Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра ІПЗ та ВТ

Група: ВТ222

Бази даних

Лабораторна робота №1

«Технології розробки додатків .NET Core»

Виконав: Закревський С.О.

Прийняв: Чижмотря О.В.

**6**

Мета роботи: Написати програму для моделювання роботи банкомату.

Хід роботи:

1.Відкриваємо програму в консолі:

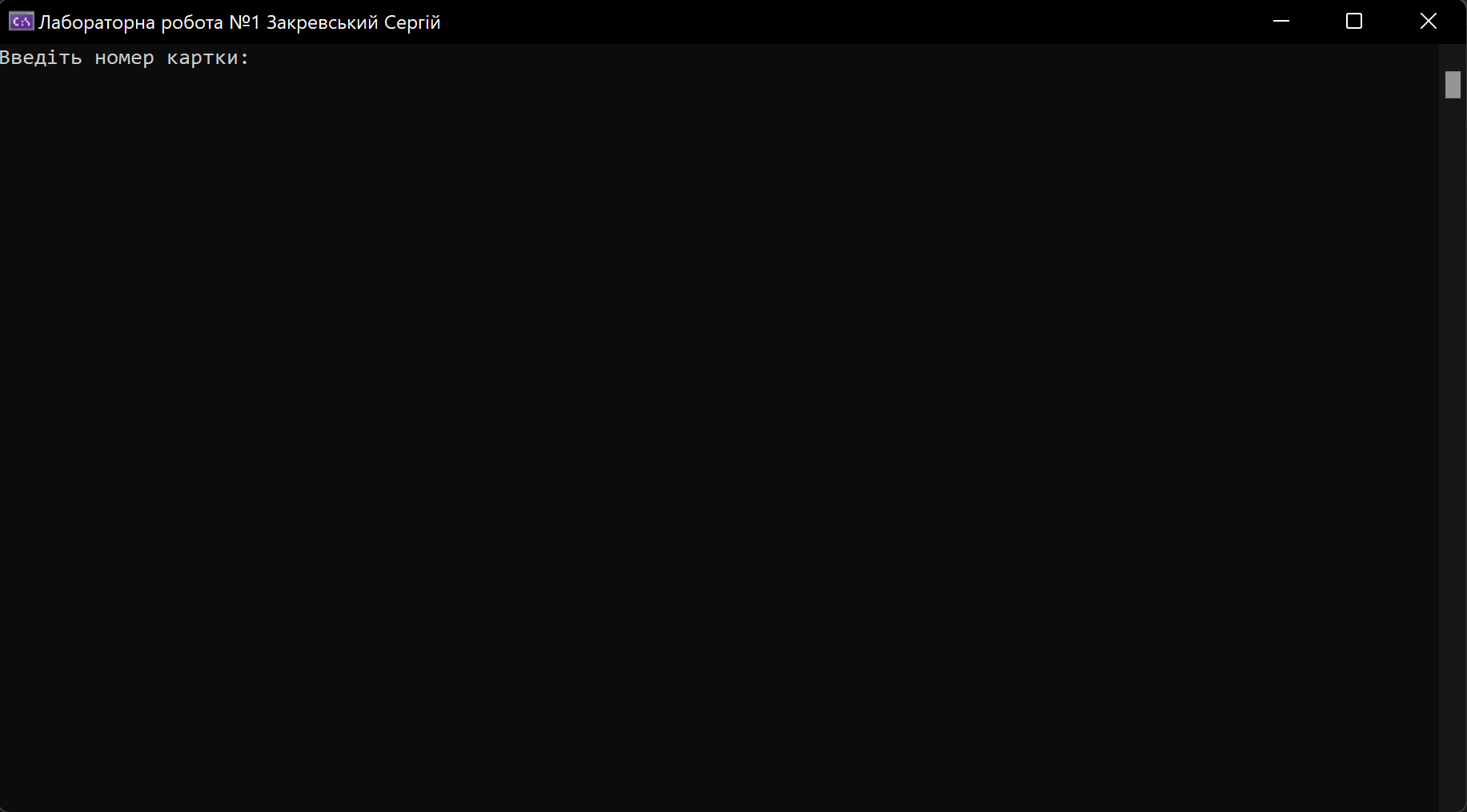


Рис 1.1 Відкриваємо програму банкомату в консолі

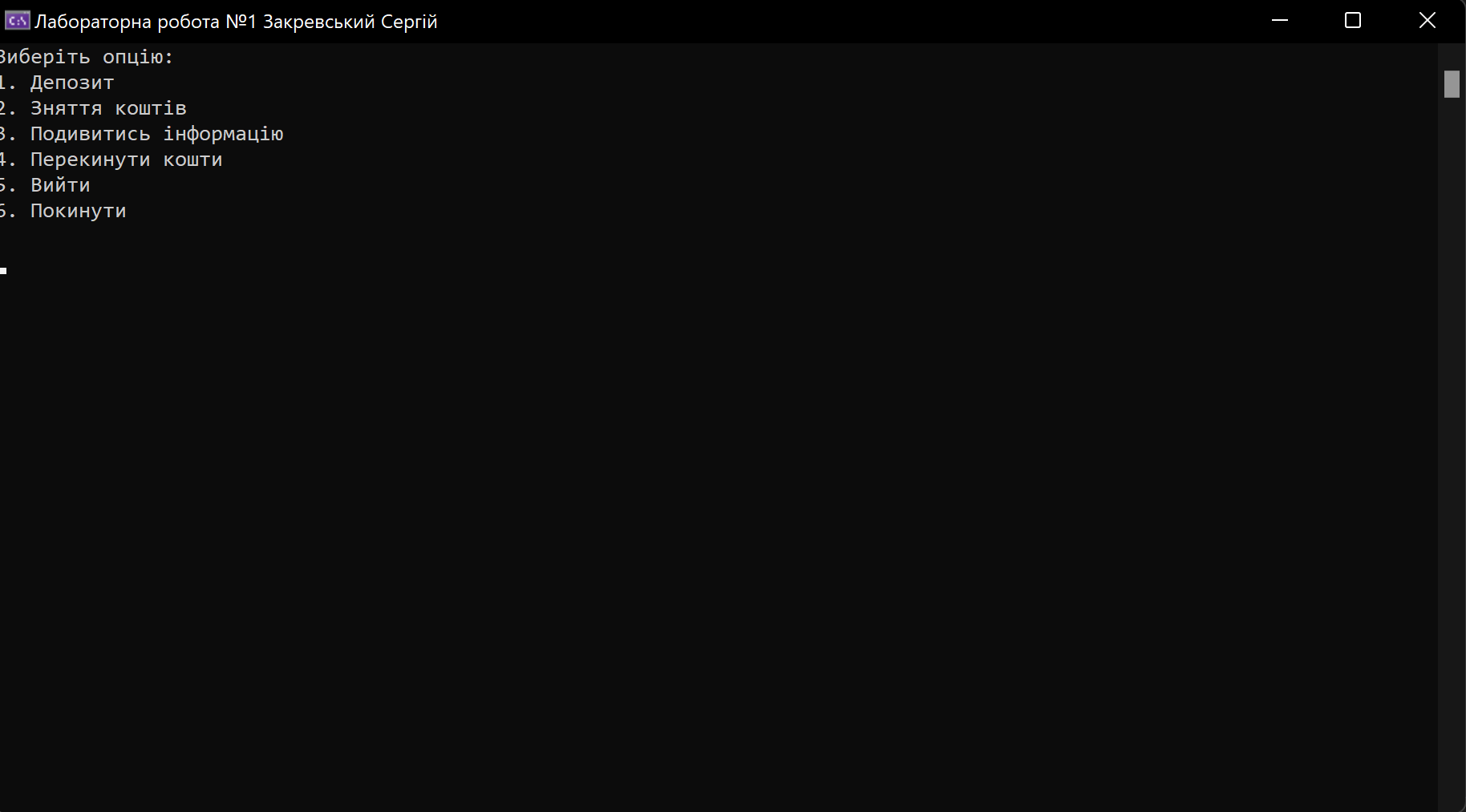


Рис 1.2 Аутентифікація успішна.Бачимо спеціальне меню

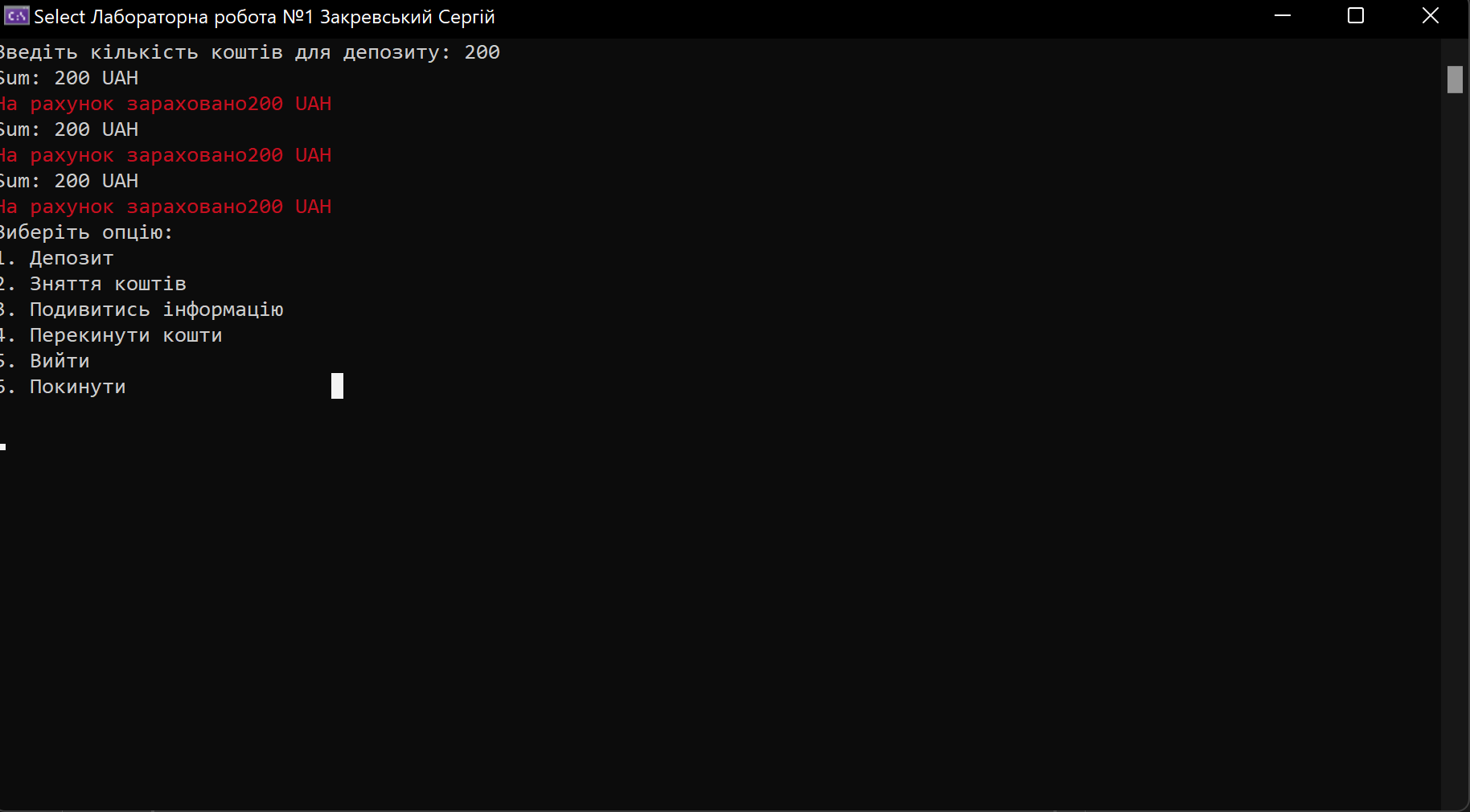


Рис 1.3 Пункт з депозитом

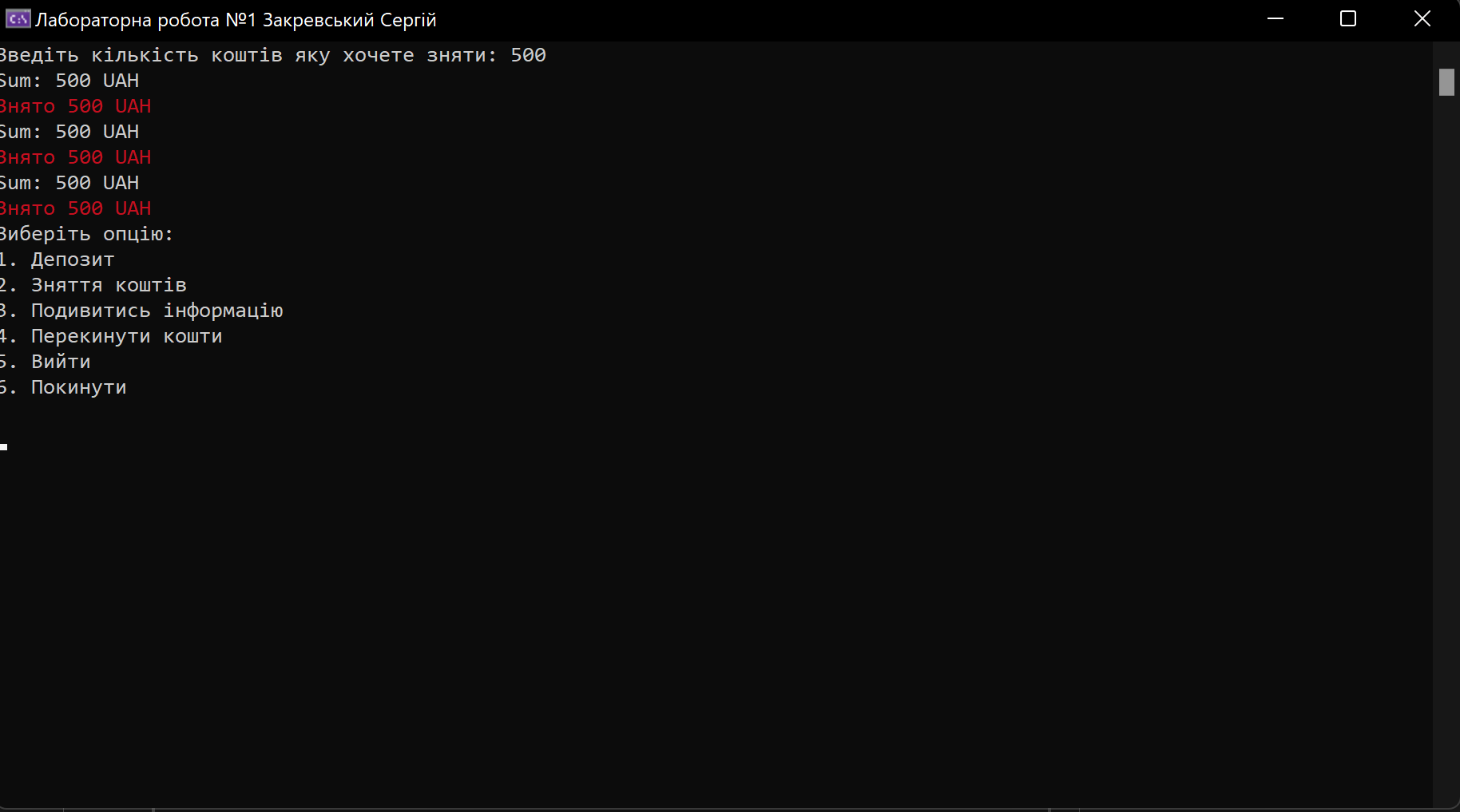


Рис 1.4 Пункт зняття коштів

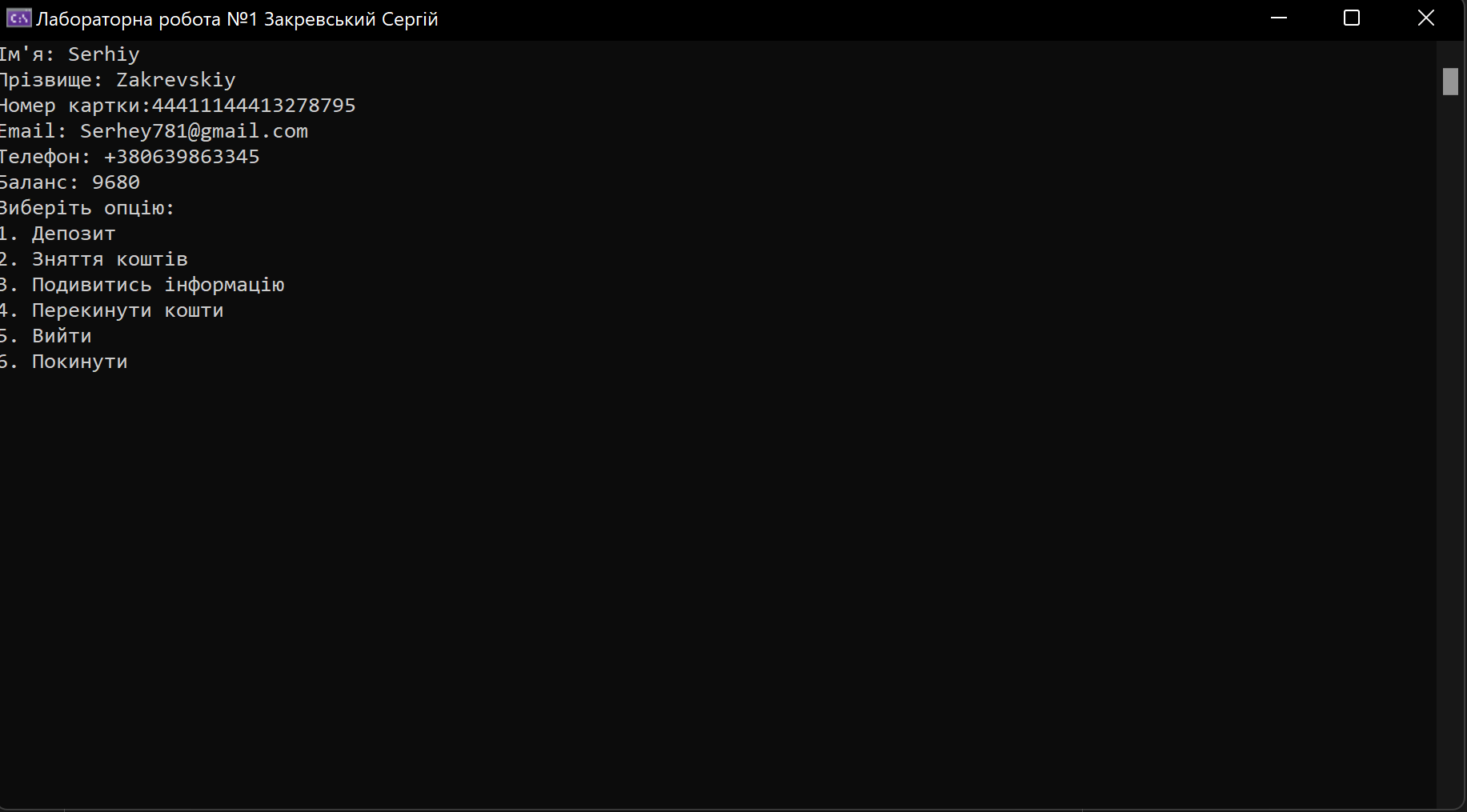


Рис 1.5 Інформація про користувача

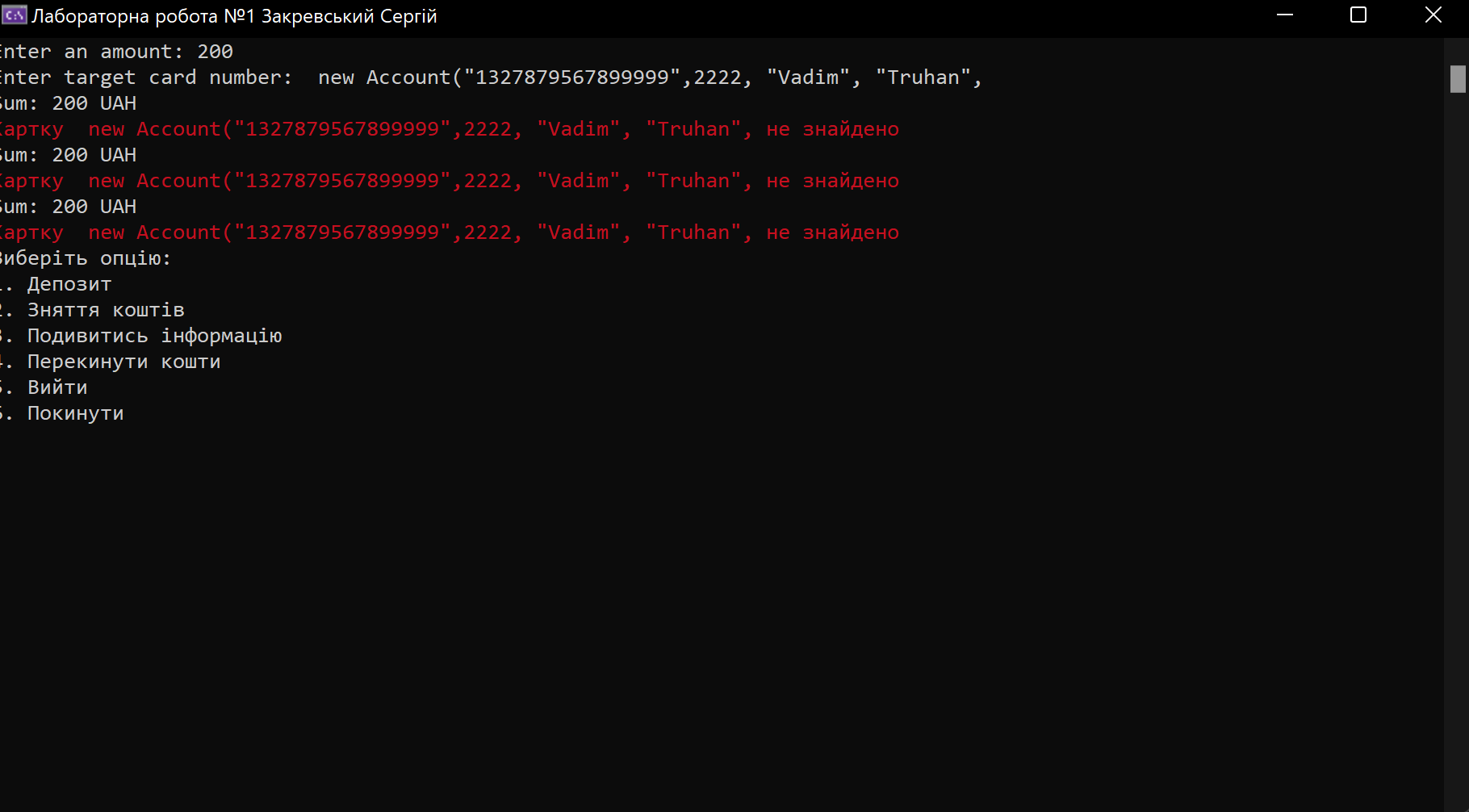


Рис 1.6 Пункт переказу на іншу карту іншого користувача

2.Відкриваємо форму:

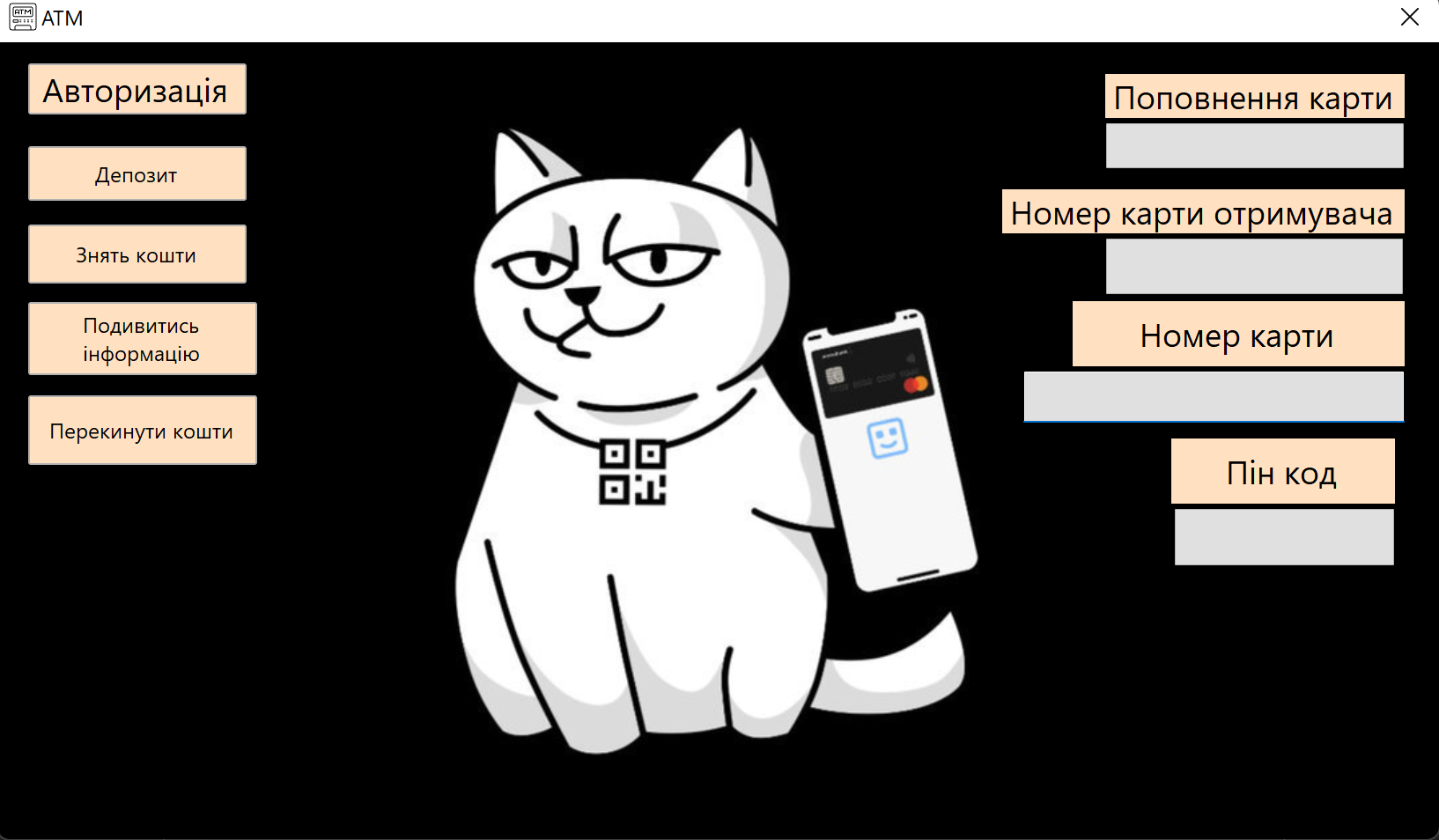


Рис 1.7 Відкрита форма

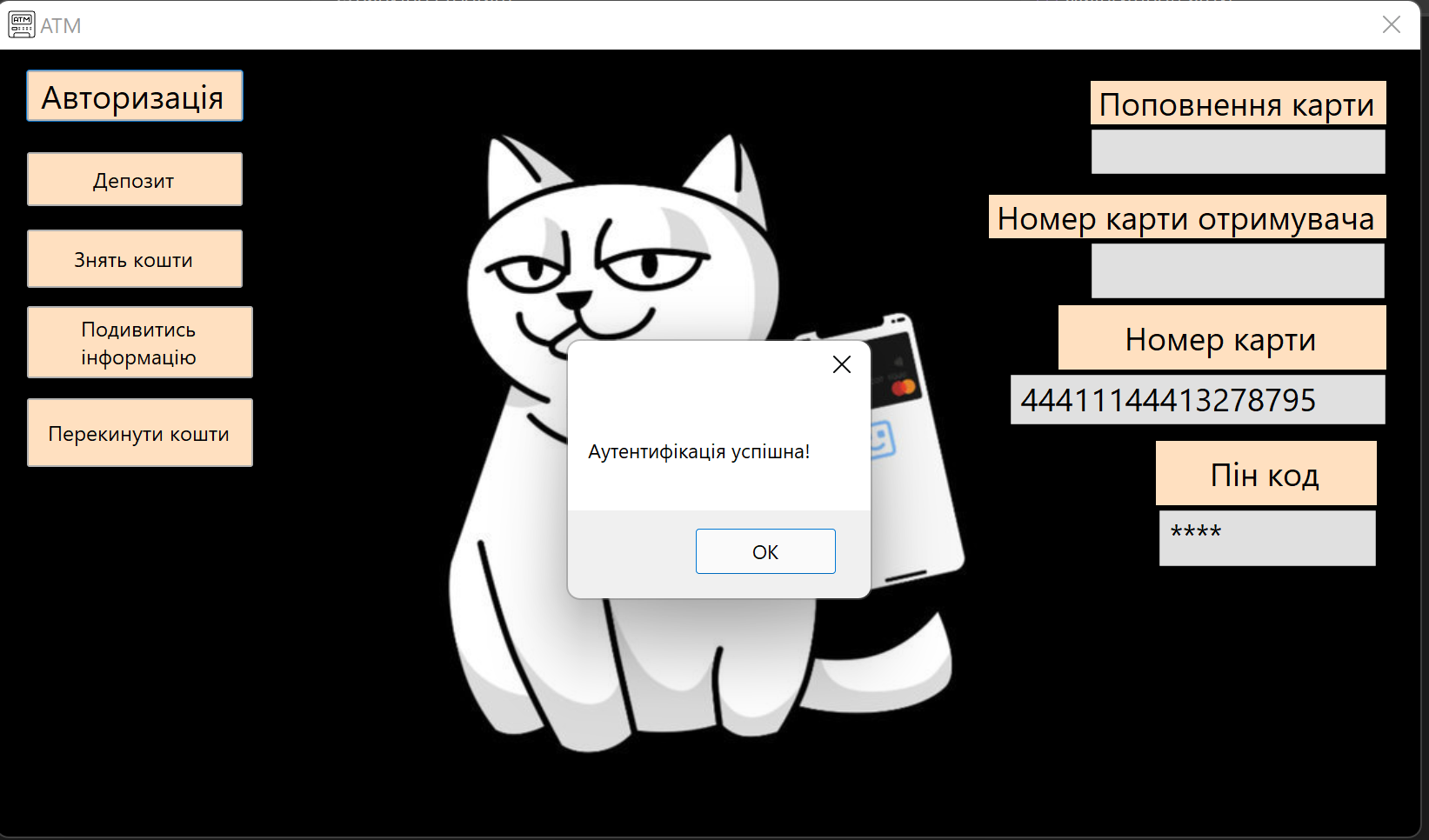


Рис 1.8 Аутентифікація успішна

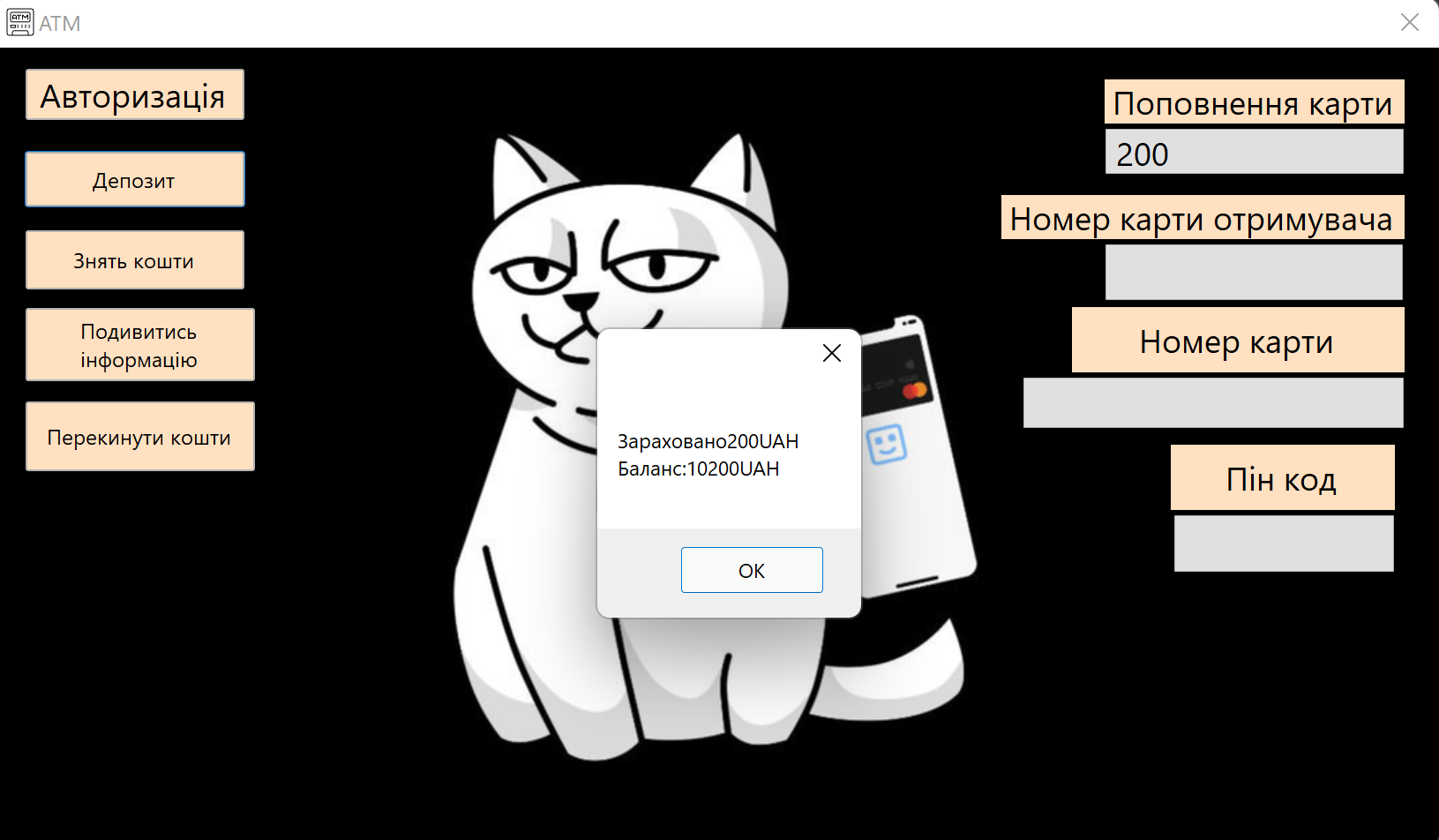


Рис 1.9 Пункт Депозит

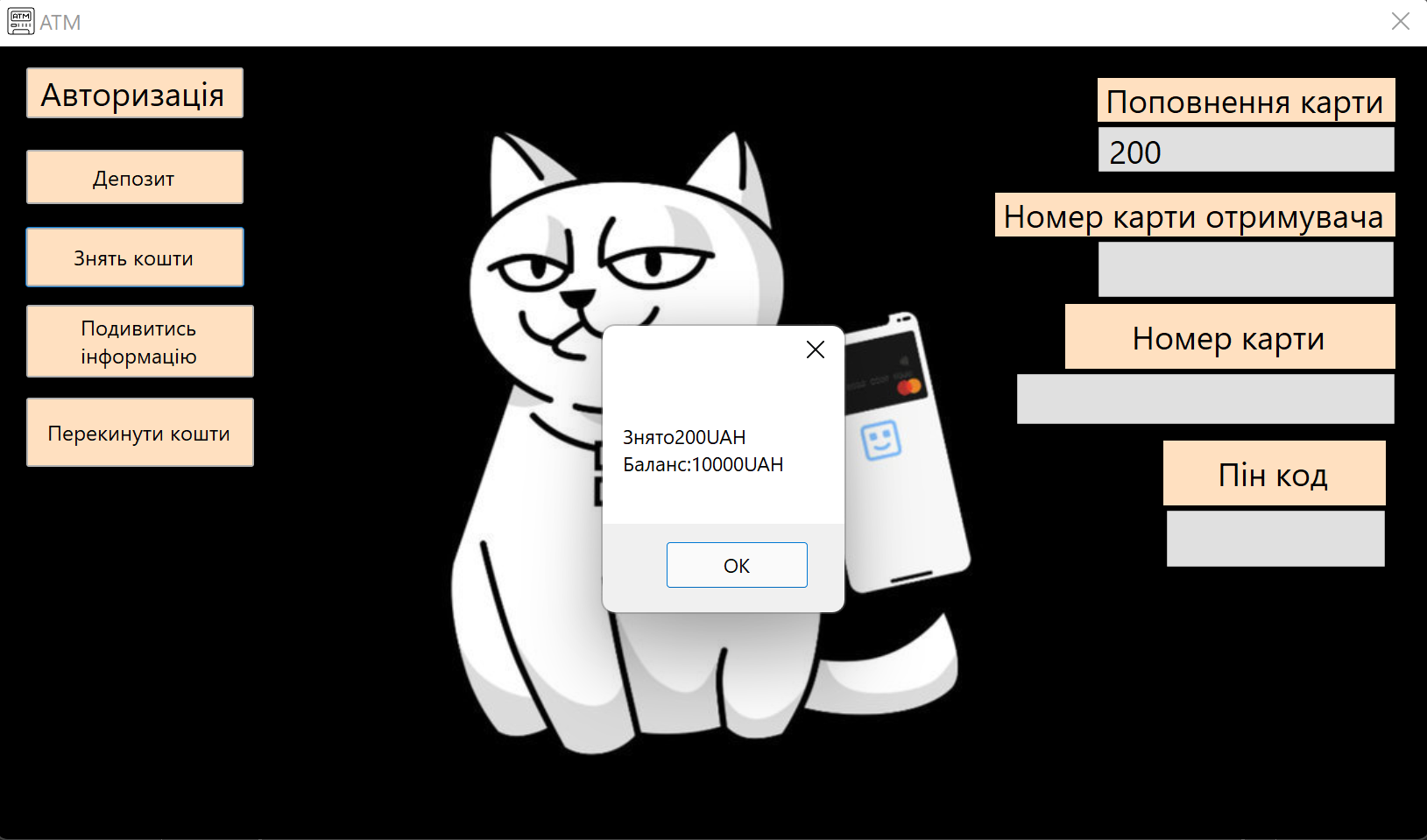


Рис 1.10 Пункт Зняття коштів

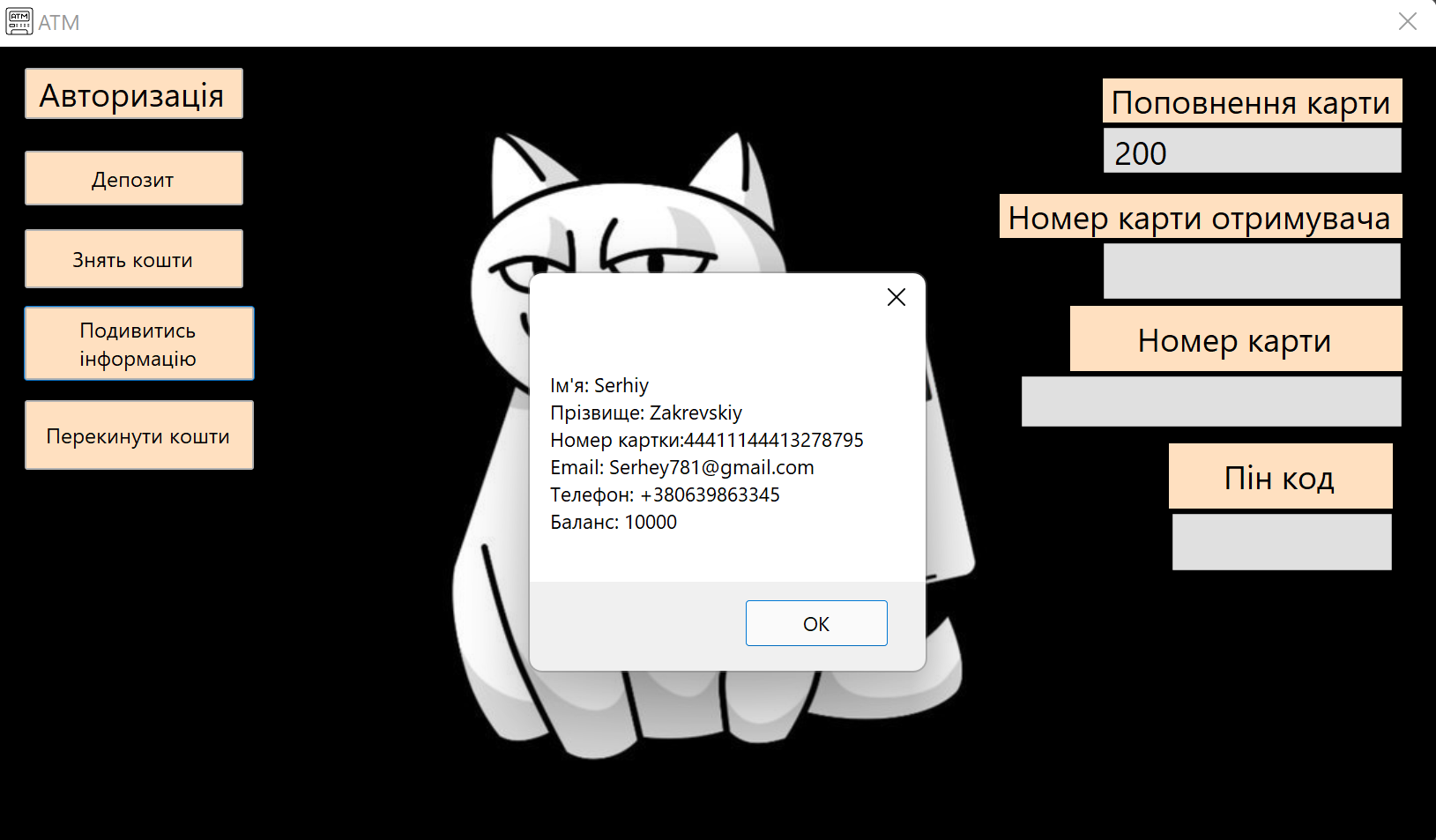


Рис 1.11 Інформація про користувача

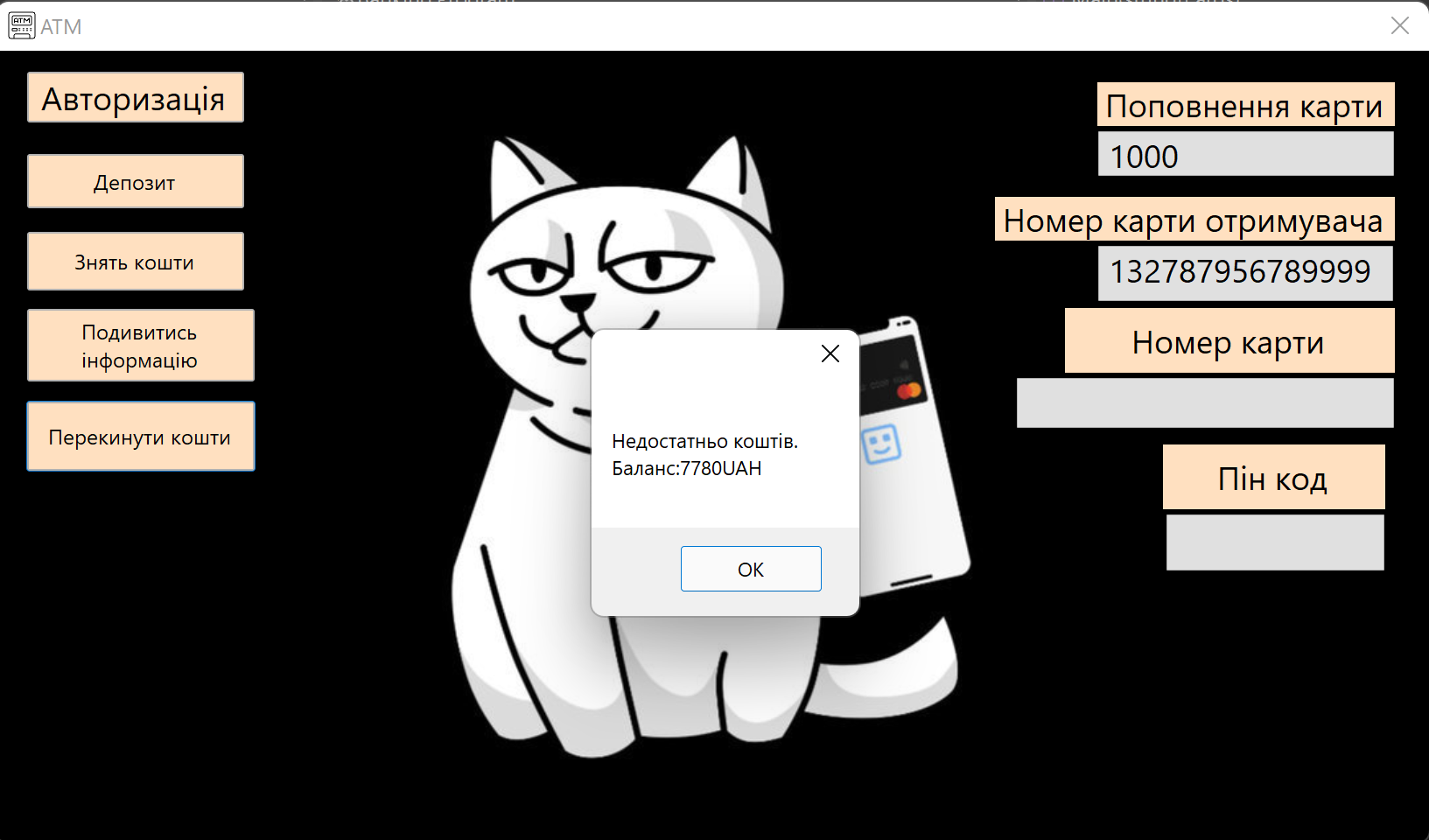


Рис 1.12 Пункт перекинути кошти

3.Вихідний код:

1) Account.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibraryBanking

{

public delegate void AccountHandler(string message);

public class Account

{

public delegate void BankOperationsHandler(Account sender,

BankOperationsEventArgs e);

public static int usercount = 0;

public string CardNumber { get; private set; }

public int CardPin { get; private set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; private set; }

public string Phone { get; private set; }

public double Balance { get; set; }

AccountHandler info;

public Account(string cardNumber, int cardPin, string firstName, string

lastName, string email, string phone, double balance)

{

CardNumber = cardNumber;

CardPin = cardPin;

FirstName = firstName;

LastName = lastName;

Email = email;

Phone = phone;

Balance = balance;

usercount++;

}

public void RegisterHandler(AccountHandler del)

{

info = del;

}

public void PrintInfo()

{

info?.Invoke($"Ім'я: {FirstName}\nПрізвище: {LastName}\nНомер картки:{ CardNumber}\nEmail: { Email}\nТелефон: { Phone}\nБаланс: { Balance}");

}

public void PrintBalance()

{

info?.Invoke($"Баланс: {Balance}");

}

}

}

Цей клас надає базову структуру для управління інформацією про банківський рахунок і дозволяє реєструвати делегати для передачі інформації або подій.

2) AutomatedTellerMachine.cs :

using ClassLibraryBanking;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibraryBanking

{

public class AutomatedTellerMachine

{

public delegate void BankOperationsHandler(AutomatedTellerMachine sender,

BankOperationsEventArgs e);

public event BankOperationsHandler Notify;

public int AtmID { get; protected set; }

public double AtmBalance { get; set; }

public string AtmAddress { get; protected set; }

public AutomatedTellerMachine(int atmID, string atmAddress, double atmBalance)

{

AtmID = atmID;

AtmAddress = atmAddress;

AtmBalance = atmBalance;

}

public void Deposit(double sum, Account[] account, int user)

{

account[user].Balance += sum;

AtmBalance += sum;

Notify?.Invoke(this, new BankOperationsEventArgs($"На рахунок зараховано{ sum } UAH", sum));

}

public void Withdraw(double sum, Account[] account, int user)

{

if (AtmBalance >= sum)

{

if (account[user].Balance >= sum)

{

{

account[user].Balance -= sum;

AtmBalance -= sum;

Notify?.Invoke(this, new BankOperationsEventArgs($"Знято {sum} UAH", sum));

}

}

else

{

Notify?.Invoke(this, new BankOperationsEventArgs("Недостатньо коштів на рахунку", sum));

}

}

else Notify?.Invoke(this, new BankOperationsEventArgs("Технічні проблеми. Будь ласка, спробуйте пізніше.", sum));

}

public bool Transfer(double sum, string cardNumber, Account[] account, int

user)

{

bool check = false;

if (account[user].Balance >= sum)

{

account[user].Balance -= sum;

if (check == false)

{

for (int j = 0; j < Account.usercount; j++)

{

if (account[j].CardNumber == cardNumber)

{

check = true;

account[j].Balance += sum;

Notify?.Invoke(this, new BankOperationsEventArgs($"{sum} UAH було перераховано на картку { cardNumber }", sum));

return true;

}

}

if (check == false) Notify?.Invoke(this, new BankOperationsEventArgs($"Картку {cardNumber} не знайдено", sum));

}

}

else

{ Notify?.Invoke(this, new BankOperationsEventArgs("Недостатньо коштів на рахунку", sum));

return false;

}

return false;

}

}

}

Цей клас реалізує базовий функціонал для операцій з банківським рахунком через банкомат, а також забезпечує сповіщення про виконані операції

3) Bank.cs :

using ClassLibraryBanking;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Mail;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibraryBanking

{

public delegate void BankHandler(string message);

public class Bank

{

public delegate void BanksHandler(Bank sender, BanksEventArgs e);

public event BankHandler Notifications;

public string BankName { get; private set; }

public string BankAddress { get; private set; }

public int ListAtms { get; set; }

public Bank(string bankName, string bankAddress, int listAtms)

{

BankName = bankName;

BankAddress = bankAddress;

ListAtms = listAtms;

}

public bool Authentication(string cardNumber, int cardPin, Account[] account,

out int user)

{

var smtpClient = new SmtpClient("smtp.gmail.com")

{

Port = 587,

UseDefaultCredentials = false,

Credentials = new NetworkCredential("vt222\_zso@student.ztu.edu.ua", "008536"),

EnableSsl = true,

DeliveryMethod = SmtpDeliveryMethod.Network

};

user = -1;

for (int i = 0; i < Account.usercount; i++)

{

if (account[i].CardNumber == cardNumber)

{

if (account[i].CardPin == cardPin)

{

user = i;

Notifications?.Invoke("Aутентифікація успішна!");

//smtpClient.Send("vt222\_zso@student.ztu.edu.ua", "serhey781@gmail.com", "Аунтефікація", "Успішно");

return true;

}

else

{

Notifications?.Invoke("Помилка введення даних! Повторіть спробу!");

//smtpClient.Send("vt222\_zso@student.ztu.edu.ua", "serhey781@gmail.com", "Аунтефікація", "Не успішна, якщо це ви підтвердіть особу");

return false;

}

}

}

return false;

}

}

}

Цей клас реалізує можливість аутентифікації користувачів банку та надсилає повідомлення через SMTP (протокол передачі пошти) у разі успішної або невдачі аутентифікації.

4) BankOperationsEventArgs.cs :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibraryBanking

{

public class BankOperationsEventArgs

{

public string Message { get; }

public double Sum { get; }

public BankOperationsEventArgs(string message, double sum)

{

Message = message;

Sum = sum;

}

}

}

Цей клас дозволяє передавати інформацію про операції в банківській системі як аргументи під час генерації подій. Наприклад, при знятті грошей або зарахуванні їх на рахунок, об'єкт BankOperationsEventArgs може бути використаний для передачі повідомлення і суми операції разом із подією, щоб інші частини програми могли відстежувати та реагувати на ці операції.

5) BanksEventArgs.cs :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibraryBanking

{

public class BanksEventArgs

{

public string Message { get; }

public BanksEventArgs(string message)

{

Message = message;

}

}

}

Цей клас дозволяє передавати інформацію про події, пов'язані з банками, як аргумент під час генерації подій. Наприклад, при реєстрації нового банку або виникненні помилки взаємодії з банками, об'єкт BanksEventArgs може бути використаний для передачі повідомлення про подію разом із подією, щоб інші частини програми могли відстежувати та реагувати на ці події.

6) Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Mail;

using System.Net;

using System.Security.Principal;

using System.Text;

using System.Threading.Tasks;

using ClassLibraryBanking;

using System.ComponentModel;

using System.Runtime.CompilerServices;

namespace Banking

{

class Program

{

public static void Main(String[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

System.Globalization.CultureInfo customCulture =

(System.Globalization.CultureInfo)

System.Threading.Thread.CurrentThread.CurrentCulture.Clone();

customCulture.NumberFormat.NumberDecimalSeparator = ".";

System.Threading.Thread.CurrentThread.CurrentCulture = customCulture;

Console.Title = "Лабораторна робота №1 Закревський Сергій";

double CheckDouble()

{

bool f;

double x;

do

{

f = double.TryParse(Console.ReadLine(), out x);

if (f == false)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Invalid value");

Console.ResetColor();

}

} while (!f);

return x;

}

int CheckInt()

{

bool f;

int x;

do

{

f = int.TryParse(Console.ReadLine(), out x);

if (f == false)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Invalid value");

Console.ResetColor();

}

} while (!f);

return x;

}

Account[] account = {

new Account("44411144413278795",1111, "Serhiy", "Zakrevskiy",

"Serhey781@gmail.com", "+380639863345", 10000.00),

new Account("1327879567899999",2222, "Vadim", "Truhan",

"TruhanVadim@gmail.com", "+380977008390", 210000.00),

};

AutomatedTellerMachine MyATM = new AutomatedTellerMachine(123456, "м.Житомир", 9999999.99);

Bank MyBank = new Bank("MyBank", "м. Житомир", 123456);

MyBank.Notifications += ShowInfo;

auth:

int user;

bool flag = false;

do

{

Console.WriteLine("Введіть номер картки: "); string cardnum = Console.ReadLine();

Console.WriteLine("Введіть пін:код: "); int pin = CheckInt();

flag = MyBank.Authentication(cardnum, pin, account, out user);

Console.ReadKey();

} while (!flag);

Console.Clear();

MyATM.Notify += DisplayMessage;

account[user].RegisterHandler(ShowInfo);

int printOption;

do

{

Console.WriteLine("Виберіть опцію:");

Console.WriteLine("1. Депозит");

Console.WriteLine("2. Зняття коштів");

Console.WriteLine("3. Подивитись інформацію");

Console.WriteLine("4. Перекинути кошти");

Console.WriteLine("5. Вийти");

Console.WriteLine("6. Покинути\n");

printOption = CheckInt();

switch (printOption)

{

case 1:

{

Console.Clear();

Console.Write("Введіть кількість коштів для депозиту: ");

double put = CheckDouble();

MyATM.Deposit(put, account, user);

break;

}

case 2:

{

Console.Clear();

Console.Write("Введіть кількість коштів яку хочете зняти: ");

double take = CheckDouble();

MyATM.Withdraw(take, account, user);

break;

}

case 3:

{

Console.Clear();

account[user].PrintInfo();

break;

}

case 4:

{

Console.Clear();

Console.Write("Enter an amount: "); double sum =

CheckDouble();

Console.Write("Enter target card number: ");

MyATM.Transfer(sum, Console.ReadLine(), account, user);

}

break;

case 5: Console.Clear(); goto auth;

case 6:Environment.Exit(0);

break;

}

} while (printOption != 0);

void DisplayMessage(AutomatedTellerMachine sender, BankOperationsEventArgs e)

{

Console.WriteLine($"Sum: {e.Sum} UAH");

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.WriteLine(e.Message);

Console.ResetColor();

}

void ShowInfo(string message)

{

Console.WriteLine(message);

}

}

}

}

Цей код моделює базовий функціонал банківської системи та дозволяє користувачу взаємодіяти з рахунками через банкомат, виконуючи різні операції.

7) WinFormsAppBanking

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>WinExe</OutputType>

<TargetFramework>net6.0-windows</TargetFramework>

<Nullable>enable</Nullable>

<UseWindowsForms>true</UseWindowsForms>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

<ProjectReference Include="..\ClassLibrary1\ClassLibraryBanking.csproj" />

</ItemGroup>

<ItemGroup>

<Compile Update="Properties\Resources.Designer.cs">

<DesignTime>True</DesignTime>

<AutoGen>True</AutoGen>

<DependentUpon>Resources.resx</DependentUpon>

</Compile>

</ItemGroup>

<ItemGroup>

<EmbeddedResource Update="Properties\Resources.resx">

<Generator>ResXFileCodeGenerator</Generator>

<LastGenOutput>Resources.Designer.cs</LastGenOutput>

</EmbeddedResource>

</ItemGroup>

</Project>

namespace WinFormsBanking

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

// To customize application configuration such as set high DPI settings or default font,

// see https://aka.ms/applicationconfiguration.

ApplicationConfiguration.Initialize();

Application.Run(new Form1());

}

}

}

Цей файл проекту описує конфігурацію та налаштування для розробки Windows-додатку

Весь вихідний код був викладений на github: <https://github.com/SerheyZakrevskiy/Lab1ATM>

Висновок:

На цій лабораторній я підтянув знання з C# написав програму для моделювання роботи банкомату.