Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра ІПЗ та ВТ

Група: ВТ23-2

Бази даних

Лабораторна робота №1

**Використання делегатів та подій у C#.**

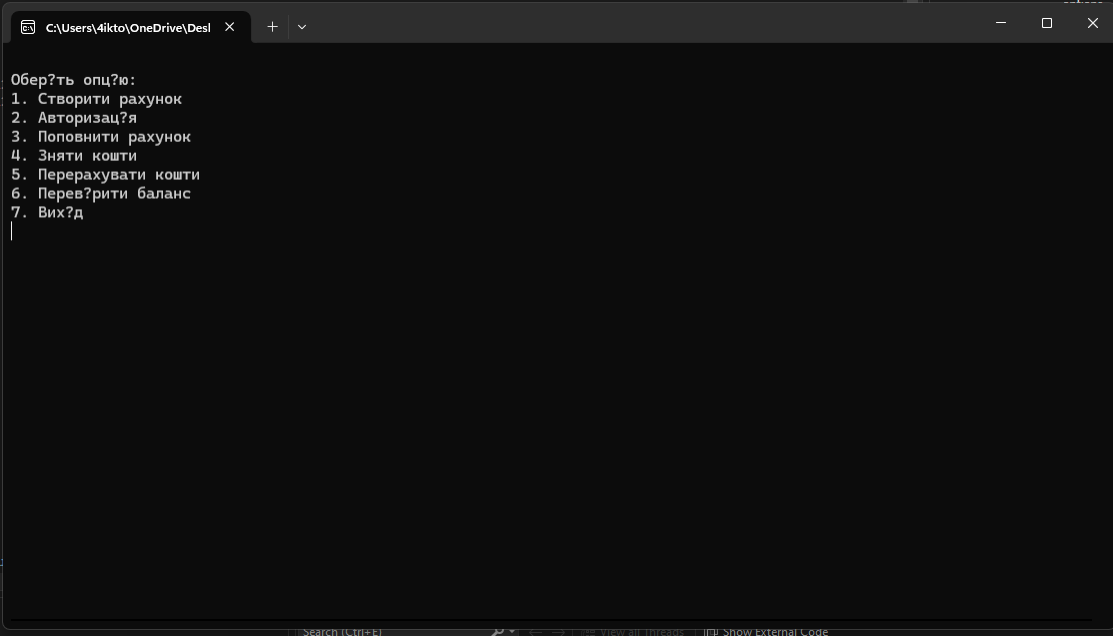
Виконав: Піхало А.М.

Прийняв: Савчук І.

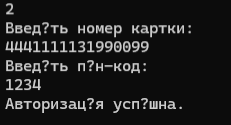
**Мета роботи:** навчитися використовувати оголошувати та використовувати делегати та події у мові програмування C#

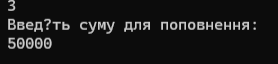
Хід роботи:

Маємо меню

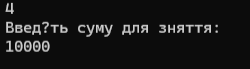


Авторизація

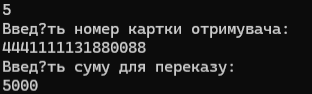


Поповнення рахунка  


Зняття коштів



Перерахування на інший рахунок



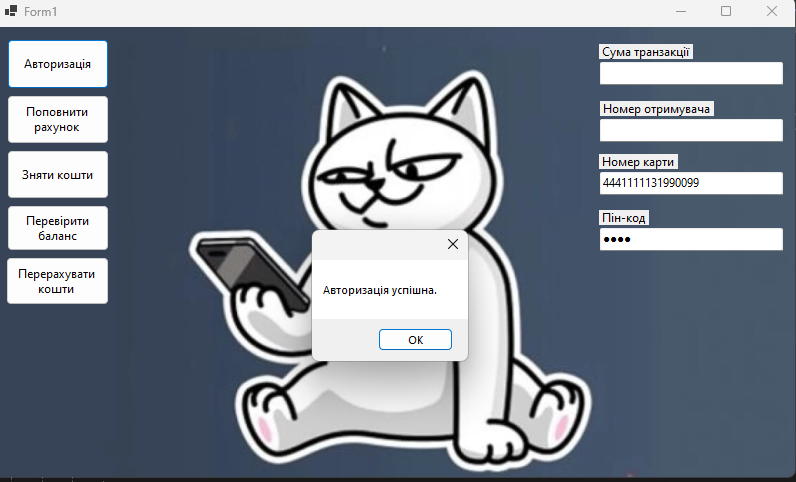
Перевірка балансу



На рахунку відразу є 5000 тому залишок 40000

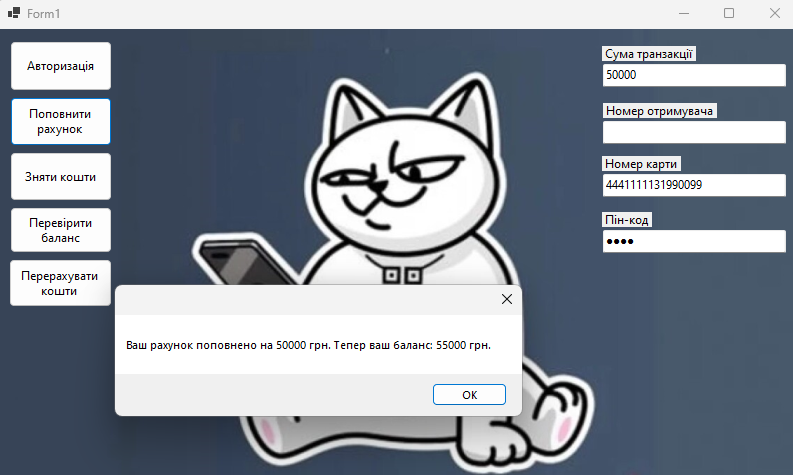
**ФОРМА**

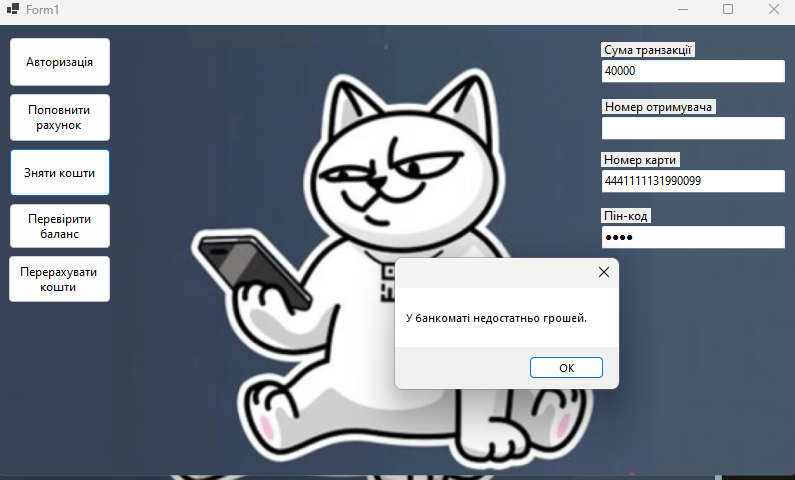
****

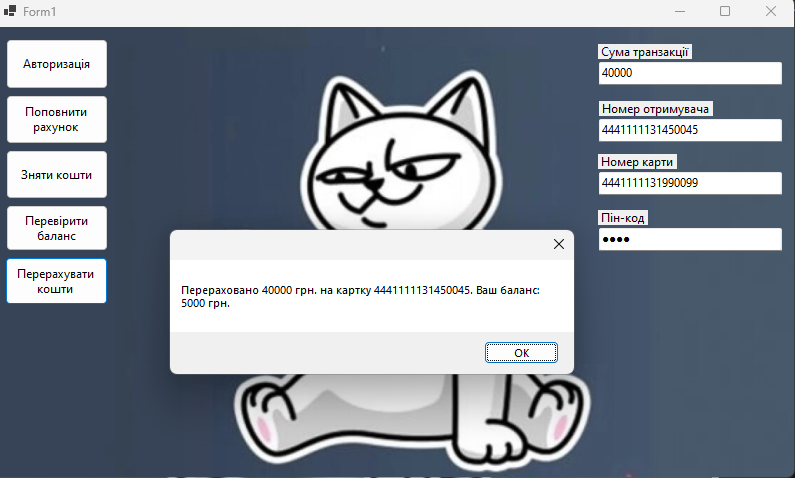
Авторизація  


Перевірка балансу   


Поповнення рахунку



Зняття коштів  


Перерахування коштів  


ConsoleAPP:

using System;

using ATMSystem;

class Program

{

static void Main(string[] args)

{

Bank bank = new Bank();

bool running = true;

Account? currentAccount = null;

while (running)

{

Console.WriteLine("\nОберіть опцію:");

Console.WriteLine("1. Створити рахунок");

Console.WriteLine("2. Авторизація");

Console.WriteLine("3. Поповнити рахунок");

Console.WriteLine("4. Зняти кошти");

Console.WriteLine("5. Перерахувати кошти");

Console.WriteLine("6. Перевірити баланс");

Console.WriteLine("7. Вихід");

string option = Console.ReadLine();

switch (option)

{

case "1":

// Створення профілю

Console.WriteLine("Введіть номер картки (16 цифр):");

string cardNumber = Console.ReadLine();

if (cardNumber.Length != 16 || !cardNumber.All(char.IsDigit))

{

Console.WriteLine("Невірний номер картки.");

break;

}

Console.WriteLine("Введіть пін-код (4 цифри):");

string pinCode = Console.ReadLine();

if (pinCode.Length != 4 || !pinCode.All(char.IsDigit))

{

Console.WriteLine("Невірний пін-код.");

break;

}

bank.CreateAccount(cardNumber, pinCode);

break;

case "2":

// Аутентифікація

Console.WriteLine("Введіть номер картки:");

cardNumber = Console.ReadLine();

Console.WriteLine("Введіть пін-код:");

pinCode = Console.ReadLine();

currentAccount = bank.FindAccount(cardNumber);

if (currentAccount != null && currentAccount.Authenticate(pinCode))

{

Console.WriteLine("Авторизація успішна.");

}

else

{

Console.WriteLine("Невірний номер картки або пін-код.");

currentAccount = null;

}

break;

case "3":

// Поповнення рахунку

if (currentAccount != null)

{

Console.WriteLine("Введіть суму для поповнення:");

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

bank.Deposit(currentAccount, amount);

}

else

{

Console.WriteLine("Будь ласка, спочатку пройдіть авторизацію.");

}

break;

case "4":

// Зняття коштів

if (currentAccount != null)

{

Console.WriteLine("Введіть суму для зняття:");

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

bank.Withdraw(currentAccount, amount);

}

else

{

Console.WriteLine("Будь ласка, спочатку пройдіть авторизацію.");

}

break;

case "5":

// Перерахування коштів

if (currentAccount != null)

{

Console.WriteLine("Введіть номер картки отримувача:");

string recipientCardNumber = Console.ReadLine();

Console.WriteLine("Введіть суму для переказу:");

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

bank.Transfer(currentAccount, recipientCardNumber, amount);

}

else

{

Console.WriteLine("Будь ласка, спочатку пройдіть авторизацію.");

}

break;

case "6":

// Перевірка балансу

if (currentAccount != null)

{

Console.WriteLine($"Ваш баланс: {currentAccount.Balance} грн.");

}

else

{

Console.WriteLine("Будь ласка, спочатку пройдіть авторизацію.");

}

break;

case "7":

running = false;

break;

default:

Console.WriteLine("Невірна опція.");

break;

}

}

}

}

Клас Bank:

namespace ATMSystem

{

public class Bank

{

public event Action<string> ShowMessage;

public Bank()

{

Account fixedAccount1 = new Account("4441111131990099", "1234");

Account fixedAccount2 = new Account("4441111131880088", "1234");

Account fixedAccount3 = new Account("4441111131450045", "1234");

AddAccount(fixedAccount1);

AddAccount(fixedAccount2);

AddAccount(fixedAccount3);

}

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

public void CreateAccount(string cardNumber, string pinCode)

{

var account = new Account(cardNumber, pinCode);

accounts.Add(account);

ShowMessage?.Invoke("Рахунок успішно створено.");

}

public void AddAccount(Account account)

{

accounts.Add(account);

}

public Account? FindAccount(string cardNumber)

{

return accounts.FirstOrDefault(acc => acc.CardNumber == cardNumber);

}

// Поповнити рахунок

public void Deposit(Account account, decimal amount)

{

if (amount > 0)

{

account.Balance += amount;

ShowMessage?.Invoke($"Ваш рахунок поповнено на {amount} грн. Тепер ваш баланс: {account.Balance} грн.");

}

else

{

ShowMessage?.Invoke("Сума поповнення повинна бути більшою за 0.");

}

}

// Зняти кошти

public void Withdraw(Account account, decimal amount)

{

if (amount <= 0)

{

ShowMessage?.Invoke("Сума зняття повинна бути більшою за 0.");

}

else if (account.Balance >= amount)

{

account.Balance -= amount;

ShowMessage?.Invoke($"Знято {amount} грн. Залишок на рахунку: {account.Balance} грн.");

}

else

{

ShowMessage?.Invoke("Недостатньо коштів на рахунку.");

}

}

// Перерахувати кошти

public void Transfer(Account sender, string recipientCardNumber, decimal amount)

{

var recipient = FindAccount(recipientCardNumber);

if (recipient == null)

{

ShowMessage?.Invoke("Рахунок отримувача не знайдено.");

return;

}

if (sender.Balance >= amount && amount > 0)

{

sender.Balance -= amount;

recipient.Balance += amount;

ShowMessage?.Invoke($"Перераховано {amount} грн. на картку {recipientCardNumber}. Ваш баланс: {sender.Balance} грн.");

}

else

{

ShowMessage?.Invoke("Недостатньо коштів або некоректна сума переказу.");

}

}

}

}

**Bank:**Цей клас забезпечує основні функції для управління банківськими рахунками, такі як створення рахунків, поповнення, зняття коштів і перекази. Він перевіряє валідність операцій і сповіщає користувача про результат через події..

Клас Account:

namespace ATMSystem

{

public class Account

{

public string CardNumber { get; set; }

public string PinCode { get; set; }

public decimal Balance { get; set; }

public Account(string cardNumber, string pinCode)

{

CardNumber = cardNumber;

PinCode = pinCode;

Balance = 5000;

}

public bool Authenticate(string pin)

{

return PinCode == pin;

}

public decimal GetBalance()

{

return Balance;

}

}

}

**Account:**Цей клас реалізує основні функції для роботи з банківським рахунком, зокрема збереження інформації про номер картки, пін-код і баланс, клас дозволяє авторизувати користувача за допомогою пін-коду, перевіряти його баланс і створювати рахунок з початковим балансом 5000 грн.

Клас AutomatedTellerMachine:

using ATMSystem;

public class AutomatedTellerMachine

{

public string ATMId { get; set; }

public string Address { get; set; }

public decimal CashAmount { get; set; }

public event Action<string>? OperationPerformed;

public AutomatedTellerMachine(string atmId, string address, decimal cashAmount)

{

ATMId = atmId;

Address = address;

CashAmount = cashAmount;

}

public void Withdraw(Account account, decimal amount)

{

if (account.Balance < amount)

{

OperationPerformed?.Invoke("Недостатньо коштів на рахунку.");

return;

}

if (CashAmount < amount)

{

OperationPerformed?.Invoke("У банкоматі недостатньо грошей.");

return;

}

account.Balance -= amount;

CashAmount -= amount;

OperationPerformed?.Invoke($"Успішно знято {amount} грн. Баланс: {account.Balance} грн.");

}

public void Deposit(Account account, decimal amount)

{

account.Balance += amount;

CashAmount += amount;

OperationPerformed?.Invoke($"Успішно зараховано {amount} грн. Баланс: {account.Balance} грн.");

}

}

**AutomatedTellerMachine:** Цей клас реалізує основні функції для виконання операцій з банкоматом, зокрема зняття та поповнення коштів. Клас зберігає інформацію про банкомат (ідентифікатор, адресу та суму в банкоматі) і дозволяє виконувати операції з рахунками, перевіряючи наявність достатньо коштів на рахунку та в банкоматі. Він сповіщає користувача про результат операцій через події.

Form.cs:

using ATMSystem;

namespace WinFormsApp1

{

public partial class Form1 : Form

{

private Bank bank;

private Account? currentAccount;

private AutomatedTellerMachine atm;

public Form1()

{

InitializeComponent();

bank = new Bank();

atm = new AutomatedTellerMachine("ATM001", "Kyiv, Shevchenka 10", 10000);

Account fixedAccount1 = new Account("4441111131990099", "1234");

Account fixedAccount2 = new Account("4441111131880088", "1234");

Account fixedAccount3 = new Account("4441111131450045", "1234");

bank.AddAccount(fixedAccount1);

bank.AddAccount(fixedAccount2);

bank.AddAccount(fixedAccount3);

bank.ShowMessage += ShowMessage;

atm.OperationPerformed += ShowMessage;

}

private void ShowMessage(string message)

{

MessageBox.Show(message);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

// Авторизація

string cardNumber = textBox3.Text;

string pinCode = textBox4.Text;

currentAccount = bank.FindAccount(cardNumber);

if (currentAccount != null && currentAccount.Authenticate(pinCode))

{

MessageBox.Show("Авторизація успішна.");

}

else

{

MessageBox.Show("Невірний номер картки або пін-код.");

currentAccount = null;

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

// Поповнення рахунку

if (currentAccount != null)

{

decimal amount;

if (decimal.TryParse(textBox1.Text, out amount) && amount > 0)

{

bank.Deposit(currentAccount, amount);

}

else

{

MessageBox.Show("Введіть правильну суму для поповнення.");

}

}

else

{

MessageBox.Show("Будь ласка, спочатку авторизуйтесь.");

}

}

private void button3\_Click\_1(object sender, EventArgs e)

{

// Зняття коштів

if (currentAccount != null)

{

if (decimal.TryParse(textBox1.Text, out decimal amount) && amount > 0)

{

atm.Withdraw(currentAccount, amount);

}

else

{

MessageBox.Show("Введіть правильну суму для зняття.");

}

}

else

{

MessageBox.Show("Будь ласка, спочатку авторизуйтесь.");

}

}

private void button4\_Click\_1(object sender, EventArgs e)

{

// Перегляд балансу

if (currentAccount != null)

{

MessageBox.Show($"Ваш баланс: {currentAccount.Balance} грн.");

}

else

{

MessageBox.Show("Будь ласка, спочатку авторизуйтесь.");

}

}

private void button5\_Click\_1(object sender, EventArgs e)

{

// Переказ коштів

if (currentAccount != null)

{

string recipientCardNumber = textBox2.Text;

decimal amount;

if (decimal.TryParse(textBox1.Text, out amount) && amount > 0)

{

bank.Transfer(currentAccount, recipientCardNumber, amount);

}

else

{

MessageBox.Show("Введіть правильну суму для переказу.");

}

}

else

{

MessageBox.Show("Будь ласка, спочатку авторизуйтесь.");

}

}

}

}

Висновок: Навчився використовувати оголошувати та використовувати делегати та події у мові програмування C#

Github- <https://github.com/entity1q/dotNET_lab_1/blob/master/~%24%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0%201.docx>

Gitlab-

<https://git.ztu.edu.ua/ipz/2023-2027/vt-23-2/pikhalo-anton/dotnet/lab_01-dotnet>