**분석용 데이터구축**

**사례연구**

**평가자료**

**-은행프로그램제출-**

수업명 :java,python 기반의 AI 활용 응용

소프트웨어 개발자 양성과정

제출일 : 2020. 11. 26

작성자 : 유승화

□프로젝트 개요

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **비고** |
| **제목** | 은행관리프로그램 |  |
| **프로젝트**  **작성기간** | ~2020.11.26 |  |
| **조원 구성** | 유승화 |  |
| **프로젝트 내용** | 은행들과 은행계좌를 관리하는 프로그램을  Collection 이용해 제작 |  |
| **주요기능** | 여러 개의 은행을 관리할 수 있다 |  |
| 여러 사람의 계좌를 관리할 수 있다 |  |
| ArrayList를 이용하여 데이터를 저장하기 편리하게 하였다 |  |
| **학습내용** | 은행과 계좌 클래스 제작 |  |
| ArrayList로 Collection에 대한 내용 학습 |  |
| 메소드를 만들어 코드를 알아보기 쉽게 작성 |  |

□최종 보고서

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **비고** |
| **제목** | 은행 관리 프로그램 |  |
| **프로젝트 내용** | 은행들과 은행계좌를 관리하는 프로그램을  Collection 이용해 제작 |  |
| **완료내용** | 여러 개의 은행을 관리할 수 있다 | 프로젝트  개요와 동일 |
| 여러 사람의 계좌를 관리할 수 있다 | 프로젝트  개요와 동일 |
| ArrayList를 이용하여 데이터를 저장하기 편리하게 하였다 | 프로젝트  개요와 동일 |
| **프로젝트 활동 후기** | | |
| **좋았던 점** | 클래스와 메소드를 이용해 예전처럼 if문과 반복문이 남발되지 않고 정리되어 좋았다 |  |
| **아쉬웠던 점 및**  **어려웠던 점** | 입력 시 예외 처리 부분을 완전히 구현하지 못해 아쉽다 |  |
| 사소한 부분을 잘못 처리했을 때 발생하는 오류를 찾기 힘들었다 |  |
| 더 간소화해서 만들 수도 있을 것 같은데 아직 어려웠다 |  |
| **학습 효과** | Collection을 이용하여 데이터를 쉽게 더하고 삭제할 수 있었다 |  |

□프로젝트 세부 사항 분석 및 구조 설계

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **주요 기능/구문** | **내용** | **입출력** |
| **1** | Class | BankCol  BankAccountCol  BankManageCol |  |
| **2** | ArrayList | account  banks |  |
| **3** | 메소드 | Login()  mainMenu() 등 |  |
| **4** | For,while | 반복문 |  |

□작성 과정, 실행결과

|  |  |  |
| --- | --- | --- |
| **구현 내용** | **결과** | **비고** |
| ArrayList 이용하여 이름,비밀번호,잔액 관리 | 완료 |  |
| ArrayList 이용하여 은행이름, 회원수 관리 | 완료 |  |

□실행결과-1

텍스트이(가) 표시된 사진

자동 생성된 설명

은행 추가하기

텍스트이(가) 표시된 사진

자동 생성된 설명

은행에 로그인하여 계좌 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

계좌 생성하여 로그인

텍스트이(가) 표시된 사진

자동 생성된 설명

은행 업무

텍스트이(가) 표시된 사진

자동 생성된 설명

관리자 메뉴1

텍스트이(가) 표시된 사진

자동 생성된 설명

관리자 메뉴2

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램 종료

□소스코드-1

1.계좌 관리 클래스------------------------------------------------

package bank;

public class BankAccountCol {

private String name="";

private String pw="";

private int id=0;

private int balance=0;

public BankAccountCol() {}

public BankAccountCol(String name, String pw, int id,int balance) {

super();//부모 클래스

this.name = name;

this.pw = pw;

this.id = id;

this.balance = balance;

if(balance<0)balance=0;

}

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 getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getBalance() {

return balance;

}

public void setBalance(int balance) {

this.balance = balance;

if(balance<0)balance=0;

}

@Override

public String toString() {

return "이름 : " + name + "\n비밀번호 : " + pw + "\nid : " + id + "\n잔액 : " + balance;

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

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

return result;

}

@Override

public boolean equals(Object obj) {//이름으로 중복확인

if (this == obj)

return true;

if (obj == null)

return false;

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

return false;

BankAccountCol other = (BankAccountCol) obj;

if (name == null) {

if (other.name != null)

return false;

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

return false;

return true;

}

}

2.은행 관리 클래스------------------------------------------------------------------------

package bank;

import java.util.ArrayList;

public class BankManageCol {

java.util.Scanner scanner = new java.util.Scanner(System.in);

public String bankNow="";

public boolean isBankAdmin=false;

public boolean isBankExit=false;

public boolean isBankUser=true;

public ArrayList<BankCol> banks=new ArrayList<BankCol>();

public BankManageCol() {}

public int searchBankIndex(String name) {

int index=-1;//bankName기본값과 bankNow기본값이 같아서("")

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

if(banks.get(i).bankName.equals(name)) {

index=i;

break;

}else {

index=-1;//없는 이름

}

}

return index;

}

public boolean isExistBank(int index) {

if(index==-1) {

System.out.println("존재하지 않는 은행입니다");

return false;

}else {

return true;

}

}

public void addBankData(BankCol newdata) {

for(BankCol b:banks) {

if(b.bankName.equals(newdata.bankName)){

System.out.println("이미 존재하는 은행이름입니다");

break;

}

}

banks.add(newdata);

}

public void addBank() {

System.out.println("<<은행 추가>>");

System.out.print("이름 : ");

String newName=scanner.nextLine();

addBankData(new BankCol(newName));

System.out.println("추가됨");

}

public void delBank() {

System.out.println("<<은행 삭제>>");

System.out.print("삭제할 이름 : ");

String delName=scanner.nextLine();

int index=searchBankIndex(delName);

if(isExistBank(index)) {

banks.remove(index);

System.out.println(delName+" 삭제됨");

}

}

public void displayBank() {

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

System.out.println(banks.get(i).toString());

}

}

public void bankAdmin() {

System.out.println("관리자 메뉴");

System.out.print("1.은행추가 2.은행삭제 3.은행정보 4.이전 메뉴로\n>>");

int inputMenu=Integer.parseInt(scanner.nextLine());

switch (inputMenu) {

case 1:

addBank();

break;

case 2:

delBank();

break;

case 3:

displayBank();

break;

case 4:

isBankAdmin=false;

return;

default:

System.out.println("잘못된 입력");

break;

}

}

public void bankLogin() {

System.out.print("원하는 은행을 입력하세요(exit로 종료) : ");

String inputBankName=scanner.nextLine();

if(inputBankName.equals("exit")) {

isBankExit=true;

return;

}

if(inputBankName.equals("admin")) {

isBankAdmin=true;

return;

}

int index = searchBankIndex(inputBankName);

if(isExistBank(index)) {

bankNow=inputBankName;

banks.get(index).runBank();

banks.get(index).isExit=false;

return;

}else {

System.out.println("은행을 추가하겠습니까?");

System.out.print("1: yes 2: no >>");

int inputLogin=Integer.parseInt(scanner.nextLine());

if(inputLogin==1) {

addBank();

}else {

return;

}

}

}

public void runBankManage() {

while(isBankExit==false) {

bankLogin();

while(isBankAdmin) {

bankAdmin();

}

}

System.out.println("은행관리종료");

}

}

3.은행업무 클래스---------------------------------------------------------------

package bank;

import java.util.ArrayList;

public class BankCol {

java.util.Scanner scanner = new java.util.Scanner(System.in);

public ArrayList<BankAccountCol> account = new ArrayList<BankAccountCol>();

public String bankName="";

public String nameLoggedin="";

public int id=0;//사용자마다 부여되는 고유번호

private String adminID="admin";

private String adminPW="1111";

private String exitName="exit";

public boolean isAdmin=false;

public boolean isExit=false;

public BankCol() {}

public BankCol(String bankName) {

this.bankName = bankName;

}

@Override

public String toString() {

return "은행이름 : "+bankName+", 회원수 : "+account.size();

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

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;

BankCol other = (BankCol) obj;

if (bankName == null) {

if (other.bankName != null)

return false;

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

return false;

return true;

}

public int searchIndex(String searchName) {

int index=-1;

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

if(searchName.equals(account.get(i).getName())) {

index=i;

break;

}else {

index=-1;//없는 계정

}

}

return index;

}

public boolean isExist(int index) {

if(index==-1) {

System.out.println("존재하지 않는 이름입니다");

return false;

}else {

return true;

}

}

public void addAccountData(BankAccountCol newdata) {

for(int i=0;i<id;i++) {//사용자명과 id 서로 다른값임, id(고유번호)중복검사

if(i==newdata.getId()) {

System.out.println("id가 중복되었습니다\n다른 id를 부여해주세요");

return;

}

if(newdata.getName().equals(account.get(i).getName())) {

System.out.println("중복된 이름입니다\n다른 이름을 입력해주세요");

return;

}

}

id++;

account.add(newdata);

}

public void addAccount() {

System.out.println("<<계정 추가>>");

System.out.print("이름 : ");

String newName=scanner.nextLine();

System.out.print("비밀번호 : ");

String newPw=scanner.nextLine();

addAccountData(new BankAccountCol(newName,newPw,id,0));

System.out.println("추가됨");

}

public void delAccount() {

System.out.println("<<계정 삭제>>");

System.out.print("삭제할 이름 : ");

String delName=scanner.nextLine();

int index=searchIndex(delName);

if(isExist(index)) {

account.remove(index);

System.out.println(delName+" 삭제됨");

}else {

System.out.println("삭제할 이름과 일치하는 계정이 없습니다");

}

}

public void displayBal(String name) {

System.out.printf("%s의 잔액 : %d원%n",name,account.get(searchIndex(name)).getBalance());

}

public void displayAll() {

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

System.out.println(account.get(i));

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

}

}

public void deposit(String name) {

System.out.println("<<입금>>");

System.out.print("입금할 금액 입력 : ");

int dep = Integer.parseInt(scanner.nextLine());

int index=searchIndex(name);

if(isExist(index)) {

account.get(index).setBalance(account.get(index).getBalance()+dep);

}

}

public void withdraw(String name) {

System.out.println("<<출금>>");

System.out.print("출금할 금액 입력 : ");

int wd = Integer.parseInt(scanner.nextLine());

int index=searchIndex(name);

if(isExist(index)) {

if((account.get(index).getBalance()-wd)>0) {

account.get(index).setBalance(account.get(index).getBalance()-wd);

}else {

System.out.println("잔액이 부족합니다");

}

}

}

public void startMenu() {

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

int loginInput = Integer.parseInt(scanner.nextLine());

switch (loginInput) {

case 1:

System.out.println("<<로그인>>");

login();

break;

case 2:

System.out.println("<<회원가입>>");

addAccount();

break;

default:

System.out.println("잘못된 입력");

break;

}

}

public void adminLogin() {

System.out.print("비밀번호 입력 : ");

String inputPassword = scanner.nextLine();

if (inputPassword.equals(adminPW)) {

System.out.println("관리자 로그인");

isAdmin=true;

return;

} else {

System.out.println("잘못된 비밀번호입니다");

return;

}

}

public void login() {

System.out.print("사용자명 (exit로 종료): ");

String inputUsername = scanner.nextLine();

if(inputUsername.equals(adminID)) {

adminLogin();

return;

}

if(inputUsername.equals(exitName)) {

isExit=true;

return;

}

int index=searchIndex(inputUsername);

if(isExist(index)) {

System.out.print("비밀번호 입력 : ");

String inputPassword = scanner.nextLine();

if(inputPassword.equals(account.get(index).getPw())) {

System.out.printf("%s님으로 로그인%n", account.get(index).getName());

nameLoggedin = account.get(index).getName();

return;

}else {

System.out.println("잘못된 비밀번호");

return;

}

}

}

public void mainMenu() {

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

System.out.print("원하는 메뉴 입력 : ");

int menuInput = Integer.parseInt(scanner.nextLine());

switch (menuInput) {

case 1:

deposit(nameLoggedin);

break;

case 2:

withdraw(nameLoggedin);

break;

case 3:

System.out.println("<<조회>>");

displayBal(nameLoggedin);

break;

case 4:

System.out.printf("%s 로그아웃%n",nameLoggedin);

nameLoggedin="";

return;

default:

System.out.println("잘못된 입력");

break;

}

}

public void adminMenu() {

System.out.println("1.계정추가 2.계정삭제 3.모든사용자정보 4.이름으로 정보 검색 5.종료");

System.out.print("원하는 메뉴 입력 : ");

int menuInput = Integer.parseInt(scanner.nextLine());

switch (menuInput) {

case 1:

addAccount();

break;

case 2:

delAccount();

break;

case 3:

System.out.println("<<전체 정보>>");

displayAll();

break;

case 4:

System.out.println("<<이름으로 정보 검색>>");

System.out.print("검색할 이름 입력 : ");

String searchName=scanner.nextLine();

int index=searchIndex(searchName);

if(isExist(index)) {

System.out.println(account.get(index));

}

break;

case 5:

System.out.println("<<관리자 로그아웃>>");

isAdmin = false;

return;

default:

System.out.println("잘못된 입력");

break;

}

}

public void runBank() {

System.out.printf("%s은행 입니다%n",bankName);

while(isExit==false) {

startMenu();

while(isAdmin==true) {

adminMenu();

}

while(nameLoggedin!="") {

mainMenu();

}

}

System.out.printf("%s은행을 이용해 주셔서 감사합니다%n",bankName);

}

}

4.실행파일-------------------------------------------------------------------------------

package bank;

public class Bank1126 {

public static void main(String[] args) {

java.util.Scanner scanner = new java.util.Scanner(System.in);

BankManageCol bm =new BankManageCol();

bm.runBankManage();

System.out.println("프로그램 종료");

}

}