using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

BankAccountSystem bankSystem = new BankAccountSystem();

while (true)

{

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

Console.WriteLine("1.Create Account");

Console.WriteLine("2.Variz");

Console.WriteLine("3.Bardasht");

Console.WriteLine("4.Account Info");

Console.WriteLine("5.Edit Info");

Console.WriteLine("6.Block Account");

Console.WriteLine("7.Exit");

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

Console.Write("choose one please : ");

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

switch (choice)

{

case 1:

bankSystem.OpenAccount();

break;

case 2:

bankSystem.Deposit();

break;

case 3:

bankSystem.Withdraw();

break;

case 4:

bankSystem.ViewAccountInfo();

break;

case 5:

bankSystem.EditAccountInfo();

break;

case 6:

bankSystem.BlockAccount();

break;

case 7:

Console.WriteLine("logging out.");

Environment.Exit(0);

break;

default:

Console.WriteLine("try again!.");

break;

}

}

}

}

class BankAccount

{

public string AccountType { get; set; }

public decimal Balance { get; set; }

public string CustomerName { get; set; }

public string NationalCode { get; set; }

public DateTime DateOfBirth { get; set; }

public string Address { get; set; }

public string FatherName { get; set; }

public string PhoneNumber { get; set; }

public void DisplayInfo()

{

Console.WriteLine($"AccountType: {AccountType}");

Console.WriteLine($"CustomerName: {CustomerName}");

Console.WriteLine($"NationalCode: {NationalCode}");

Console.WriteLine($"DateOfBirth: {DateOfBirth.ToShortDateString()}");

Console.WriteLine($"Address: {Address}");

Console.WriteLine($"FatherName: {FatherName}");

Console.WriteLine($"PhoneNumber: {PhoneNumber}");

Console.WriteLine($"Balance: {Balance} ریال");

}

public void EditInfo()

{

Console.Write("new name: ");

CustomerName = Console.ReadLine();

Console.Write("new address: ");

Address = Console.ReadLine();

Console.Write("new numphone: ");

PhoneNumber = Console.ReadLine();

}

public void BlockAccount()

{

Console.WriteLine("Blocked Successfully.");

// Implement the logic for blocking the account here

}

}

class BankAccountSystem

{

private List<BankAccount> accounts = new List<BankAccount>();

public void OpenAccount()

{

BankAccount newAccount = new BankAccount();

Console.Write("Account type (P/Q): ");

newAccount.AccountType = Console.ReadLine();

Console.Write("CusName: ");

newAccount.CustomerName = Console.ReadLine();

Console.Write("NCode: ");

newAccount.NationalCode = Console.ReadLine();

Console.Write("Birth date (YYYY/MM/DD): ");

newAccount.DateOfBirth = DateTime.Parse(Console.ReadLine());

Console.Write("Address: ");

newAccount.Address = Console.ReadLine();

Console.Write("Father's name: ");

newAccount.FatherName = Console.ReadLine();

Console.Write("Number: ");

newAccount.PhoneNumber = Console.ReadLine();

accounts.Add(newAccount);

Console.WriteLine("Account Created Successfully.");

}

public void Deposit()

{

Console.Write("Ncode: ");

string NationalCode = Console.ReadLine();

BankAccount account = FindAccountByName(NationalCode);

if (account != null)

{

Console.Write("Amount: ");

decimal amount = Convert.ToDecimal(Console.ReadLine());

account.Balance += amount;

Console.WriteLine("New balance: " + account.Balance);

Console.WriteLine("Deposit Has Done Successfully.");

}

else

{

Console.WriteLine("Doesn't Exist !!");

}

}

private BankAccount FindAccountByName(string NationalCode)

{

foreach (BankAccount account in accounts)

{

if (account.NationalCode == NationalCode)

{

return account;

}

}

return null;

}

public void Withdraw()

{

Console.Write("Ncode: ");

string NationalCode = Console.ReadLine();

BankAccount account = FindAccountByName(NationalCode);

if (account != null)

{

Console.Write("Amount: ");

decimal amount = Convert.ToDecimal(Console.ReadLine());

if (amount <= account.Balance)

{

account.Balance -= amount;

Console.WriteLine("New balance: " + account.Balance);

Console.WriteLine("Withdraw Has Done Successfully.");

}

else

{

Console.WriteLine("Insufficient Balance.");

}

}

else

{

Console.WriteLine("Doesn't Exist.");

}

}

public void ViewAccountInfo()

{

Console.Write("Ncode: ");

string NationalCode = Console.ReadLine();

BankAccount account = FindAccountByName(NationalCode);

if (account != null)

{

account.DisplayInfo();

}

else

{

Console.WriteLine("Doesn't Exist!!");

}

}

public void EditAccountInfo()

{

Console.Write("Ncode: ");

string NationalCode = Console.ReadLine();

BankAccount account = FindAccountByName(NationalCode);

if (account != null)

{

account.EditInfo();

Console.WriteLine("Edited Successfully.");

}

else

{

Console.WriteLine("Doesn't Exist.");

}

}

public void BlockAccount()

{

Console.Write("Ncode: ");

string NationalCode = Console.ReadLine();

BankAccount account = FindAccountByName(NationalCode);

if (account != null)

{

account.BlockAccount();

Console.WriteLine("Blocked Successfully.");

}

else

{

Console.WriteLine("Doesn't Exist.");

}

}

private BankAccount FindAccountByNationalCodeAndPhoneNumber(string nationalCode, string phoneNumber)

{

return accounts.Find(account => account.NationalCode == nationalCode && account.PhoneNumber == phoneNumber);

}

}

// ...