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

EmployeeDelegate. (UniCast Delegate)

1. Create a MultiCast Delegate by adding address of another method to the above delegate to list the details of Marketing Executive.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Delegates

{

//Multicaste Delegate//

public delegate void EmployeeDelegate();

public class MultiDele

{

static void Main(string[] args)

{

EmployeeDelegate emp1 = new EmployeeDelegate(KilometerTravel);

EmployeeDelegate emp2 = new EmployeeDelegate(TourAllowance);

EmployeeDelegate emp3 = new EmployeeDelegate(TelephoneAllowance);

EmployeeDelegate emp4 = new EmployeeDelegate(Details);

EmployeeDelegate emp5 = emp1 + emp2 + emp3+ emp4;

emp5();

}

public static void KilometerTravel()

{

int km = 6;

Console.WriteLine("Kilometertravel: {0}", km);

}

public static void TourAllowance()

{

int km = 6;

int ta = (int)km \* 3;

Console.WriteLine("Tour Allowance:{0}", ta);

}

public static void TelephoneAllowance()

{

int tele = 1000;

Console.WriteLine("Telephone Allowance:{0}", tele);

}

public static void Details()

{

int ID = 1;

string name = "Shruti";

int salary = 50000;

Console.WriteLine("MarketingExecutive ID:{0}", ID);

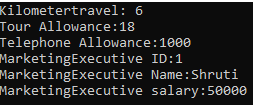
Console.WriteLine("MarketingExecutive Name:{0}", name);

Console.WriteLine("MarketingExecutive salary:{0}", salary);

}

}

}



1. Write a console Application for banking domain, having Account class with data members as account number, customer name, balance . It should have Withdraw and Deposit methods for performing banking transaction. It should also define UnderBalance, BalanceZero events. These events would be raised when balance of account is less than certain amount and equal to zero respectively.

## Now try these to get a complete grip…

1. Write separate ICICIBank class with EventHandlers for UnderBalance and

BalanceZero Events.

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

1. Write separate HDFCBank class with EventHandlers for UnderBalance and

BalanceZero Events.

* 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".

using System;

namespace ConsoleApp1

{

class Banking

{

public double AccountNumber = 659800035600;

public string Name = "Shruti Satam";

public double BankBalance = 45000;

public void data()

{

Banking Accounts = new Banking();

Console.WriteLine("Accountnumber = {0} \nAccount Holder = {1} \nBank Balance = {2}", AccountNumber, Name, BankBalance);

Console.WriteLine("-------------------------------------------");

}

public void deposit(double money)

{

if (money > 0)

{

BankBalance += money;

Console.WriteLine("Bank Balance = " + BankBalance);

Console.WriteLine("-------------------------------------------");

}

else

{

Console.WriteLine("Enter Correct Amount : ");

}

Console.WriteLine("Bank Balance = " + BankBalance);

Console.WriteLine("-------------------------------------------");

}

public void withdraw(double amount)

{

if (amount < BankBalance)

{

BankBalance -= amount;

}

else if (amount > BankBalance)

{

Console.WriteLine("Insufficient Balance");

}

else

{

Console.WriteLine("Select the Correct Option");

}

Console.WriteLine("Bank Balance = " + BankBalance);

Console.WriteLine("-------------------------------------------");

}

public void balance()

{

Console.WriteLine("Bank balance = " + BankBalance);

Console.WriteLine("-------------------------------------------");

}

}

class ICICIBank : Banking

{

public void withdraw(double amount)

{

if (amount < BankBalance)

{

BankBalance -= amount;

}

else if (amount > BankBalance)

{

Console.WriteLine("Transaction cannot be continued as balance is insufficient in Account");

}

else

{

Console.WriteLine("Select the Correct Option");

}

Console.WriteLine("Bank Balance = " + BankBalance);

Console.WriteLine("-------------------------------------------");

}

public void balance()

{

if (BankBalance == 0)

{

Console.WriteLine("Transaction cannot be continued as balance is zero in Account");

}

Console.WriteLine("Bank Balance=" + BankBalance);

Console.WriteLine("-------------------------------------------");

}

}

class HDFCBank : Banking

{

public void withdraw(double amount)

{

if (amount < BankBalance & BankBalance > 1000)

{

BankBalance -= amount;

}

else if (amount < BankBalance & BankBalance < 1000)

{

Console.WriteLine("Transaction cannot be continued below specified limit of Rs.- 1000");

}

else

{

Console.WriteLine("Select the Correct Option");

}

Console.WriteLine("Bank Balance = " + BankBalance);

Console.WriteLine("-------------------------------------------");

}

public void balance()

{

if (BankBalance == 0)

{

Console.WriteLine("Transaction cannot be continued as balance is zero in Account");

}

Console.WriteLine("Bank Balance=" + BankBalance);

Console.WriteLine("-------------------------------------------");

}

}

class Transaction

{

public static void Main()

{

try

{

again:

Console.WriteLine("Choose the Bank For furthur Transaction :- \n 1.ICICI \n 2.HDFC \n 3.Exit ");

Console.WriteLine("-------------------------------------------");

string bank = Console.ReadLine();

if (bank == "ICICI")

{

Transaction.ICICIBankTransactions();

}

else if (bank == "HDFC")

{

Transaction.HDFCBankTransactions();

}

else if (bank == "EXIT")

{

Console.WriteLine("Thank You");

}

else

{

Console.WriteLine("Select Correct Option");

goto again;

}

Console.WriteLine("-------------------------------------------");

}

catch (Exception ex)

{

Console.WriteLine(ex.GetType().Name);

}

}

public static void ICICIBankTransactions()

{

ICICIBank i = new ICICIBank();

i.data();

transaction:

Console.WriteLine("Choose the Option for transactions :- \n 1.Deposit \n 2.Withdraw \n 3.Balance \n 4.Exit");

int choice = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("-------------------------------------------");

{

if (choice == 1)

{

Console.WriteLine("Enter the Amount : ");

int amt = Convert.ToInt32(Console.ReadLine());

i.deposit(amt);

goto transaction;

}

else if (choice == 2)

{

Console.WriteLine("Enter the Amount : ");

int amt = Convert.ToInt32(Console.ReadLine());

i.withdraw(amt);

goto transaction;

}

else if (choice == 3)

{

i.balance();

goto transaction;

}

else if (choice == 4)

{

Console.WriteLine("Thank You");

}

else

{

Console.WriteLine("select correct choice");

goto transaction;

}

i.balance();

if (i.BankBalance < 1000)

{

Console.WriteLine("Underbalance Account !!!" + i.BankBalance);

}

else if (i.BankBalance == 0)

{

Console.WriteLine("Bank Balance is zero");

}

Console.WriteLine("-------------------------------------------");

}

}

public static void HDFCBankTransactions()

{

HDFCBank h = new HDFCBank();

h.data();

transaction:

Console.WriteLine("Choose the Option for transactions :- \n 1.Deposit \n 2.Withdraw \n 3.Balance \n 4.Exit");

int choice = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("-------------------------------------------");

{

if (choice == 1)

{

Console.WriteLine("Enter the Amount : ");

int amt = Convert.ToInt32(Console.ReadLine());

h.deposit(amt);

goto transaction;

}

else if (choice == 2)

{

Console.WriteLine("Enter the Amount : ");

int amt = Convert.ToInt32(Console.ReadLine());

h.withdraw(amt);

goto transaction;

}

else if (choice == 3)

{

h.balance();

goto transaction;

}

else if (choice == 4)

{

Console.WriteLine("Thank You");

}

else

{

Console.WriteLine("select correct choice");

goto transaction;

}

h.balance();

if (h.BankBalance < 1000)

{

Console.WriteLine("Underbalance Account !!!" + h.BankBalance);

}

else if (h.BankBalance == 0)

{

Console.WriteLine("Bank Balance is zero");

}

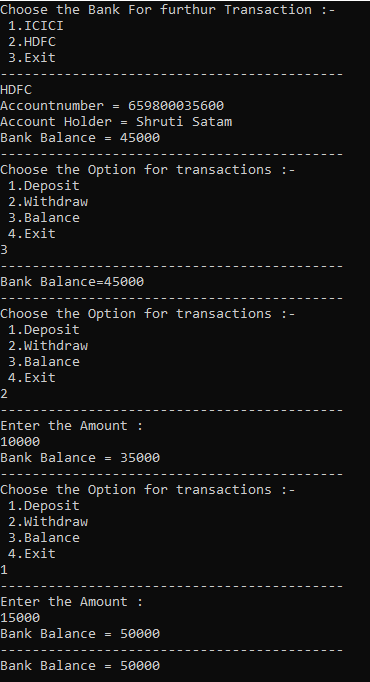
Console.WriteLine("-------------------------------------------");

}

}

}

}



## By the way

* Delegates are used as callbacks.
* Multicast delegate helps to execute a series of tasks by executing methods added to it.
* 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 the class.