=0;

DBcon.getInstance();

String sql="delete from account where accID='"+accID+"'";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int resetAccTable() {

int returnValue=0;

DBcon.getInstance();

String sql="drop table account";

DBcon.statementUpdate(sql);

sql="create table account(accID number(10), name varchar2(50), pw varchar2(50), balance number(10))";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public ArrayList<AccountDTO> select(){

ArrayList<AccountDTO> arr=new ArrayList<AccountDTO>();

ResultSet rs;

DBcon.getInstance();

String sql="select \* from account";

rs=DBcon.statementQuery(sql);

try {

while (rs.next()) {

arr.add(new AccountDTO(

rs.getLong(1),

rs.getString(2),

rs.getString(3),

rs.getInt(4)

));

}

} catch (SQLException e) {

e.printStackTrace();

}

DBcon.dbClose();

return arr;

}

}

은행 DAO------------------------------------------------------------

package dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import dto.BankDTO;

import util.DBcon;

public class BankDAO {

public int insertBank(BankDTO dto) {

int returnValue=0;

DBcon.getInstance();

String sql="insert into bank values(%d,'%s',%d)";

sql=String.format(sql, dto.getBankID(),dto.getBankName(),dto.getAccID());

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int updateBname(String bankName,String rename) {

int returnValue=0;

DBcon.getInstance();

String sql="update bank set name'"+rename+"' where name='"+bankName+"'";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int deleteBank(String bankName) {

int returnValue=0;

DBcon.getInstance();

String sql="delete from bank where name='"+bankName+"'";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public int resetBankTable() {

int returnValue=0;

DBcon.getInstance();

String sql="drop table bank;";

DBcon.statementUpdate(sql);

sql="create table bank("

+ "bankID number(10),"

+ "bankName varchar2(50),"

+ "accID number(10))";

DBcon.statementUpdate(sql);

sql="commit";

DBcon.statementUpdate(sql);

DBcon.dbClose();

return returnValue;

}

public ArrayList<BankDTO> select(){

ArrayList<BankDTO> arr=new ArrayList<BankDTO>();

ResultSet rs;

DBcon.getInstance();

String sql="select \* from bank";

rs=DBcon.statementQuery(sql);

try {

while (rs.next()) {

arr.add(new BankDTO(

rs.getLong(1),

rs.getString(2),

rs.getLong(3)

));

}

} catch (SQLException e) {

e.printStackTrace();

}

DBcon.dbClose();

return arr;

}

}

은행 메뉴----------------------------------------------------------------------------

package dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import dto.AccountDTO;

import util.DBcon;

import util.DBinput;

public class BankMenu {

private long tempID=1;

String loginName;

long loginID;

int loginBal;

boolean isLogin=false;

boolean isExit=false;

AccountDAO acc=new AccountDAO();

public void login() {

System.out.print("이름 :");

String inputName=DBinput.inputString();

System.out.print("비밀번호 :");

String inputPW=DBinput.inputString();

ResultSet rs;

DBcon.getInstance();

String sql="select \* from account where name='"+inputName+"' and pw='"+inputPW+"'";

rs=DBcon.statementQuery(sql);

try {

if (rs.next()) {

loginName=rs.getString(2);

loginID=rs.getLong(1);

loginBal=rs.getInt(4);

System.out.println(loginName+" 로그인");

isLogin=true;

}else {

System.out.println("잘못된 이름/비밀번호");

}

} catch (SQLException e) {

e.printStackTrace();

}

DBcon.dbClose();

}

public void addAcc() {

System.out.print("사용할 이름 :");

String inputName=DBinput.inputString();

System.out.print("사용할 비밀번호 :");

String inputPW=DBinput.inputString();

acc.insertAcc(new AccountDTO(tempID,inputName,inputPW,0));

tempID++;

}

public void addBal() {

System.out.print("입금할 금액 입력 :");

int inputInt=DBinput.inputInt();

acc.updateBal(loginID, loginBal+inputInt);

loginBal+=inputInt;

System.out.println(inputInt+" 입금");

}

public void delBal() {

System.out.print("출금할 금액 입력 :");

int inputInt=DBinput.inputInt();

if(loginBal-inputInt<0) {

System.out.println("잔액이 부족합니다");

}else {

acc.updateBal(loginID, loginBal-inputInt);

loginBal-=inputInt;

System.out.println(inputInt+" 출금");

}

}

public void displayBal() {

System.out.println(loginName+"의 현재 잔액 : "+loginBal);

}

public void logout() {

isLogin=false;

loginBal=0;

loginID=0;

loginName=null;

}

public void startMenu() {

System.out.printf("로그인:1, 회원가입:2, 종료:3%n입력 : ");

int loginInput = DBinput.inputInt();

switch (loginInput) {

case 1:

login();

break;

case 2:

addAcc();

break;

case 3:

isExit=true;

return;

default:

System.out.println("잘못된 입력");

break;

}

}

public void mainMenu() {

System.out.println("1.입금 2.출금 3.조회 4.로그아웃");

System.out.print("원하는 메뉴 입력 : ");

int menuInput = DBinput.inputInt();

switch (menuInput) {

case 1:

System.out.println("입금");

addBal();

break;

case 2:

System.out.println("출금");

delBal();

break;

case 3:

displayBal();

break;

case 4:

logout();

return;

default:

System.out.println("잘못된 입력");

break;

}

}

public void init() {

acc.resetAccTable();

}

public void run() {

init();

while(isExit==false) {

while(isLogin==false && isExit==false) {

startMenu();

}

while (isLogin==true) {

mainMenu();

}

}

}

}

DTO

계좌 DTO---------------------------------------------------------------------------

package dto;

public class AccountDTO {

private long accID;

private String name;

private String pw;

private int balance=0;

public AccountDTO() {}

public AccountDTO(long accID, String name, String pw, int balance) {

super();

this.accID = accID;

this.name = name;

this.pw = pw;

if(balance<0)balance=0;

this.balance = balance;

}

public long getAccID() {

return accID;

}

public void setAccID(long accID) {

this.accID = accID;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPw() {

return pw;

}

public void setPw(String pw) {

this.pw = pw;

}

public int getBalance() {

return balance;

}

public void setBalance(int balance) {

if(balance<0)balance=0;

this.balance = balance;

}

@Override

public String toString() {

return "AccountDTO [accID=" + accID + ", name=" + name + ", pw=" + pw + ", balance=" + balance + "]";

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + (int) (accID ^ (accID >>> 32));

return result;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

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

return false;

AccountDTO other = (AccountDTO) obj;

if (accID != other.accID)

return false;

return true;

}

}

은행 DTO------------------------------------------------------------------------

package dto;

public class BankDTO {

private long bankID;

private String bankName;

private long accID;

public BankDTO() {}

public BankDTO(long bankID, String bankName, long accID) {

super();

this.bankID = bankID;

this.bankName = bankName;

this.accID = accID;

}

public long getBankID() {

return bankID;

}

public void setBankID(long bankID) {

this.bankID = bankID;

}

public String getBankName() {

return bankName;

}

public void setBankName(String bankName) {

this.bankName = bankName;

}

public long getAccID() {

return accID;

}

public void setAccID(long accID) {

this.accID = accID;

}

@Override

public String toString() {

return "BankDTO [bankID=" + bankID + ", bankName=" + bankName + ", accID=" + accID + "]";

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + (int) (accID ^ (accID >>> 32));

result = prime \* result + (int) (bankID ^ (bankID >>> 32));

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

BankDTO other = (BankDTO) obj;

if (accID != other.accID)

return false;

if (bankID != other.bankID)

return false;

if (bankName == null) {

if (other.bankName != null)

return false;

} else if (!bankName.equals(other.bankName))

return false;

return true;

}

}

UTIL

DB 연결 클래스-------------------------------------------------------

package util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class DBcon {

private DBcon() {}

private static Connection dbCon = null;

private static Statement st = null;

private static ResultSet rs = null;

public static Connection getInstance() {

if(dbCon == null) {

try {

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

String url ="jdbc:oracle:thin:@localhost:1521:xe", id = "human", pw = "human";

dbCon = DriverManager.getConnection(url, id, pw);

//System.out.println("-- connected --");

}catch (ClassNotFoundException e) {

e.printStackTrace();

}catch (SQLException e) {

e.printStackTrace();

}

}

return dbCon;

}

public static int statementUpdate(String sql) {

int resultValue = 0;

try {

if(st == null) {

st = dbCon.createStatement();

}

resultValue = st.executeUpdate(sql);

//System.out.println("-- 업데이트 --");

} catch (SQLException e) {

e.printStackTrace();

}

return resultValue;

}

public static ResultSet statementQuery(String sql) {

try {

if(st == null) {

st = dbCon.createStatement();

}

rs = st.executeQuery(sql);

} catch (SQLException e) {

e.printStackTrace();

}

return rs;

}

public static void dbClose() {

try {

if( rs != null ) rs.close();

if( st != null ) st.close();

if(dbCon != null) dbCon.close();

} catch (Exception e) {

e.printStackTrace();

} finally {

rs = null;

st = null;

dbCon = null;

}

}

}

사용자 입력 클래스------------------------------------------------------------

package util;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

public class DBinput {

static Scanner sc = new Scanner(System.in);

public static int inputInt() {

int num=0;

try {

num = Integer.parseInt(sc.nextLine());

} catch (NumberFormatException e) {

e.printStackTrace();

}finally {

}

return num;

}

public static Double inputDouble() {

Double num=null;

try {

num = Double.parseDouble(sc.nextLine());

} catch (NumberFormatException e) {

e.printStackTrace();

}

return num;

}

public static String inputString() {

String str=null;

str = sc.nextLine();

return str;

}

public static Date inputDate() {

Date d=null;

System.out.println("날짜 입력(yyyy-mm-dd hh24:mi:ss : ");

SimpleDateFormat transFormat=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String input = sc.nextLine();

try {

d=transFormat.parse(input);

} catch (ParseException e) {

e.printStackTrace();

}

return d;

}

public static String dateToString(Date d) {

DateFormat df = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

return df.format(d);

}

}

EX

실행파일--------------------------------------------------------------

package ex;

import dao.BankMenu;

public class Bank1216 {

public static void main(String[] args) {

BankMenu bank1=new BankMenu();

bank1.run();

System.out.println("종료");

}

}