

# Лабораторна робота №1

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

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

**Виділений час:** 12 годин (4 години лабораторних робіт та 8 годин самостійної роботи).

## Завдання:

- Опрацювати теорію:
  - відеозаписи лекцій №01-02;
- Написати програму для моделювання роботи банкомату.  
Рішення має складатися з трьох проектів:
  - бібліотека класів;
  - віконний проект;
  - консольний проект.

У бібліотеці класів реалізуйте загальні типи даних, які будуть використані у віконному та консольному проекті.

У віконному додатку розробіть інтерфейс, який наближений до інтерфейсу традиційного банкомату. Консольний додаток повинен забезпечити ті ж самі можливості, що має віконний додаток, але за допомогою консольного меню.

Передбачити можливості:

- аутентифікація (перевірка введення номеру картки та пін-коду);
- перегляд балансу на картці;
- зняття коштів;
- зарахування коштів на картку;
- перерахування коштів на картку із заданим номером.

Повинна бути передбачена можливість додавання власних обробників подій на виконання операцій аутентифікації, перегляду балансу, зняття коштів та перерахування коштів на іншу картку.

Додайте власні обробники подій, які будуть виводити повідомлення:

- для віконного додатку - за допомогою виклику `MessageBox.Show("Повідомлення");`
- для консольного додатку - за допомогою виклику `Console.WriteLine("Повідомлення");`

Також можна відправляти повідомлення на електронну пошту.

Для відправки листів на електронну пошту можна використовувати код:

```
var smtpClient = new SmtplibClient("smtp.gmail.com")
{
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.13.000 – Лр.1			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лаговський А.В.			Звіт з лабораторної роботи №1		Лім.	Арк.
Перевір.		Морозов А. В.						1
Реценз.							ФІКТ, гр. ІПЗ-22-1	
Н. Контр.								
Зав.каф.								

```

        Port = 587,
        UseDefaultCredentials = false,
        Credentials = new NetworkCredential("Електронна пошта", "Пароль"),
        EnableSsl = true,
        DeliveryMethod = SmtpDeliveryMethod.Network
    };
    smtpClient.Send("Пошта відправника", "Пошта отримувача", "Тема листа", "Текст
листа");

```

Для того, щоб даний код працював потрібно в гугл-акаунті включити доступ сторонніх додатків (ця операція здійснюється за адресою: <https://myaccount.google.com/u/1/lesssecureapps>).

Якщо Ви використовуєте даний код, то при виконанні коміту видаліть власний пароль, який прописується в коді.

При виконанні завдання створіть класи, які забезпечать роботу :

- **Account** - клас, що представляє картковий рахунок клієнта (властивості: номер картки, прізвище та ім'я власника, баланс на рахунку і т.д.);
- **AutomatedTellerMachine** - клас, що моделює роботу банкомата (містить кількість грошей, які містяться у банкоматі, ідентифікатор банкомата, адресу і т.п.);
- **Bank** - клас, що представляє банк (назва банку, список банкоматів і т.п.).

Це не повний список класів, за необхідності створюйте інші класи, які Вам будуть необхідні при реалізації завдання.

3. Запустити виконану роботу у репозиторій на **GitHub** за назвою **DotNetLab1** і відкрити доступ для викладачів:

- [morozov@ztu.edu.ua](mailto:morozov@ztu.edu.ua)
- [4ov.ztu@gmail.com](mailto:4ov.ztu@gmail.com)

## Класи:

### Account:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ATMClassLibrary
{
    public class Account
    {
        private float funds;

        public void AddMoneyToAcc(float value)
        {
            funds += value;
        }

        public bool WithdrawMoney(float value)
        {
            if (value > funds)
            {
                return false;
            }
            else
            {
                funds -= value;
                return true;
            }
        }
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

        public User Authorization(string cardNumber, int PIN, User[] users)
        {
            for (int i = 0; i < users.Length; i++)
            {
                if (users[i].IsCardNumberCorrect(cardNumber) &&
users[i].IsPinCorrect(PIN))
                {
                    return users[i];
                }
            }
            return null;
        }

        public bool TransferCard(float value, string cardNumber, Account account, User[]
users, int userId)
        {
            for (int i = 0; i < users.Length; i++)
            {
                if (users[i].IsCardNumberCorrect(cardNumber) && i != userId)
                {
                    if (account.WithdrawMoney(value))
                    {
                        users[i].AddMoneyToAcc(value);
                        return true;
                    }
                    else
                    {
                        return false;
                    }
                }
            }

            return false;
        }

        public float GetAccountInfo()
        {
            return funds;
        }
    }
}

```

## AutomatedTellerMachine:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ATMClassLibrary
{
    public class AutomatedTellerMachine_
    {
        private int moneyAmount;
        private string ATMId;
        private string ATMAddress;

        public int _moneyAmount
        {
            get { return moneyAmount; }
        }

        public string _ATMId
        {
            get { return ATMId; }
        }
    }
}

```

		{				Арк.
		get { return ATMId; }				
		}				
Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	3

```

public string _ATMAddress
{
    get { return ATMAddress; }
}

public AutomatedTellerMachine_(int moneyAmount, string ATMid, string ATMAddress)
{
    this.moneyAmount = moneyAmount;
    this.ATMid = ATMid;
    this.ATMAddress = ATMAddress;
}

public void AddMoney(int value)
{
    moneyAmount += value;
}

public void WithdrawMoney(int value)
{
    moneyAmount -= value;
}
}
}

```

## Bank:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ATMClassLibrary
{
    public class Bank
    {
        private string bankName;
        private List<AutomatedTellerMachine_> _automatedTellerMachines;

        public Bank(string bankName, List<AutomatedTellerMachine_>
_automatedTellerMachines)
        {
            this.bankName = bankName;
            this._automatedTellerMachines = _automatedTellerMachines;
        }

        public string _bankName
        {
            get { return bankName; }
        }

        public List<AutomatedTellerMachine_> AutomatedTellerMachines
        {
            get { return _automatedTellerMachines; }
        }
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

## User:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ATMClassLibrary
{
    public class User : Account
    {
        private string cardNumber;
        private int PIN;
        private Account account;
        private bool auth = false;
        public User(string cardNumber, int PIN)
        {
            this.cardNumber = cardNumber;
            this.PIN = PIN;
            account = new Account();
        }

        public bool IsCardNumberCorrect(string value)
        {
            if (value == cardNumber)
            {
                auth = true;
                return true;
            }
            else
            {
                return false;
            }
        }

        public bool IsPinCorrect(int value)
        {
            if (value == PIN)
            {
                auth = true;
                return true;
            }
            else
            {
                return false;
            }
        }

        public Account GetAccount()
        {
            if (auth)
            {
                return account;
            }
            else
            {
                return null;
            }
        }

        public bool IsAuthenticated()
        {
            return auth;
        }
    }
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## Console App:

```
using ATMClassLibrary;
using System;
using System.Collections.Generic;
using System.Linq;

namespace ConsoleATM
{
    internal class Program
    {
        private static List<User> users = new List<User>
        {
            new User("1111222233334444", 5678),
            new User("9876543210123456", 4321),
            new User("5555666677778888", 8765),
            new User("1234987650123456", 9876),
            new User("9876123498765432", 3456),
        };

        private static List<AutomatedTellerMachine_> atmsList = new
List<AutomatedTellerMachine_>
        {
            new AutomatedTellerMachine_(50000, "qweiqw123", "street1"),
            new AutomatedTellerMachine_(75000, "asdasd456", "street2"),
            new AutomatedTellerMachine_(100000, "zxczxc789", "street3"),
            new AutomatedTellerMachine_(30000, "rtyrty321", "street4"),
            new AutomatedTellerMachine_(90000, "fghfgh654", "street5"),
        };

        private static Bank bank1 = new Bank("PrivatBank", atmsList);

        private delegate bool TransferMoneyDelegate(float amount, string
recipientCardNumber, Account senderAccount, User[] users, int senderIndex);
        private delegate void AddMoneyToAccountDelegate(float amount, Account
userAccount);

        private static void Main()
        {
            Console.WriteLine("Welcome to the ATM system!");

            User authorizedUser = AuthorizeUser();

            if (authorizedUser != null)
            {
                Console.ForegroundColor = ConsoleColor.Green;
                Console.WriteLine("Authorization successful!");
                Console.ResetColor();

                Account userAccount = authorizedUser.GetAccount();
                Console.WriteLine($"Current account balance:
{userAccount.GetAccountInfo()}");

                DisplayMainMenu(authorizedUser);
            }
            else
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Authorization failed. Exiting...");
                Console.ResetColor();
            }
        }

        private static User AuthorizeUser()
        {
            Console.WriteLine("Enter card number:");
            string cardNumber = Console.ReadLine();

            Console.WriteLine("Enter PIN:");
            int pin;
```

```

        if (int.TryParse(Console.ReadLine(), out pin))
        {
            foreach (var user in users)
            {
                if (user.IsCardNumberCorrect(cardNumber) && user.IsPinCorrect(pin))
                {
                    return user;
                }
            }
        }

        return null;
    }

    private static void DisplayMainMenu(User user)
    {
        while (true)
        {
            Console.ForegroundColor = ConsoleColor.Cyan;
            Console.WriteLine("\nMain Menu:");
            Console.ResetColor();
            Console.WriteLine("1. Transfer Money");
            Console.WriteLine("2. Add Money to Account");
            Console.WriteLine("3. Current Balance");
            Console.WriteLine("0. Exit");

            Console.Write("Enter your choice (1-3): ");
            string choice = Console.ReadLine();

            TransferMoneyDelegate transferMoneyDelegate = TransferMoney;
            AddMoneyToAccountDelegate addMoneyToAccountDelegate = AddMoneyToAccount;

            switch (choice)
            {
                case "1":
                    if (TransferMoneyAction(user, transferMoneyDelegate))
                    {
                        Console.ForegroundColor = ConsoleColor.Green;
                        Console.WriteLine("Transfer successful!");
                        Console.ResetColor();
                    }
                    else
                    {
                        Console.ForegroundColor = ConsoleColor.Red;
                        Console.WriteLine("Transfer failed. Insufficient funds or
invalid recipient.");
                        Console.ResetColor();
                    }
                    break;
                case "2":
                    AddMoneyToAccountAction(user, addMoneyToAccountDelegate);
                    break;
                case "3":
                    GetCurrentBalance(user);
                    break;
                case "0":
                    Console.ForegroundColor = ConsoleColor.Yellow;
                    Console.WriteLine("Exiting program. Goodbye!");
                    Console.ResetColor();
                    Environment.Exit(0);
                    break;
                default:
                    Console.ForegroundColor = ConsoleColor.Red;
                    Console.WriteLine("Invalid choice. Please enter a number from 0
to 3.");
                    Console.ResetColor();
                    break;
            }
        }
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private static bool TransferMoneyAction(User sender, TransferMoneyDelegate
transferMoneyDelegate)
{
    Console.WriteLine("Enter recipient's card number:");
    string recipientCardNumber = Console.ReadLine();

    Console.WriteLine("Enter amount to transfer:");
    float amount;
    if (float.TryParse(Console.ReadLine(), out amount))
    {
        return transferMoneyDelegate.Invoke(amount, recipientCardNumber,
sender.GetAccount(), users.ToArray(), users.IndexOf(sender));
    }

    return false;
}

private static void AddMoneyToAccountAction(User user, AddMoneyToAccountDelegate
addMoneyToAccountDelegate)
{
    Console.WriteLine("Enter amount to add to your account:");
    float amount;
    if (float.TryParse(Console.ReadLine(), out amount))
    {
        addMoneyToAccountDelegate.Invoke(amount, user.GetAccount());
    }
    else
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Invalid amount. Failed to add money to your
account.");
        Console.ResetColor();
    }
}

private static bool TransferMoney(float amount, string recipientCardNumber,
Account senderAccount, User[] users, int senderIndex)
{
    return senderAccount.TransferCard(amount, recipientCardNumber, senderAccount,
users, senderIndex);
}

private static void AddMoneyToAccount(float amount, Account userAccount)
{
    userAccount.AddMoneyToAcc(amount);

    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine($"Added {amount} to your account.");
    Console.ResetColor();
}

private static void GetCurrentBalance(User user)
{
    float balance = user.GetAccount().GetAccountInfo();
    Console.ForegroundColor = ConsoleColor.Cyan;
    Console.WriteLine($"Your Current Balance Is: {balance}");
    Console.ResetColor();
}
}
}

```



```

Welcome to the ATM system!
Enter card number:
1111222233334444
Enter PIN:
5678
Authorization successful!
Current account balance: 0

Main Menu:
1. Transfer Money
2. Add Money to Account
3. Current Balance
0. Exit
Enter your choice (1-3): 2
Enter amount to add to your account:
123
Added 123 to your account.

Main Menu:
1. Transfer Money
2. Add Money to Account
3. Current Balance
0. Exit
Enter your choice (1-3): 3
Your Current Balance Is: 123

Main Menu:
1. Transfer Money
2. Add Money to Account
3. Current Balance
0. Exit
Enter your choice (1-3): 1
Enter recipient's card number:
5555666677778888
Enter amount to transfer:
12
Transfer successful!

Main Menu:
1. Transfer Money
2. Add Money to Account
3. Current Balance
0. Exit
Enter your choice (1-3): |

```

## WPF

### MainWindow:

```

<Window x:Class="ATMwpf.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:ATMwpf"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Grid>
        <Frame Name="MainPage" NavigationUIVisibility="Hidden"/>
        <Button x:Name="authButton" Content="Log In" HorizontalAlignment="Center"
            Margin="0,257,0,0" VerticalAlignment="Top" Height="48" Width="310" Click="Button_Click"
            FontWeight="Bold" FontSize="24"/>
        <Image x:Name="logoimage" HorizontalAlignment="Center" Height="125"
            Margin="0,76,0,0" VerticalAlignment="Top" Width="554" Source="/View/PrivatBank-corporate-
            logo-latina.png"/>
        <Button x:Name="exitButton" Content="Exit" HorizontalAlignment="Center"
            Margin="0,327,0,0" VerticalAlignment="Top" Height="48" Width="174" FontSize="24"
            FontWeight="Bold" Click="exitButton_Click"/>
    </Grid>
</Window>

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace ATMwpf
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            MainPage.Navigate(new Uri("AuthorizationPage.xaml", UriKind.Relative));
            logoimage.Visibility = Visibility.Hidden;
            authButton.Visibility = Visibility.Hidden;
            exitButton.Visibility = Visibility.Hidden;
        }

        private void exitButton_Click(object sender, RoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }
    }
}

```

## AuthorizationPage:

```

<Page x:Class="ATMwpf.AuthorizationPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:ATMwpf"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="AuthorizationPage">

    <Grid>
        <Frame Name="AuthorizationPage" NavigationUIVisibility="Hidden"/>
        <TextBox x:Name="cardNumInput" HorizontalAlignment="Left" Margin="293,113,0,0"
            TextWrapping="Wrap" VerticalAlignment="Top" Width="324" Height="45" FontSize="24"
            FontWeight="Bold"/>
        <TextBox x:Name="PINinput" HorizontalAlignment="Left" Margin="293,182,0,0"
            TextWrapping="Wrap" VerticalAlignment="Top" Width="324" Height="45" FontSize="24"
            FontWeight="Bold"/>
        <Label Content="" HorizontalAlignment="Left" Margin="172,123,0,0"
            VerticalAlignment="Top"/>
        <Label x:Name="cardLabel" Content="Card:" HorizontalAlignment="Left"
            Margin="199,115,0,0" VerticalAlignment="Top" Width="71" FontWeight="Bold" FontSize="24"/>
    
```

Арк.

```

        <Label x:Name="PINlabel" Content="PIN:" HorizontalAlignment="Left"
Margin="199,182,0,0" VerticalAlignment="Top" Width="58" FontWeight="Bold" FontSize="24"/>
        <Button x:Name="tryAuthButton" Content="Log In" HorizontalAlignment="Center"
Margin="0,296,0,0" VerticalAlignment="Top" Height="48" Width="310" FontSize="24"
FontWeight="Bold" Click="tryAuthButton_Click"/>

    </Grid>
</Page>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using ATMClassLibrary;
using ATMwpf.View;

namespace ATMwpf
{
    /// <summary>
    /// Interaction logic for AutorizationPage.xaml
    /// </summary>
    ///
    public partial class AutorizationPage : Page
    {
        private User authenticatedUser;
        public AutorizationPage()
        {
            InitializeComponent();
        }
        private static List<User> users = new List<User>
        {
            new User("1111222233334444", 5678),
            new User("9876543210123456", 4321),
            new User("5555666677778888", 8765),
            new User("1234987650123456", 9876),
            new User("9876123498765432", 3456),
        };

        private static List<AutomatedTellerMachine_> atmsList = new
List<AutomatedTellerMachine_>
        {
            new AutomatedTellerMachine_(50000, "qweiqw123", "street1"),
            new AutomatedTellerMachine_(75000, "asdasd456", "street2"),
            new AutomatedTellerMachine_(100000, "zxczxc789", "street3"),
            new AutomatedTellerMachine_(30000, "rtyrty321", "street4"),
            new AutomatedTellerMachine_(90000, "fghfgh654", "street5"),
        };

        private static Bank bank1 = new Bank("PrivatBank", atmsList);

        private User AuthorizeUser()
        {
            string cardNumber = cardNumInput.Text;

            if (int.TryParse(PINinput.Text, out int pin))
            {
                foreach (var user in users)

```

Арк.

ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1

11

Змн. Арк. № докум. Підпис Дата

```

        if (user.IsCardNumberCorrect(cardNumber) && user.IsPinCorrect(pin))
        {
            return user;
        }
    }
}

return null;
}

private void tryAuthButton_Click(object sender, RoutedEventArgs e)
{
    User authorizedUser = AuthorizeUser();

    if (authorizedUser != null)
    {
        authenticatedUser = authorizedUser;
        cardNumInput.Visibility = Visibility.Hidden;
        PINinput.Visibility = Visibility.Hidden;
        tryAuthButton.Visibility = Visibility.Hidden;
        PINlabel.Visibility = Visibility.Hidden;
        cardLabel.Visibility = Visibility.Hidden;

        MainMenuPage mainMenuPage = new MainMenuPage(authorizedUser);

        NavigationService.Navigate(mainMenuPage);
    }
    else
    {
        MessageBox.Show("Authorization failed. Exiting...");
        Application.Current.Shutdown();
    }
}

public User GetAuthenticatedUser()
{
    return authenticatedUser;
}
}
}

```

## MainMenuPage:

```

<Page x:Class="ATMwpf.View.MainMenuPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:ATMwpf.View"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="MainMenuPage">

    <Grid>
        <Frame x:Name="MainContentFrame" NavigationUIVisibility="Hidden"/>
        <Button x:Name="checkBalanceButton" Content="Check Balance"
            HorizontalAlignment="Center" VerticalAlignment="Top" Height="48" Width="310"
            FontSize="24" FontWeight="Bold" Click="checkBalanceButton_Click" Margin="0,225,0,0"/>
        <Button x:Name="goBackButton" Content="Back" HorizontalAlignment="Center"
            Margin="0,330,0,0" VerticalAlignment="Top" Height="48" Width="310" FontSize="24"
            FontWeight="Bold" Click="Button_Click"/>
        <Button x:Name="addMoneyButton" Content="Add Funds" HorizontalAlignment="Center"
            Margin="0,84,0,0" VerticalAlignment="Top" Height="48" Width="310" FontSize="24"
            FontWeight="Bold" Click="addMoneyButton_Click"/>
        <Button x:Name="transferMoneyButton" Content="Transfer Funds"
            HorizontalAlignment="Center" Margin="0,154,0,0" VerticalAlignment="Top" Height="48"
            Width="310" FontSize="24" FontWeight="Bold" Click="transferMoneyButton_Click"/>
    
```

Арк.

12

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА 23.12.16.000 Лр.1
------	------	----------	--------	------	---

```
</Grid>
</Page>
```

```
using ATMClassLibrary;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
```

```
namespace ATMwpf.View
{
```

```
    public partial class MainMenuPage : Page
    {
```

```
        private User currentUser;
        public MainMenuPage(User user)
        {
            InitializeComponent();
            currentUser = user;
        }
```

```
        private void addMoneyButton_Click(object sender, RoutedEventArgs e)
        {
            if (currentUser != null && currentUser.IsAuthenticated())
            {
                NavigationService.Navigate(new addFundsPage(currentUser));
                addMoneyButton.Visibility = Visibility.Hidden;
                transferMoneyButton.Visibility = Visibility.Hidden;
                checkBalanceButton.Visibility = Visibility.Hidden;
                goBackButton.Visibility = Visibility.Hidden;
            }
            else
            {
                MessageBox.Show("User not authenticated.");
            }
        }
```

```
        private void MainContentFrame_Navigated(object sender, NavigationEventArgs e)
        {
        }
```

```
        private void transferMoneyButton_Click(object sender, RoutedEventArgs e)
        {
            this.NavigationService.Navigate(new TransferMoneyPage(currentUser));
        }
```

```
        private void checkBalanceButton_Click(object sender, RoutedEventArgs e)
        {
            if (currentUser != null && currentUser.IsAuthenticated())
            {
                MessageBox.Show($"Your current balance is:
{currentUser.GetAccount().GetAccountInfo():C}");
            }
            else
            {
                MessageBox.Show("User not authenticated.");
            }
        }
```

						Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
        this.NavigationService.Navigate(new AuthorizationPage());
    }
}

```

## TransferMoneyPage:

```

<Page x:Class="ATMwpf.TransferMoneyPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:ATMwpf"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="TransferMoneyPage">

    <Grid>
        <Frame x:Name="TransferMoney" NavigationUIVisibility="Hidden"
        Navigated="TransferMoney_Navigated"/>
        <TextBox x:Name="amountToTransfer" HorizontalAlignment="Center"
        TextWrapping="Wrap" VerticalAlignment="Top" Width="320" Height="48" FontSize="24"
        FontWeight="Bold" Margin="0,225,0,0"/>
        <Label Content="Amount To Transfer:" HorizontalAlignment="Center"
        VerticalAlignment="Top" FontSize="24" FontWeight="Bold" Margin="0,162,0,0"/>
        <Label Content="Card Number:" HorizontalAlignment="Center" Margin="0,36,0,0"
        VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>
        <TextBox x:Name="cardNumberInput" HorizontalAlignment="Center"
        TextWrapping="Wrap" VerticalAlignment="Top" Width="320" Height="48" FontSize="24"
        FontWeight="Bold" Margin="0,100,0,0"/>
        <Button x:Name="transferBttn" Content="Transfer" HorizontalAlignment="Center"
        Margin="0,300,0,0" VerticalAlignment="Top" Height="48" Width="156" FontWeight="Bold"
        FontSize="24" Click="transferBttn_Click"/>
        <Button x:Name="cancelBttn" Content="Cancel" HorizontalAlignment="Center"
        Margin="0,365,0,0" VerticalAlignment="Top" Height="48" Width="110" FontWeight="Bold"
        FontSize="24" Click="cancelBttn_Click"/>
    </Grid>
</Page>

using ATMClassLibrary;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace ATMwpf
{
    /// <summary>
    /// Interaction logic for TransferMoneyPage.xaml
    /// </summary>
    public partial class TransferMoneyPage : Page
    {
        private User currentUser;

```

					ЖИТОМІРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    InitializeComponent();
    currentUser = user;
}

private void TransferMoney_Navigated(object sender, NavigationEventArgs e)
{
}

private void cancelBtn_Click(object sender, RoutedEventArgs e)
{
    this.NavigationService.GoBack();
}

private void transferBtn_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(amountToTransfer.Text) ||
string.IsNullOrEmpty(cardNumberInput.Text))
    {
        MessageBox.Show("Please enter both amount and card number.");
        return;
    }

    if (!float.TryParse(amountToTransfer.Text, out float amount))
    {
        MessageBox.Show("Invalid amount. Please enter a valid numeric value.");
        return;
    }

    currentUser.GetAccount().WithdrawMoney(amount);
    MessageBox.Show($"Successfully deducted {amount:C} from your account. Your
updated balance is {currentUser.GetAccount().GetAccountInfo():C}.");
}

}
}

```

### AddFundsPage:

```

<Page x:Class="ATMwpf.View.addFundsPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:ATMwpf.View"
mc:Ignorable="d"
d:DesignHeight="450" d:DesignWidth="800"
Title="addFundsPage">

    <Grid>
        <TextBox x:Name="amountTextBox" HorizontalAlignment="Center" Margin="0,145,0,0"
TextWrapping="Wrap" VerticalAlignment="Top" Width="364" FontSize="24" FontWeight="Bold"/>
        <Label Content="Amount of money to be added:" HorizontalAlignment="Center"
Margin="0,65,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>
        <Button x:Name="addMoneyButton" Content="Add Money" HorizontalAlignment="Center"
Margin="0,215,0,0" VerticalAlignment="Top" Height="45" Width="210" FontSize="24"
FontWeight="Bold" Click="addMoneyButton_Click"/>
        <Button Content="Back" HorizontalAlignment="Center" Margin="0,300,0,0"
VerticalAlignment="Top" Height="45" Width="160" Click="Button_Click" FontSize="24"
FontWeight="Bold"/>
    </Grid>
</Page>

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

```

using ATMClassLibrary;
using System.Windows;
using System.Windows.Controls;

namespace ATMwpf.View
{
    public partial class addFundsPage : Page
    {
        private User currentUser;

        public addFundsPage(User user)
        {
            InitializeComponent();
            currentUser = user;

            if (currentUser == null || !currentUser.IsAuthenticated())
            {
                MessageBox.Show("User not authenticated. Exiting...");
                Application.Current.Shutdown();
            }

            private void Button_Click(object sender, RoutedEventArgs e)
            {
                this.NavigationService.Navigate(new MainMenuPage(currentUser));
            }

            private void addMoneyButton_Click(object sender, RoutedEventArgs e)
            {
                if (currentUser != null && currentUser.IsAuthenticated())
                {
                    if (float.TryParse(amountTextBox.Text, out float amount))
                    {
                        AddMoneyToUserBalance(amount);
                        MessageBox.Show($"Successfully added {amount:C} to your account.");
                    }
                    else
                    {
                        MessageBox.Show("Invalid amount. Please enter a valid numeric
value.");
                    }
                }
                else
                {
                    MessageBox.Show("User not authenticated. Exiting...");
                    Application.Current.Shutdown();
                }
            }

            private void AddMoneyToUserBalance(float amount)
            {
                currentUser.GetAccount().AddMoneyToAcc(amount);
            }
        }
    }
}

```



Результат виконання додатку:



Log In

Exit

Card: 1111222233334444

PIN: 5678

Log In


					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Amount of money to be added:

12312


Add Money

Back



Successfully added \$12,312.00 to your account.

OK



Your current balance is: \$12,312.00

OK

Add Funds

Transfer Funds

Check Balance

Back

Card Number:

1321233121232222

Successfully deducted \$100.00 from your account. Your updated balance is \$12,212.00.

OK

Transfer

Cancel

**Висновок:** під час виконання даної лабораторної роботи я навчився оголошувати та використовувати делегати та події у мові програмування C#.

**GitHub:** <https://github.com/AdamLahovskyi/DotNetLab1>