**LAB #12**

**ABDUL BASIT**

**193227**

**BSCS-6C**

TASK#1

public class AccountRecord {

private int account;

private String firstName;

private String lastName;

private double balance;

private double combinedValue;

public AccountRecord()

{

this(0,"","",0.0);

}

public AccountRecord(int acc, String first, String last, double bal)

{

setAccount(acc);

setFirstName(first);

setLastName(last);

setBalance(bal);

}

public AccountRecord(int acc, double tran){

setAccount(acc);

setBalance(tran);

}

public void setAccount(int acc){

account = acc;

}

public int getAccount(){

return account;

}

public void setFirstName(String first){

firstName = first;

}

public String getFirstName(){

return firstName;

}

public void setLastName(String last){

lastName = last;

}

public String getLastName(){

return lastName;

}

public void setBalance(double bal){

balance = bal;

}

public double getBalance(){

return balance;

}

public void combine(TransactionRecord record){

combinedValue = balance + record.getAmount();

}

public double getCombinedValue(){

return combinedValue;

}

}

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Formatter;

public class DataFile {

private Formatter file1;

private Formatter file2;

private AccountRecord masterRecords[];

private AccountRecord transactionRecords[];

public DataFile() throws FileNotFoundException{

try{

file1 = new Formatter("oldmast.txt");

file2 = new Formatter("trans.txt");

}

catch(FileNotFoundException filenotfound){

System.err.format("%s", "File can not be found");

System.exit(1);

}

catch(SecurityException authorization){

System.err.format("%s" , "This file is immutable");

System.exit(1);

}

masterRecords = new AccountRecord[4];

masterRecords[0] = new AccountRecord(100,"Alan", "Jones", 348.17);

masterRecords[1] = new AccountRecord(300,"Mary", "Smith", 27.19);

masterRecords[2] = new AccountRecord(500,"Sam", "Sharp", 0.00);

masterRecords[3] = new AccountRecord(700,"Suzy", "Green", -14.22);

transactionRecords = new AccountRecord[4];

transactionRecords[0] = new AccountRecord(100, 27.14);

transactionRecords[1] = new AccountRecord(300, 62.11);

transactionRecords[2] = new AccountRecord(400, 100.56);

transactionRecords[3] = new AccountRecord(900, 82.17);

writingFile1();

writingFile2();

}

public void writingFile1(){

file1.format("%d %s %s %f%n", masterRecords[0].getAccount(), masterRecords[0].getFirstName(),masterRecords[0].getLastName(),masterRecords[0].getBalance());

file1.format("%d %s %s %f%n", masterRecords[1].getAccount(), masterRecords[1].getFirstName(),masterRecords[1].getLastName(),masterRecords[1].getBalance());

file1.format("%d %s %s %f%n", masterRecords[2].getAccount(), masterRecords[2].getFirstName(),masterRecords[2].getLastName(),masterRecords[2].getBalance());

file1.format("%d %s %s %f%n", masterRecords[3].getAccount(), masterRecords[3].getFirstName(),masterRecords[3].getLastName(),masterRecords[3].getBalance());

file1.close();

}

public void writingFile2(){

file2.format("%d %f %n", transactionRecords[0].getAccount(), transactionRecords[0].getBalance() );

file2.format("%d %f%n", transactionRecords[1].getAccount(), transactionRecords[1].getBalance());

file2.format("%d %f%n", transactionRecords[2].getAccount(), transactionRecords[2].getBalance());

file2.format("%d %f%n", transactionRecords[3].getAccount(), transactionRecords[3].getBalance());

file2.close();

}

public Formatter getFile1(){

return file1;

}

public Formatter getFile2(){

return file2;

}

public AccountRecord[] getMasterRecords(){

return masterRecords;

}

public AccountRecord[] getTransactionRecords(){

return transactionRecords;

}

}

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Formatter;

import java.util.Scanner;

public class FileMatch {

private Scanner readF1;

private Scanner readF2;

private DataFile data;

private AccountRecord record;

private Formatter newM,logF;

public FileMatch() throws FileNotFoundException{

record = new AccountRecord();

data = new DataFile();

newM = new Formatter("newM.txt");

logF = new Formatter("log.txt");

try {

readF1 = new Scanner(new File("trans.txt"));

readF2 = new Scanner(new File("oldmast.txt"));

} catch (FileNotFoundException ex) {

System.err.format("File cannot be found");

}

compare();

}

public void compare(){

for(AccountRecord master : data.getMasterRecords()){

for(AccountRecord trans : data.getTransactionRecords()){

if(master.getAccount() == trans.getAccount()){

newM.format("%d %f%n",master.getAccount(), master.getBalance() + trans.getBalance());

newM.flush();

break;

}

else if(master.getAccount() != trans.getAccount() && checkTrans(master) == false){

newM.format("%d %n" , master.getAccount());

newM.flush();

break;

}

}

}

populateLog();

}

public boolean checkTrans(AccountRecord master){

for(AccountRecord trans : data.getTransactionRecords())

if(master.getAccount() == trans.getAccount())

return true;

return false;

}

public void populateLog(){

for(AccountRecord transaction : data.getTransactionRecords()){

boolean hold = false;

for(AccountRecord master : data.getMasterRecords())

if(transaction.getAccount() == master.getAccount())

hold = true;

if(hold == false){

logF.format("Unmatched Transaction Record for account Number :%d%n" , transaction.getAccount());

logF.flush();

continue;

}

}

}

}

public class TransactionRecord {

private int accountNumber;

private double amount;

public double getAmount(){

return amount;

}

public int getAccountNumber(){

return accountNumber;

}

public void setAmount(double am){

amount = am;

}

public void setAccountNumber(int accNum){

accountNumber = accNum;

}

}

import java.io.FileNotFoundException;

public class Main {

public static void main(String[] args) throws FileNotFoundException{

FileMatch file = new FileMatch();

}

}

**OUTPUT:**







