Documentation for Program

David G. Fields

Seth Riley-Nelson

Henry Onesengmani

Professor: Ms. Forde

Project #1 BankTestDriver

CSC 225-800 Advanced JAVA Programming

Summer 2014

Table of Contents

[Program Abstract 2](#_Toc369190065)

[Program Design 2](#_Toc369190066)

[UML Diagram 2](#_Toc369190067)

[Source Code 2](#_Toc369190068)

[Input 10](#_Toc369190069)

[Assigned Test Data 10](#_Toc369190070)

[Output 13](#_Toc369190071)

[Assigned Test Run 13](#_Toc369190072)

# Program Abstract

This program will be used with the Customer program to manage the accounts of customers for ES&L Banking.

# Program Design

## UML Diagram

|  |
| --- |
| **BankTestDriver** |
| **- FEE 1.50**  **- MAX 30**  + index  ~ name  ~ idNumber  ~ balance  ~ phoneNumber |
| + BankTestDriver()  + parse.Double(String):void  + parse.Int(String):void  + showInputDialog(String):void  + showMessageDialog(String):void  +readLine():String  + inFile.close():void  + System.out.println(String):void  + System.exit(0):void |

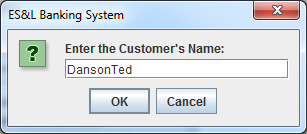
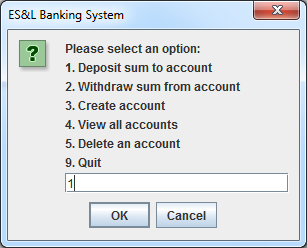
# Source Code

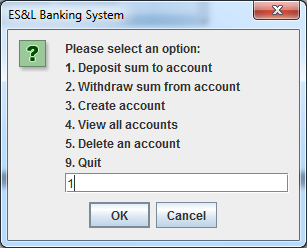
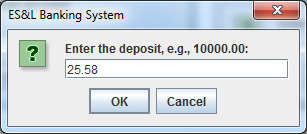
/\*\*Program CSC 225 Prog   
\*Course Title: Advanced JAVA Programming  
\*Course Number: CSC 225-800  
\*Instructor: Ms. Christine Forde  
\*@authors David Fields, Seth Riley-Nelson, Henry Onesengmani  
\*@version 1.0, 06/18/2014  
\*  
\*Description: Program CSC 225 Prog 1 BankTestDriver  
\*Will be used with the Customer program to manage the accounts  
\*of ES&L Bank customers.  
\*  
\*Input:We will input a deposit or withdrawal amount. We will have the  
\*ability to create or delete an account.  
\*  
\*Compute:The program will compute banking fees,and interest earned   
\* on different transactions.   
\*  
\*  
\*@authors David Fields, Seth Riley-Nelson, Henry Onesengmani  
\*@version 1.0, 9/24/2013  
\*/  
  
  
import java.util.\*;  
import java.io.\*;  
import javax.swing.\*;  
import java.text.\*;  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

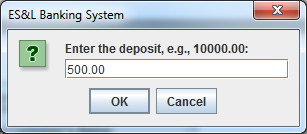
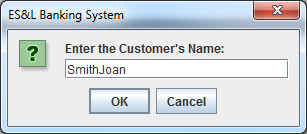
Due Date:06/18/2014<p>  
Program Description: BankTestDriver - Client class to be used with the   
service class Customer, to manage the accounts of ES&L Bank customers.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
public class BankTestDriver  
{  
 //----------------------------------------------------------------------  
 // Reads data from a Customer input file (customer.txt). Put the records  
 // in an array. Display the unsorted records to the user. Present a menu  
 // of choices to the user:1) deposit sum, 2) withdraw amount,3) create a  
 // new customer, 4) view all customers, 5) delete a customer 9) Quit  
 // At end of file, sort the array in alphabetical order by name and  
 // display all customer's accounts,  
 //-----------------------------------------------------------------  
  
 private NumberFormat fmt = NumberFormat.getCurrencyInstance();  
 final double FEE = 1.50;  
 final int MAX = 30;  
  
 Customer [] custsArray = new Customer[MAX];  
 int count = 0; //keeps track of the number of objects placed in  
 // the custArray array  
  
 int index; // points to the current object in the array that  
 // is being accessed  
  
 StringTokenizer tokenizer;  
 String line;  
 String file = "Customer.txt";  
  
 String name;  
 String idNumber;  
 double balance;  
 String phoneNumber;  
  
  
 boolean quit = false;  
 int choice;  
  
 String OutPutText = "";  
 String xnam;  
  
//When you run it, is the customer information supposed to show? Like right after you run it  
  
 /\*\*  
 \*The runBankTest method runs the BankTest system.  
 \*  
 \* @param args command line arguments - ignored  
 \* It reads the input file and store the records in custsArray  
 \* The method contains a loop that prompts the users with a  
 \* JOptionPane.showInput dialog window for choices.  
 \*/  
//--------------------------------------------------------------------  
  
public void runBankTest()  
{  
 System.out.println("Welcome to the ES&L Bank");  
  
 try  
 {  
 FileReader fr = new FileReader (file);  
 BufferedReader inFile = new BufferedReader (fr);  
  
 line = inFile.readLine();  
  
 while (line != null && count < MAX)  
 {  
 tokenizer = new StringTokenizer (line);  
 name = tokenizer.nextToken();  
  
  
 try  
 {  
 idNumber = tokenizer.nextToken();  
  
 balance = Double.parseDouble(tokenizer.nextToken());  
  
 phoneNumber = tokenizer.nextToken();  
  
 custsArray [count++] = new Customer(name,  
 idNumber,  
 balance,  
 phoneNumber);  
  
  
 }// end inner try block  
  
  
  
 catch (NumberFormatException exception)  
 {  
 System.out.println(" \t\tUnsorted customer.txt records");  
 System.out.println( "Error in input. Line ignored: ");  
 System.out.println (line);  
 } // end catch block  
  
 line = inFile.readLine(); // read another record  
  
 } //end while block  
  
 inFile.close();  
  
   
 System.out.println("");  
  
 for (int scan = 0; scan < count; scan++)  
 System.out.println (custsArray[scan]);  
  
  
  
//--------------------------------------------------------------  
// Get choices from the user. Loop until the users enteres a 9  
  
  
 while (!quit)  
 {  
 String strChoice =  
 JOptionPane.showInputDialog(null,"Please select an option: \n"  
 + "\t1. Deposit sum to account\n"  
 + "\t2. Withdraw sum from account\n"  
 + "\t3. Create account\n"  
 + "\t4. View all accounts\n"  
 + "\t5. Delete an account\n"  
 + "\t9. Quit", "ES&L Banking System",   
 JOptionPane.QUESTION\_MESSAGE);  
  
 choice = Integer.parseInt(strChoice);  
//-----------------------------------------------------------------------  
 switch (choice)  
 {  
 case 1://deposit  
 xnam =  
 JOptionPane.showInputDialog  
 (null,"Enter the Customer's Name: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 index = custsArray[0].findIndex(custsArray, xnam, count);  
  
 if (index != -1)  
 {  
 System.out.println("");  
  
 String strDepositAmt =  
 JOptionPane.showInputDialog  
 (null,"Enter the deposit, e.g., 10000.00: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 double depositAmt = Double.parseDouble(strDepositAmt);  
  
 custsArray[index].deposit(depositAmt);  
  
 JOptionPane.showMessageDialog  
 (null, xnam + " balance after deposit: " +  
 fmt.format(custsArray[index].getBalance() )  
 + "\n" +  
 xnam + " balance after interest is added: " +  
 fmt.format(custsArray[index].addInterest() ),  
 "ES&L Banking System",  
 JOptionPane.INFORMATION\_MESSAGE);  
  
  
 } // end if stmt  
  
 else  
 {  
 System.out.println("");  
 System.out.println( xnam + " was not found");  
 } //end else stmt  
  
 break; // end choice equals 1  
  
  
//---------------------------------------------------------------  
 case 2: //withdraw  
  
 //viewCustomers();  
 xnam =  
 JOptionPane.showInputDialog(null,"Enter the Customer's Name: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 index = custsArray[0].findIndex(custsArray, xnam, count);  
  
 if (index != -1)  
 {   
 System.out.println("");  
  
 String strWithdrawAmt =  
 JOptionPane.showInputDialog  
 (null,"Enter the withdrawal, e.g., 10.00: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
   
 double withdrawAmt = Double.parseDouble(strWithdrawAmt);  
  
 JOptionPane.showMessageDialog  
 (null, xnam + " balance after withdrawal: " +  
 fmt.format(custsArray[index].withdraw(withdrawAmt) )  
 + "\n" + xnam + " balance after interest is added: " +  
 fmt.format(custsArray[index].addInterest() ),  
 "ES&L Banking System",  
 JOptionPane.INFORMATION\_MESSAGE);  
  
 } // end if stmt  
  
   
 else  
 {  
 System.out.println("");  
 System.out.println( xnam + " was not found");  
 } //end else stmt  
  
   
   
 break; //end choice equals 2  
  
//-------------------------------------------------------------  
  
 case 3: //create an account  
  
 if (count < custsArray.length)  
 {  
 name = JOptionPane.showInputDialog  
 (null,"Enter Customer's name: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 idNumber = JOptionPane.showInputDialog  
 (null,"Enter Customer's Number, e.g., 11111: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 String strBalance = JOptionPane.showInputDialog  
 (null,"Enter Customer's Balance, e.g., 1000.00: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 balance = Double.parseDouble(strBalance);  
  
 phoneNumber = JOptionPane.showInputDialog  
 (null,"Enter Customer's phone number: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
  
 //--------------------------------------------------------------------  
 custsArray[count]= new Customer();  
 custsArray[count].addNewCustomer(custsArray, name, count,  
 idNumber, balance,  
 phoneNumber);  
 count++;  
 }  
  
 else  
 JOptionPane.showMessageDialog(null,  
 "The array is full. No new record added ",  
 "ES&L Banking System",  
 JOptionPane.INFORMATION\_MESSAGE);  
  
 break; //end choice equals 3  
  
  
//------------------------------------------------------------------  
 case 4: //view all accounts  
  
 //sort for output by names  
 custsArray[0].nameSort(custsArray,count);  
  
 OutPutText = "";  
  
 for (int scan = 0; scan < count; scan++)  
 {  
 OutPutText =  
 (OutPutText + custsArray[scan].getName() + " " +  
 custsArray[scan].getIdNumber() + " " +  
 fmt.format(custsArray[scan].getBalance()) + " " +  
 custsArray[scan].getPhoneNumber()+ "\n");  
  
 }  
  
 JOptionPane.showMessageDialog(null,OutPutText,  
 "ES&L Banking System",  
 JOptionPane.INFORMATION\_MESSAGE);  
  
 break; // end choice 4  
 //-----------------------------------------------------------------  
 case 5: //Delete a customer  
  
 xnam =  
 JOptionPane.showInputDialog(null,"Enter the Customer's Name: ",  
 "ES&L Banking System",JOptionPane.QUESTION\_MESSAGE);  
  
 index = custsArray[0].findIndex(custsArray, xnam, count);  
   
 count = custsArray[0].deleteCustomer(custsArray, index, count, xnam);  
  
 break; // end choice 5  
  
 //-----------------------------------------------------------------  
  
 default: quit = true; //set done to true to exit the system  
  
//--------------------------------------------------------------------  
  
 }// end switch statement  
  
 }// end while loop for menu  
  
  
  
  
  
//----------------------------------------------------------------------  
  
 custsArray[0].nameSort(custsArray,count);//sort for output by names  
  
 System.out.println("");  
 System.out.println(" \t\tSorted updated customer.txt records");  
  
 for (int scan = 0; scan < count; scan++)  
 System.out.println (custsArray[scan]);  
  
  
 }//end try block  
  
//----------------------------------------------------------------  
  
 catch (FileNotFoundException exception)  
 {  
 System.out.println ("The file " + file + " was not found");  
 }  
  
  
  
 catch (IOException exception)  
 {  
 System.out.println (exception);  
 }  
  
  
//----------------------------------------------------------  
  
  
  
} //end method runBankTest()  
  
//------------------------------------------------------------  
  
  
public static void main (String[] args)  
 {  
 BankTestDriver bankTest = new BankTestDriver();  
 bankTest.runBankTest();  
 System.exit(0);  
 } // end main method  
  
//---------------------------------------------------------------  
  
}// end class BankTestDriver  
Sample Run

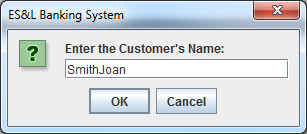
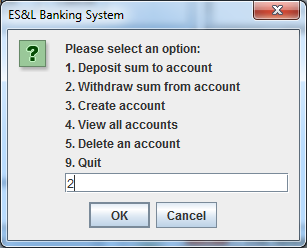
## Input

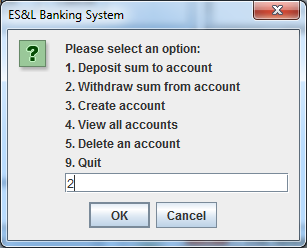
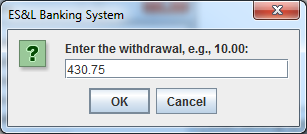
### Assigned Test Data

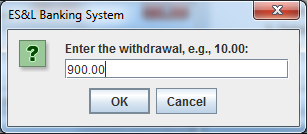
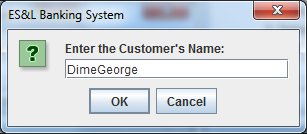


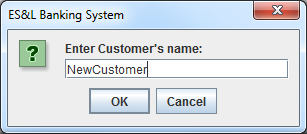
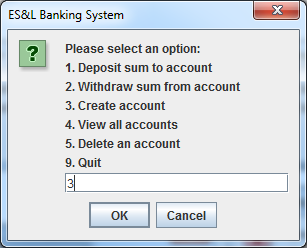
Ï

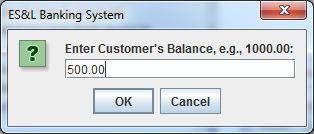
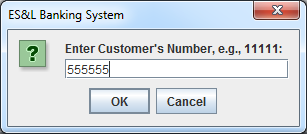


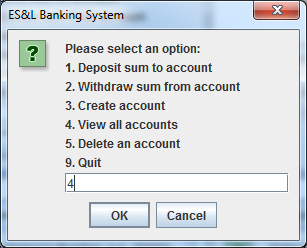
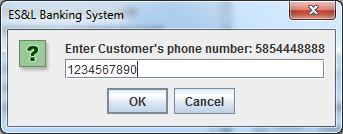


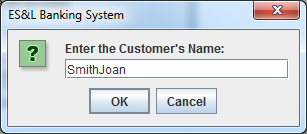
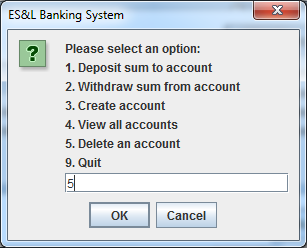












## Output

### Assigned Test Run

