1. In Assignment 3 of previous session, print the details of Managers with the help ofdelegate

EmployeeDelegate. (UniCast Delegate)

1. Create a MultiCast Delegate by adding address of another method to the above delegate to list the details of MarketingExecutive.
2. Write a console Application for banking domain, having Accountclass with data members asaccountnumber,customername,balance.ItshouldhaveWithdrawand Deposit methods for performing banking transaction. It should also define UnderBalance, BalanceZeroevents. These events would be raised when balance of account is less than certain amount and equal to zerorespectively.

## Now try these to get a completegrip…

1. Writeseparate ICICIBankclass with EventHandlersfor UnderBalanceand

BalanceZeroEvents.

* 1. Show message "Transaction cannot be continued as balance is insufficient/zero in Account".

1. Writeseparate HDFCBankclasswith EventHandlersforUnderBalanceand

BalanceZeroEvents.

* 1. EventHandler should allow account holder to withdraw money till balance goes to Rs - 1000;below this show message "Transaction cannot be continued below specified limit of Rs.-1000".

## By the way

* Delegates are used ascallbacks.
* Multicast delegate helps to execute a series of tasks by executing methods added toit.
* Events can be declared in a class and raised in any method of a class. But they have to be handled by client code that uses theclass.

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Runtime.CompilerServices;

usingSystem.Text;

usingSystem.Threading.Tasks;

using static System.Console;

namespace Assignment6ques1\_delegate

{

public delegate void DispEmp(intemp\_id, string emp\_name

, float salary);

public class Program

{

public static void DisplayMaanagerDetails(intemp\_id, string emp\_name, float Salary)

{

float PA = 8 \* Salary / 100;

float FA = 13 \* Salary / 100;

float OA = Salary / 100;

floatGrossSalary = Salary + PA + FA + OA;

WriteLine("Manager Details...");

WriteLine("emp\_id: {0}\nemp\_name: {1}\nGrossSalary:{2}", emp\_id, emp\_name, GrossSalary);

}

static void Main(string[] args)

{

DispEmpobj = new DispEmp(Program.DisplayMaanagerDetails);

obj(100, "Priyanshu Gupta", 5000.90F);

ReadLine();

}

}

// question 2

public class Employee

{

public static void DisplayMarketicExeDetails(intemp\_id, string emp\_name, float Salary)

{

WriteLine("Enter Distance travelled during Marketing Executive in km");

int Distance = int.Parse(ReadLine());//taking distance input

intTourAllowances = 5 \* Distance; //5 rupees per km

intTelephone\_Allowances = 1000;

floatGrossSalary = Salary + TourAllowances + Telephone\_Allowances;

WriteLine("Marketing Executive Details...");

WriteLine("emp\_id: {0}\nemp\_name: {1}\nGrossSalary:{2}", emp\_id, emp\_name, GrossSalary);

}

public static void DisplayMaanagerDetails(intemp\_id, string emp\_name, float Salary)

{

float PA = 8 \* Salary / 100;

float FA = 13 \* Salary / 100;

float OA = Salary / 100;

floatGrossSalary = Salary + PA + FA + OA;

WriteLine("Manager Details...");

WriteLine("emp\_id: {0}\nemp\_name: {1}\nGrossSalary:{2}", emp\_id, emp\_name, GrossSalary);

}

static void Main(string[] args)

{

DispEmpobj = new DispEmp(Employee.DisplayMaanagerDetails);

obj += DisplayMarketicExeDetails;

obj(100, "Priyanshu Gupta", 5000.90F);

ReadLine();

}

}

//question 3

public class Account

{

public string account\_number;

public string customer\_name;

public double balance;

public Account(string acc\_num, string cus\_name)

{

account\_number = acc\_num;

customer\_name = cus\_name;

balance = 0.00D;

}

}

public class AccountDetails

{

//Account account = new Account("89674532", "Uma BAharti");

public double GetMoney()

{

double amount = 0;

Console.WriteLine("Amount");

amount = double.Parse(ReadLine());

return amount;

}

public void ShowAccountDetails(Account ac)

{

Console.WriteLine("===============================================");

Console.WriteLine("Account holder number: {0} ", ac.account\_number);

Console.WriteLine("Account holder name : {0} ", ac.customer\_name);

Console.WriteLine("Customer has balance : {0:F} ", ac.balance);

Console.WriteLine("=============================================");

}

public static int Main()

{

byte choice = 0;

double amount = 0;

Account accounholder = new Account("234567", "Priya");

AccountDetailsacholde = new AccountDetails();

Console.WriteLine("Enter customer's initial deposits");

try

{

accounholder.balance = acholde.GetMoney();

}

catch (Exception e) { Console.WriteLine(e.Message); }

acholde.ShowAccountDetails(accounholder);

do

{

WriteLine("1. Show Account details");

WriteLine("2. Deposit Money");

WriteLine("3. Withdraw Money");

WriteLine("Enter Your Choice");

choice = byte.Parse(ReadLine());

// Console.Clear();

switch (choice)

{

case 1:

acholde.ShowAccountDetails(accounholder);

// WriteLine("click enter key for next choice");

// Console.ReadKey();

break;

case 2:

WriteLine("Enter Your deposit");

amount = double.Parse(ReadLine());

accounholder.balance = accounholder.balance + amount;

// Console.Clear();

// acholde.ShowAccountDetails(accounholder);

break;

case 3:

Console.WriteLine("Enter Withdrawal");

amount = acholde.GetMoney();

try

{

if (amount >accounholder.balance)

{

WriteLine("You cannot withdraw money there is insufficient" +

"balance in your account");

throw new Exception("under balance");

}

else if (amount == 0)

{

throw new Exception("zero balance");

}

}

catch (Exception e) { Console.WriteLine(e.Message); }

accounholder.balance -= amount;

//Console.Clear();

// acholde.ShowAccountDetails(accounholder);

break;

default:

WriteLine("Enter valid choice");

break;

}

ReadLine();

} while (choice != 0);

return 0;

}

}

// question 4

public class ICICBANK

{

public string account\_number;

public string customer\_name;

public double balance;

public ICICBANK(string acc\_num, string cus\_name)

{

account\_number = acc\_num;

customer\_name = cus\_name;

balance = 0.00D;

}

}

public class AccountDetails

{

Account account = new Account("89674532", "Uma BAharti");

public double GetMoney()

{

double amount = 0;

Console.WriteLine("Amount");

amount = double.Parse(ReadLine());

return amount;

}

public void ShowAccountDetails(ICICBANK ac)

{

Console.WriteLine("===============================================");

WriteLine("ICICIBANK................");

Console.WriteLine("Account holder number: {0} ", ac.account\_number);

Console.WriteLine("Account holder name : {0} ", ac.customer\_name);

Console.WriteLine("Customer has balance : {0:F} ", ac.balance);

Console.WriteLine("=============================================");

}

public static int Main()

{

byte choice = 0;

double amount = 0;

ICICBANK accounholder = new ICICBANK("234567", "Priya");

AccountDetailsacholde = new AccountDetails();

Console.WriteLine("Enter customer's initial deposits");

try

{

accounholder.balance = acholde.GetMoney();

}

catch (Exception e) { Console.WriteLine(e.Message); }

acholde.ShowAccountDetails(accounholder);

do

{

WriteLine("1. Show Account details");

WriteLine("2. Deposit Money");

WriteLine("3. Withdraw Money");

WriteLine("Enter Your Choice");

try

{

choice = byte.Parse(ReadLine());

}

catch (Exception e) { WriteLine(e.Message); }

Console.Clear();

switch (choice)

{

case 1:

acholde.ShowAccountDetails(accounholder);

WriteLine("click enter key for next choice");

Console.ReadKey();

break;

case 2:

WriteLine("Enter Your deposit");

amount = double.Parse(ReadLine());

accounholder.balance = accounholder.balance + amount;

Console.Clear();

acholde.ShowAccountDetails(accounholder);

break;

case 3:

Console.WriteLine("Enter Withdrawal");

amount = acholde.GetMoney();

try

{

if (amount >accounholder.balance)

{

WriteLine("Transaction cannot be continued as " +

"balance is insufficient");

throw new Exception("under balance");

}

else if (amount == 0)

{

WriteLine("Transaction cannot be continued as balance is " +

"Zero in account ");

throw new Exception("zero balance");

}

}

catch (Exception e) { Console.WriteLine(e.Message); }

accounholder.balance -= amount;

Console.Clear();

acholde.ShowAccountDetails(accounholder);

break;

default:

WriteLine("Enter valid choice");

break;

}

ReadLine();

} while (choice != 0);

return 0;

}

}

// question 5

public class HDFCBANK

{

public string account\_number;

public string customer\_name;

public double balance;

public HDFCBANK(string acc\_num, string cus\_name)

{

account\_number = acc\_num;

customer\_name = cus\_name;

balance = 0.00D;

}

}

public class AccountDetails

{

Account account = new Account("89674532", "Uma BAharti");

public double GetMoney()

{

double amount = 0;

Console.WriteLine("Amount");

amount = double.Parse(ReadLine());

return amount;

}

public void ShowAccountDetails(HDFCBANK ac)

{

Console.WriteLine("===============================================");

WriteLine("HDFCBANK................");

Console.WriteLine("Account holder number: {0} ", ac.account\_number);

Console.WriteLine("Account holder name : {0} ", ac.customer\_name);

Console.WriteLine("Customer has balance : {0:F} ", ac.balance);

Console.WriteLine("=============================================");

}

public static int Main()

{

byte choice = 0;

double amount = 0;

HDFCBANK accounholder = new HDFCBANK("234567", "Priya");

AccountDetailsacholde = new AccountDetails();

Console.WriteLine("Enter customer's initial deposits");

try

{

accounholder.balance = acholde.GetMoney();

}

catch (Exception e) { Console.WriteLine(e.Message); }

acholde.ShowAccountDetails(accounholder);

do

{

WriteLine("1. Show Account details");

WriteLine("2. Deposit Money");

WriteLine("3. Withdraw Money");

WriteLine("Enter Your Choice");

try

{

choice = byte.Parse(ReadLine());

}

catch (Exception e) { WriteLine(e.Message); }

Console.Clear();

switch (choice)

{

case 1:

acholde.ShowAccountDetails(accounholder);

WriteLine("click enter key for next choice");

Console.ReadKey();

break;

case 2:

WriteLine("Enter Your deposit");

amount = double.Parse(ReadLine());

accounholder.balance = accounholder.balance + amount;

Console.Clear();

acholde.ShowAccountDetails(accounholder);

break;

case 3:

Console.WriteLine("Enter Withdrawal");

amount = acholde.GetMoney();

accounholder.balance = accounholder.balance - amount;

try

{

if (accounholder.balance< 1000)

{

throw new Exception("Transaction cannot be continued " +

"belowed specified limit of Rs 1000, under balance");

}

else if (accounholder.balance == 0)

{

throw new Exception("Transaction cannot be continued " +

"belowed specified limit of Rs 1000, zero balance");

}

}

catch (Exception e) { Console.WriteLine(e.Message); }

accounholder.balance -= amount;

Console.Clear();

acholde.ShowAccountDetails(accounholder);

break;

default:

WriteLine("Enter valid choice");

break;

}

ReadLine();

} while (choice != 0);

return 0;

}

}

}