using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace BankingApplication

{

public class bankclass

{

//creation of variables accountno,accountholdername,accountholderaddress,accountphonenumber,accounttype,balanceamount

private long accountno;

public long Accountno

{

get { return accountno; }

set { accountno = value; }

}

public long accountnodump = 0;

private string accountholdername;

public string Accountholdername

{

get { return accountholdername; }

set { accountholdername = value; }

}

private string accountholderaddress;

public string Accountholderaddress

{

get { return accountholderaddress; }

set { accountholderaddress = value; }

}

private string accountholderphonenumber;

public string Accountholderphonenumber

{

get { return accountholderphonenumber; }

set { accountholderphonenumber = value; }

}

private string accounttype;

public string Accounttype

{

get { return accounttype; }

set { accounttype = value; }

}

private long balanceamount;

public long Balanceamount

{

get { return balanceamount; }

set { balanceamount = value; }

}

public long balanceamountdump = 500;

//creation of an empty constructor

public bankclass()

{

}

//creation of an parameter constructor

public bankclass(long accountno, string accountholdername, string accountholderaddress, string accountholderphonenumber, string accounttype, long balanceamount)

{

this.accountno = accountno;

this.accountholdername = accountholdername;

this.accountholderaddress = accountholderaddress;

this.accountholderphonenumber = accountholderphonenumber;

this.accounttype = accounttype;

this.balanceamount = balanceamount;

}

}

//creation of an interface which contains the methods for mainmenu,addnewaccount,withdrawamount,deposit,balanceenquiry and exit methods

public interface allmethods

{

string mainmenu();

void addnewaccount();

void balanceenquiry(allmethodsdetails allmethoddetailsbalanceobj);

void withdrawamount(allmethodsdetails allmethoddetailswithdrawobj);

void depositamount(allmethodsdetails allmethoddetailsdepositobj);

void exit();

}

//creation of a class which contains all methods details

public class allmethodsdetails : allmethods

{

//creation of an object for a bankclass

bankclass bankclassobj = new bankclass();

//creation of a list for bankclass

public static List<bankclass> mybankclassobj = new List<bankclass>();

//creation of an object to a Program class

Program pgmobj = new Program();

//creation of a mainmenu method

public string mainmenu()

{

Console.WriteLine();

Console.WriteLine(" MAIN MENU WIZARD");

Console.WriteLine("Please select any of the options as per your requirement");

Console.WriteLine(" 1 : Add a new account");

Console.WriteLine(" 2 : Balance Enquiry");

Console.WriteLine(" 3 : Withdraw the amount");

Console.WriteLine(" 4 : Deposit the amount");

Console.WriteLine(" 5 : Exit from the wizard");

Console.WriteLine();

Console.Write("Enter the number : ");

string inputvalue = Console.ReadLine();

return inputvalue;

}

//creation of addnewaccount method

public void addnewaccount()

{

Console.WriteLine();

Console.WriteLine(" ADDING A NEW ACCOUNT WIZARD ");

Console.WriteLine(" Please enter your details");

bankclassobj.accountnodump++;

bankclassobj.Accountno = bankclassobj.accountnodump;

Console.Write(" Enter your name : ");

bankclassobj.Accountholdername = Console.ReadLine();

Console.Write(" Enter your address : ");

bankclassobj.Accountholderaddress = Console.ReadLine();

Console.Write(" Enter your Phonenumber : ");

bankclassobj.Accountholderphonenumber = Console.ReadLine();

Console.Write(" Enter which type of account do you want : ");

bankclassobj.Accounttype = Console.ReadLine();

bankclassobj.Balanceamount = bankclassobj.balanceamountdump;

bankclass bankclassobj2 = new bankclass(bankclassobj.Accountno, bankclassobj.Accountholdername, bankclassobj.Accountholderaddress, bankclassobj.Accountholderphonenumber, bankclassobj.Accounttype, bankclassobj.Balanceamount);

setlist(bankclassobj2);

getlist();

Console.Write(" Do you want to continue ? : y/n --> ");

string status = Console.ReadLine();

if (status == "y")

{

addnewaccount();

}

else if(status == "n")

{

pgmobj.callingthemethod();

}

}

//creation of a setlist method which adds all the values into a list

public void setlist(bankclass bobj1)

{

mybankclassobj.Add(bobj1);

}

//creation of a getlist method which adds the values in to an list array

public bankclass[] getlist()

{

return mybankclassobj.ToArray();

}

//creation of balanceenquiry method

public void balanceenquiry(allmethodsdetails allmethoddetailsobj)

{

allmethodsdetails allobj = new allmethodsdetails();

allobj = allmethoddetailsobj;

bankclass[] bankclasobj = allobj.getlist();

Console.WriteLine();

Console.WriteLine(" BALANCE ENQUIRY WIZARD");

Console.WriteLine();

Console.Write(" Please enter the Account number : ");

string x = Console.ReadLine();

int a = int.Parse(x);

foreach (bankclass obj in bankclasobj)

{

if (obj.Accountno == a)

{

Console.WriteLine(" Your Account Number is : " + obj.Accountno);

Console.WriteLine(" Your Account Holder Name is : " + obj.Accountholdername);

Console.WriteLine(" Your Account Holder Address is : " + obj.Accountholderaddress);

Console.WriteLine(" Your Account Holder Phonenumber is : " + obj.Accountholderphonenumber);

Console.WriteLine(" Your Account Type is : " + obj.Accounttype);

Console.WriteLine(" Your Account Balance is : " + obj.Balanceamount);

}

}

Console.WriteLine(" Do you want to continue ? y / n --> ");

string status = Console.ReadLine();

if (status == "y")

{

balanceenquiry(allobj);

}

else if (status == "n")

{

pgmobj.callingthemethod();

}

}

//creation of withdrawamount method

public void withdrawamount(allmethodsdetails allmethoddetailswithdrawobj)

{

allmethodsdetails allobj = new allmethodsdetails();

allobj = allmethoddetailswithdrawobj;

bankclass[] bankclasobj = allobj.getlist();

Console.WriteLine();

Console.WriteLine(" WITHDRAWING AMOUNT WIZARD");

Console.WriteLine();

Console.Write(" Please enter the Account number : ");

string x = Console.ReadLine();

int a = int.Parse(x);

long balance = 0;

foreach (bankclass obj in bankclasobj)

{

if (obj.Accountno == a)

{

Console.WriteLine(" Hi , " + obj.Accountholdername );

Console.WriteLine(" Your Account Balance is : " + obj.Balanceamount);

Console.WriteLine();

balance = obj.Balanceamount;

if (balance >= 300)

{

Console.Write(" How much amount you want to withdraw : ");

string amountremoved = Console.ReadLine();

long amountremove = Int64.Parse(amountremoved);

balance = balance - amountremove;

if (balance >= 300)

{

obj.Balanceamount = balance;

Console.WriteLine(" Your Account Balance is : " + obj.Balanceamount);

}

else

{

Console.WriteLine(" SORRY you can't withdraw the amount since your balance is less than the minimum balance");

}

}

else

{

Console.WriteLine("SORRY you can't withdraw the amount since your balance is less than the minimum balance");

}

}

}

Console.WriteLine(" Do you want to continue ? y / n --> ");

string status = Console.ReadLine();

if (status == "y")

{

withdrawamount(allobj);

}

else if (status == "n")

{

pgmobj.callingthemethod();

}

}

//creation of depositamount method

public void depositamount(allmethodsdetails allmethoddetailsdepositobj)

{

allmethodsdetails allobj = new allmethodsdetails();

allobj = allmethoddetailsdepositobj;

bankclass[] bankclasobj = allobj.getlist();

Console.WriteLine();

Console.WriteLine(" DEPOSIT AMOUNT WIZARD");

Console.WriteLine();

Console.Write(" Please enter the Account number : ");

string x = Console.ReadLine();

int a = int.Parse(x);

long balance = 0;

foreach (bankclass obj in bankclasobj)

{

if (obj.Accountno == a)

{

Console.WriteLine(" Hi , " + obj.Accountholdername);

Console.WriteLine(" Your Account Balance is : " + obj.Balanceamount);

balance = obj.Balanceamount;

Console.WriteLine();

Console.Write(" How much amount you want to deposit : ");

string amountdeposited = Console.ReadLine();

long amountdeposit = Int64.Parse(amountdeposited);

balance = balance + amountdeposit;

obj.Balanceamount = balance;

Console.WriteLine(" Congrats your amount has been deposited");

Console.WriteLine(" Present balance is : " + obj.Balanceamount);

}

}

Console.WriteLine(" Do you want to continue ? y / n --> ");

string status = Console.ReadLine();

if (status == "y")

{

depositamount(allobj);

}

else if (status == "n")

{

pgmobj.callingthemethod();

}

}

//creation of exit method

public void exit()

{

System.Environment.Exit(0);

}

}

class Program

{

public void callingthemethod()

{

allmethodsdetails allmethodsdetailsobjinmain = new allmethodsdetails();

string inputvalue = allmethodsdetailsobjinmain.mainmenu();

int input = int.Parse(inputvalue);

switch (input)

{

case 1:

{

allmethodsdetailsobjinmain.addnewaccount();

break;

}

case 2:

{

allmethodsdetailsobjinmain.balanceenquiry(allmethodsdetailsobjinmain);

break;

}

case 3:

{

allmethodsdetailsobjinmain.withdrawamount(allmethodsdetailsobjinmain);

break;

}

case 4:

{

allmethodsdetailsobjinmain.depositamount(allmethodsdetailsobjinmain);

break;

}

case 5:

{

allmethodsdetailsobjinmain.exit();

break;

}

default:

{

Console.WriteLine();

Console.WriteLine("Enter the correct choice by seeing the menu again");

Console.WriteLine();

callingthemethod();

break;

}

}

}

static void Main(string[] args)

{

Program pgm = new Program();

pgm.callingthemethod();

}

}

}