/\* Anthony Positano / Homework 3

The goal of this assignment is to implement an inheritance structure in the form of bank accounts,

where the user is able to create and update different accounts that inherit from the BankAccount class.

\*/

import java.util.\*;

class BankAccount {

private int accountNum;

private double balance;

public BankAccount() {

accountNum = 0;

balance = 0;

}

public BankAccount(int accountNum, double balance) {

this.accountNum = accountNum;

this.balance = balance;

}

public int getAccountNum() {

return accountNum;

}

public void setAccountNum(int num) {

accountNum = num;

}

public double getBalance() {

return balance;

}

public void deposit(double deposit) {

balance += deposit;

}

public void withdrawal(double withDrw) {

if (withDrw > balance) {

System.out.println("Insufficient Funds");

} else {

balance -= withDrw;

}

}

public String toString() {

return "Account Number: " + Integer.toString(accountNum) + "\n"

+ "Balance: $" + Double.toString(balance);

}

public boolean equals(Object obj) {

if (((BankAccount) obj).accountNum == accountNum) {

return true;

} else {

return false;

}

}

}

class CheckingAccount extends BankAccount { // equals

private int lastCheckNum;

public CheckingAccount() {

lastCheckNum = 1;

}

public CheckingAccount(int accountNum, double amount) {

lastCheckNum = 1;

super.deposit(amount);

super.setAccountNum(accountNum);

}

public int getCheckNum() {

return lastCheckNum;

}

public void setCheckNum(int num) {

lastCheckNum = num;

}

public void deposit(int num, double deposit) {

if (num > lastCheckNum) {

super.deposit(deposit);

lastCheckNum = num;

} else {

System.out.println("Check Number Is Not Valid");

}

}

public String toString() {

return super.toString() + "\nLast Check Number: " + Integer.toString(lastCheckNum);

}

public boolean equals(Object obj) {

if (((CheckingAccount) obj).lastCheckNum == lastCheckNum && super.equals(obj)) {

return true;

} else {

return false;

}

}

}

class InterestAccount extends BankAccount { // toString, equals

private double interestRate;

public InterestAccount() {

interestRate = 0;

}

public InterestAccount(int accountNum, double interestRate, double balance) {

this.interestRate = interestRate;

super.setAccountNum(accountNum);

super.deposit(balance);

}

public void setInterestRate(double interestRate) {

this.interestRate = interestRate;

}

public double getInterestRate() {

return interestRate;

}

public void addInterest() {

double balance = getBalance() \* interestRate;

super.deposit(balance);

}

public String toString() {

return super.toString() + "\nInterest Rate: " + Double.toString(interestRate) + "%";

}

public boolean equals(Object obj) {

if (((InterestAccount) obj).interestRate == interestRate && super.equals(obj)) {

return true;

} else {

return false;

}

}

}

class FixedDepositAccount extends InterestAccount { // toString, equals

private boolean approved;

public FixedDepositAccount() {

approved = false;

}

public FixedDepositAccount(int accountNum, double balance, double interest) {

super.setAccountNum(accountNum);

super.deposit(balance);

super.setInterestRate(interest);

approved = false;

}

public boolean getApproved() {

return approved;

}

public void setApproved(boolean b) {

approved = b;

}

public void addInterest() {

if (approved) {

super.addInterest();

} else {

System.out.println("Account Not Approved");

}

}

public String toString() {

return super.toString() + "\nApproved: " + Boolean.toString(approved);

}

public boolean equals(Object obj) {

if (((FixedDepositAccount) obj).approved == approved && super.equals(obj)) {

return true;

} else {

return false;

}

}

}

public class Main {

public static void printMenu() {

System.out.println("MENU");

System.out.println("1 - Deposit");

System.out.println("2 - Withdrawal");

System.out.println("3 - Add Interest");

System.out.println("4 - Open a Checking Account");

System.out.println("5 - Open an Interest Account");

System.out.println("6 - Open a Fixed Deposit Account");

System.out.println("7 - Change Fixed Deposit Account Permissions");

System.out.println("8 - Print Specific Account Information");

System.out.println("9 - Print All Account Information");

System.out.println("0 - Quit\n");

}

public static void executeMenu(ArrayList<BankAccount> list, Scanner scnr, int userNum, int time) {

if (userNum == 1) {

boolean found = false;

System.out.println("Enter Account Number");

int accountNum = scnr.nextInt();

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

if (list.get(i).getAccountNum() == accountNum) {

System.out.println("Enter Deposit Amount");

double deposit = scnr.nextDouble();

if (list.get(i) instanceof CheckingAccount) {

System.out.println("Enter Check Number");

int num = scnr.nextInt();

((CheckingAccount) (list.get(i))).deposit(num, deposit);

System.out.print("Your New Account Balance is: $");

System.out.printf("%.2f", list.get(i).getBalance());

} else {

list.get(i).deposit(deposit);

}

found = true;

}

}

if (!found) {

System.out.println("Invalid Account Number");

}

} else if (userNum == 2) {

boolean found = false;

System.out.println("Enter Account Number");

int account = scnr.nextInt();

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

if (list.get(i).getAccountNum() == account) {

System.out.println("Enter Withdrawal Amount");

double withDrw = scnr.nextDouble();

list.get(i).withdrawal(withDrw);

found = true;

System.out.print("Your New Account Balance is: $");

System.out.printf("%.2f", list.get(i).getBalance());

}

}

if (!found) {

System.out.println("Invalid Account Number");

}

} else if (userNum == 3) {

boolean found = false;

System.out.println("Enter Account Number");

int num = scnr.nextInt();

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

if (list.get(i).getAccountNum() == num) {

if (list.get(i) instanceof InterestAccount) {

((InterestAccount) (list.get(i))).addInterest();

System.out.println("Your New Account Balance is: $" + list.get(i).getBalance());

} else if (list.get(i) instanceof FixedDepositAccount) {

if (((FixedDepositAccount) (list.get(i))).getApproved()) {

((FixedDepositAccount) (list.get(i))).addInterest();

System.out.print("Your New Account Balance is: $");

System.out.printf("%.2f", list.get(i).getBalance());

found = true;

} else {

System.out.println("Account Not Approved");

found = true;

}

}

}

}

if (!found) {

System.out.println("Invalid Account Number");

}

} else if (userNum == 4) {

System.out.println("Enter Starting Balance");

double balance = scnr.nextDouble();

list.add(new CheckingAccount(time, balance));

System.out.println("Your Account Number is: " + time);

} else if (userNum == 5) {

System.out.println("Enter Interest Rate");

double interestRate = scnr.nextDouble();

System.out.println("Enter Starting Balance");

double balance = scnr.nextDouble();

list.add(new InterestAccount(time, interestRate, balance));

System.out.println("Your Account Number is: " + time);

} else if (userNum == 6) {

System.out.println("Enter Starting Balance");

double balance = scnr.nextDouble();

System.out.println("Enter Interest Rate");

double interest = scnr.nextDouble();

list.add(new FixedDepositAccount(time, balance, interest));

System.out.println("Your Account Number is: " + time);

} else if (userNum == 7) {

boolean found = false;

boolean status = false;

System.out.println("Enter Account Number");

int num = scnr.nextInt();

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

if (list.get(i).getAccountNum() == num && list.get(i) instanceof FixedDepositAccount) {

status = ((FixedDepositAccount) (list.get(i))).getApproved();

status = !status;

((FixedDepositAccount) (list.get(i))).setApproved(status);

found = true;

}

}

if (found) {

if (status) {

System.out.println("Account Permission Approved");

} else {

System.out.println("Account Permission Denied");

}

} else {

System.out.println("Invalid Account Number");

}

} else if (userNum == 8) {

boolean found = false;

System.out.println("Enter Account Number");

int num = scnr.nextInt();

if (list.isEmpty()) {

System.out.println("No Account Information");

} else {

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

if (list.get(i).getAccountNum() == num) {

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

found = true;

}

}

}

if (!found) {

System.out.println("Invalid Account Number");

}

} else if (userNum == 9) {

if (list.isEmpty()) {

System.out.println("No Account Information");

} else {

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

System.out.println(list.get(i).toString() + "\n");

}

}

}

}

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

ArrayList<BankAccount> accountsList = new ArrayList<>();

int userNum = 99;

int time = 1;

while (userNum != 0) {

System.out.println();

printMenu();

userNum = scnr.nextInt();

if (userNum > 0 && userNum < 10) {

executeMenu(accountsList, scnr, userNum, time);

} else if (userNum != 0) {

System.out.println("Invalid Option");

}

time++;

}

System.out.println("Goodbye");

}

}