PAT3 (LP4) Challenge Exercise

Transferring Funds

The Account Class file found at the end of this document (also found in Chapter 4.4 - Listing 4.3 Bank Account Example) contains a definition for a simple bank account class with methods to withdraw, deposit, get the balance and account number, and print a summary. Copy it to your program, compile and run, and study it to see how it works. Then write the following additional code:

1. Add a method public void transfer(Account acct, double amount) to the Account class that allows the user to transfer funds from one bank account to another. If acct1 and acct2 are Account objects, then the call acct1.transfer(acct2,957.80) should transfer $957.80 from acct1 to acct2. Be sure to clearly document which way the transfer goes!

2. Write a class TransferTest with a main method that creates two bank account objects and enters a loop that does the following:

􀀀 Asks if the user would like to transfer from account1 to account2, transfer from account2 to account1, or quit.

􀀀 If a transfer is chosen, asks the amount of the transfer, carries out the operation, and prints the new balance for each account.

􀀀 Repeats until the user asks to quit, then prints a summary for each account.

3. Add a static method to the Account class that lets the user transfer money between two accounts without going through either account. You can (and should) call the method transfer just like the other one - you are overloading this method. Your new method should take two Account objects and an amount and transfer the amount from the first account to the

second account. The signature will look like this:

public static void transfer(Account acct1, Account acct2, double amount)

Modify your TransferTest class to use the static transfer instead of the instance version.

**Account Class**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
// Account.java Represents a bank account with basic services such as deposit  
// and withdraw.  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
  
import java.text.NumberFormat;   
  
public class Account  
{  
 private final double RATE = 0.035; // interest rate of 3.5%   
 private long acctNumber;  
 private double balance;  
 private String name;   
  
 //-----------------------------------------------------------------  
 // Sets up the account by defining its owner, account number,  
 // and initial balance.  
 //-----------------------------------------------------------------  
 public Account(String owner, long account, double initial)  
 {  
 name = owner;  
 acctNumber = account;  
 balance = initial;  
 }   
 //-----------------------------------------------------------------  
 // Deposits the specified amount into the account. Returns the  
 // new balance.  
 //-----------------------------------------------------------------  
 public double deposit(double amount)  
 {  
 balance = balance + amount;  
 return balance;  
 }   
 //-----------------------------------------------------------------  
 // Withdraws the specified amount from the account and applies  
 // the fee. Returns the new balance.  
 //-----------------------------------------------------------------  
 public double withdraw(double amount, double fee)  
 {  
 balance = balance - amount - fee;   
  
 return balance;  
 }   
 //-----------------------------------------------------------------  
 // Adds interest to the account and returns the new balance.  
 //-----------------------------------------------------------------  
 public double addInterest()  
 {  
 balance + = (balance \* RATE);  
 return balance;  
 }   
 //-----------------------------------------------------------------  
 // Returns the current balance of the account.  
 //-----------------------------------------------------------------  
 public double getBalance()  
 {  
 return balance;  
 }   
 //-----------------------------------------------------------------  
 // Returns a one-line description of the account as a string.  
 //-----------------------------------------------------------------  
 public String toString()  
 {  
 NumberFormat fmt = NumberFormat.getCurrencyInstance();  
 return acctNumber + "\t" + name + "\t" + fmt.format(balance);  
 }

// ADD A PUBLIC METHOD CALLED transfer WHICH ACCEPTS 1 ACCOUNT OBJECT   
// AND AN AMOUNT TO TRANSFER. THIS METHOD WILL WITHDRAWL MONEY FROM this  
// ACCOUNT OBJECT AND DEPOSIT INTO THE PASSES ACCOUNT OBJECT

// ADD A PUBLIC STATIC METHOD CALLED transfer WHICH ACCEPTS 2 ACCOUNT OBJECTS   
// AND AN AMOUNT TO TRANSFER. THIS METHOD WILL WITHDRAWL MONEY FROM ONE ACCOUNT   
// OBJECT AND DEPOSIT IT INTO THE OTHER ACCOUNT OBJECT  
  
}

**CREATE A NEW CLASS CALLED TransferTest**

**import** java.util.Scanner;

**public** **class** TransferTest {

**public** **static** **void** main(String[] args) {

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

**int** option = 0;

// CREATE AN ACCOUNT 1 OBJECT

// CREATE AN ACCOUNT 2 OBJECT

**while** (option != 3) {

System.***out***.println("Enter 1 to transfer from Account 1 to Account 2");

System.***out***.println("Enter 2 to transfer from Account 2 to Account 1");

System.***out***.println("Enter 3 to quit");

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

option = scan.nextInt();

**switch**(option) {

**case** 1:

// USE THE STATIC TRANSFER METHOD TO TRANSFER FROM ACCOUNT 1 TO ACCOUNT 2

**break**;

**case** 2:

// USE THE STATIC TRANSFER METHOD TO TRANSFER FROM ACCOUNT 2 TO ACCOUNT 1

**break**;

**case** 3:

**break**;

**default**:

System.***out***.println("You have entered an incorrect option,

please try again.");

}

}

// PRINT OUT THE BALANCE OF ACCOUNT 1

// PRINT OUT THE BALANCE OF ACCOUNT 2

}

}