

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра виробництва приладів

(повна назва кафедри)

КУРСОВА РОБОТА

з технології розроблення програмного забезпечення

(назва кредитного модуля)

на тему: "Автоматизована система клієнта банку"

Студентів 2 курсу групи ПБ-01
за освітньою програмою " Комп'ютерно-інтегровані системи
та технології в приладобудуванні"
спеціальності 151 " Автоматизація та комп'ютерно-
інтегровані технології "

Нижник В.А.

(прізвище та ініціали)

Статкевич О.Б.

(прізвище та ініціали)

Мітцих О.М.

(прізвище та ініціали)

Анісімов Д.Д.

(прізвище та ініціали)

Керівник доцент, к.т.н. Барандич К.С.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Залікова оцінка _____

Кількість балів: _____

Члени комісії

_____ (підпис)

_____ (вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

_____ (вчене звання, науковий ступінь, прізвище та ініціали)

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра виробництва приладів

Дисципліна (кредитний модуль) Курсова робота з технології розроблення програмного забезпечення

Викладач доцент, к.т.н. Барандич К.С.

Курс 2 Група ПБ-01 Семестр 3

ЗАВДАННЯ

на курсову роботу студентів

Бригади №4 “4 Мушкетера inc.”

на тему

«Автономна інформаційна система клієнта банку»

1. Строк здачі студентом виконаної роботи (проекту) 24.12.2021 р.

2. Вихідні дані до роботи (проекту): опорний конспект лекцій

3. Зміст пояснювальної записки (відповідно до пунктів РСО) _____

Титульний аркуш. Бланк завдання та календарний план. Вступ. 1. Технічне завдання.
2. Архітектура проекту. 3. Програмний код. 4. Результати розробки. 5. Програмна документація. 6. Керівництво користувача. 7. Результати тестування. Висновки Література. Додатки.

4. Перелік графічного матеріалу з точним зазначенням обов'язкових креслень та їх формату _____

5. Дата видачі завдання на курсову роботу 03.09.2021

КАЛЕНДАРНИЙ ПЛАН

(відповідно до РСО)

№ п/п	Назва пунктів курсової роботи (проекту)	Строк виконання роботи (проекту)	Бали, нараховані за виконання календарного плану	Підпис викладача
1	Формування вимог до програмного забезпечення	20.09.2021р.		
2	Створення технічного завдання на розробку	27.09.2021р.		
3	Архітектурне проектування з забезпеченням нефункціональних вимог (безпека, надійність, розширюваність, тощо)	11.10.2021р.		
4	Документування архітектури	18.10.2021р.		
5	Програмування основного функціоналу (графічний інтерфейс користувача, бізнес-логіка, тощо)	15.11.2021р.		
6	Оформлення програмної документації	22.11.2021р.		
7	Тестування	29.11.2021р.		
8	Виправлення багів	06.12.2021р.		
9	Написання детального керівництва користувача	13.12.2021р.		
10	Оформлення пояснювальної записки до курсової роботи	20.12.2021р.		

Керівник роботи доцент, к.т.н. Барандич К.С _____ підпис

Студенти Нижник В.А., Статкевич О.Б., Мітцих О.М., Анісімов Д.Д. _____ підпис

Кількість балів за виконання роботи _____

Кількість балів за дотримання етапів календарного плану _____

Сумарна кількість балів _____ Сумарна оцінка _____

Керівник роботи доцент, к.т.н. Барандич К.С. _____ підпис

Зміст

<u>Технічне завдання</u>	2
<u>Архітектура</u>	5
<u>Програмний код</u>	7
<u>Результати розробки ПЗ</u>	45
<u>Програмна документація</u>	53
<u>Керівництво користувача</u>	58
<u>Результати тестувань</u>	66
<u>Висновок</u>	66

Технічне завдання

Вступ

Дане технічне завдання поширюється на розробку автоматизованої системи банку, яка призначена для введення і обробки даних в межах зазначених завдань, а також для їх вирішення, зберігання отриманих результатів та використання розробниками програмних і апаратних засобів обчислювальної техніки.

Підстава для розробки

Система розробляється на підставі наказу викладача дисципліни “Технології розробки програмного забезпечення” Катерини Барандич від 01.09.2021 року і відповідно до навчального плану на 2021-2022 навчальний рік.

Призначення розробки

Автоматизована інформаційна система клієнта банку призначена для покращення оперативності і якості керування фінансами клієнта.

Основним завданням є надання клієнту можливості дистанційного керування фінансами .

Вимоги до програмного продукту

Коректно взаємодіяти із данною базою даних

Використання мови програмування “c++”(можливі зміни у данному пункті)

Вимоги до функціональних характеристик.

Система повинна представляти сукупність методичних і програмних засобів вирішення наступних завдань:

- Забезпечити можливість клієнтів : поповнення коштів, перевід коштів з карти на карту ,оплати соцпослуг ,мобільних номерів , кредит (до 50000грн), виплати в місяць(до 200000), депозит.
- Історія використання коштів карти
- Авторизація та безпека

Для цих завдань повинні бути реалізовані:

- інструменти керування для користувачів
- механізм оплати
- алгоритм запису і зберігання інформації що до використання карти клієнта

- механізм входу в систему
- механізм перевірки коректності введених даних(логін – унікальність, пароль >10 символів , латинські літери та усі спе символи окрім “/”)

Не функціональні характеристики:

Також на стартовій сторінці продукту повинні бути:

- відомості про картки
- робочі кнопки для оплати соцпослуг ,поповнення мобільного(можливе доповнення списку)

Усі інші послуги розміщені у інших вікнах або спливаючих вікнах

Вимоги до надійності. Програмний продукт повинен відповідати сучасному рівню вимог, які висуваються до надійності програмного забезпечення:

- забезпечувати коректне зчитування і запис даних
- передбачати контроль уведеної інформації і блокування некоректних дій користувача при роботі з системою
- забезпечувати цілісність інформації, що зберігається в базі даних.

Вимоги до параметрів і складу технічних засобів. Мінімальні системні вимоги для роботи програмного продукту мають бути наступними:

достатня кількість роз'ємів для підключення необхідного обладнання, тактова частота процесора – 1 ГГц; обсяг вільного дискового простору 100 ГБ; обсяг оперативної пам'яті 4 ГБ; роздільна здатність сенсорного монітора 1080x1920; відеоадаптер з підтримкою DirectX 9 і достатньою кількістю роз'ємів для підключення моніторів.

Вимоги до інформаційної та програмної сумісності.

Програма повинна працювати в операційних системах Windows 7/7.1/10.

Вимоги до програмної документації

Розроблювана система повинна включати довідкову інформацію про свою роботу, а також механізм підказок користувачеві. До складу супроводжуючої документації повинні входити: розрахунково-пояснювальна записка, яка містить опис системи; під керівництвом програміста.

Етапи розробки

Розробка програмного продукту відбудуватиметься в межах ітераційно-інкрементної моделі життєвого циклу. Кожен інкремент включатиме перелік робіт за етапами 3-6 (табл. 1). Час тривалості одного інкременту – 2 тижні, однієї ітерації – 3 дні.

Таблиця 1 – Основні етапи життєвого циклу

Номер етапу	Назва етапу	Срок виконання	Звітність
1	Аналіз вимог до системи	16.09.2021 – 30.09.2021	Технічне завдання на розробку програмного продукту.
2	Проектування системної архітектури	01.10.2021 – 20.10.2021	Блок-схема системи. Розрахунково-пояснювальна записка з описанням структури керування системою, шляхи взаємодії компонентів системи (підсистем) між собою.
3	Аналіз вимог до підсистем	Залежно від часу, що буде виділятися на один інкремент	Технічне завдання на окрему підсистему для команди розробки.
4	Проектування програмної архітектури		Структура підсистеми, опис інтерфейсів для взаємодії з зовнішнім середовищем.
5	Розробка		Програмні модулі, що реалізують необхідні методи. Керівництво програміст
6	Програмне тестування		Тести. Керівництво користувача
7	Системне та приймальне тестування	18.01.2022 – 25.01.2022	Програмний продукт
8	Експлуатація та супровід	25.01.2022 – 15.01.2023	Технічне завдання на розширення системи. Документація щодо внесення змін та виправлень.

Архітектура

Архітектура проекту складається з двох основних частин, а саме система і репозиторій.

В репозиторії будуть зберігатися дані користувача і дані про картки.

Інформація зберігатиметься в файлі на пристрой користувача в форматі json.

Для доступу до функціоналу системи необхідно пройти авторизацію, користувач має ввести логін і пароль. Для забезпечення коректності введених даних буде проводитися перевірка та порівняння з даними користувача в репозиторії.

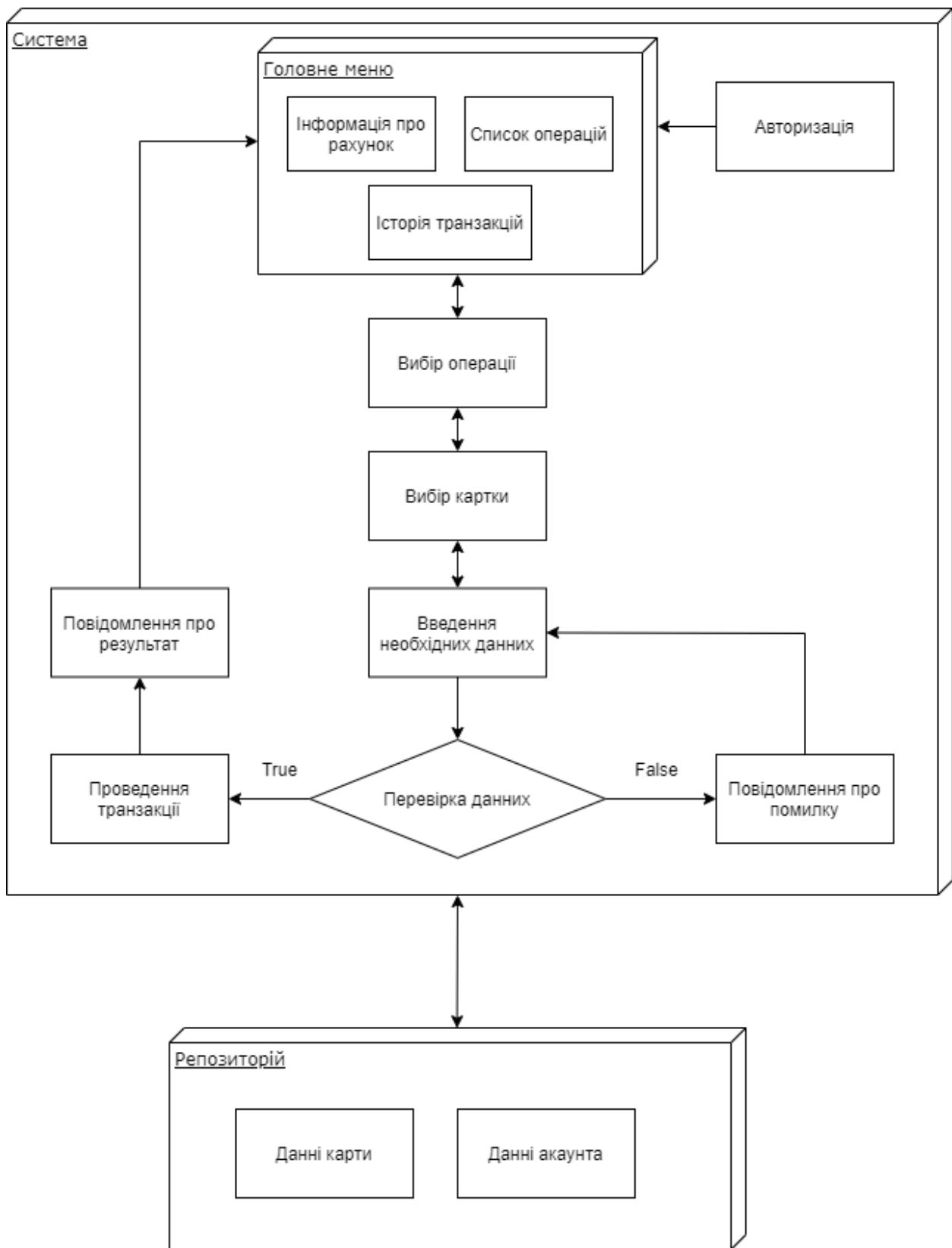
Після авторизації клієнт матиме доступ до головного меню. Інтерфейс буде надавати можливість змінити картку, переглянути інформацію про неї, її баланс та переглянути список можливих операцій.

Користувач зможе вибрати необхідну операцію, рахунок з якого відбудеться транзакція і ввести додаткові данні, якщо це потрібно. При необхідності буде можливість повернутися до головного меню.

Після підтвердження операції буде здійснюватися перевірка коректності введених даних, а також збір даних картки з репозиторію і перевірка необхідних умов для проведення транзакції.

При умові успішного завершення відбудеться процес переведення грошей і виведеться повідомлення про успішність, в протилежному випадку користувач отримає повідомлення з вказанням на помилку і поверне в меню введення даних транзакції.

Для даного проекту була складена блок схема (рис.1) на сторінці 6.



Програмний код

LoginForm.h

```
#pragma once
#include "MyForm.h"
namespace BankApp {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для LoginForm
    /// </summary>
public ref class LoginForm : public System::Windows::Forms::Form
{
private:
    AccountRepository^ repo = gcnew AccountRepository();
public:
    LoginForm(void)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~LoginForm()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::TextBox^ LoginTextBox;
private: System::Windows::Forms::TextBox^ PasswordTextBox;
protected:

private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Button^ button1;
private: System::ComponentModel::.IContainer^ components;
protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    ///
```

```

/// содержимое этого метода с помощью редактора кода.
/// </summary>
void InitializeComponent(void)
{
    this->components = (gcnew System::ComponentModel::Container());
    this->LoginTextBox = (gcnew System::Windows::Forms::TextBox());
    this->PasswordTextBox = (gcnew
System::Windows::Forms::TextBox());
    this->contextMenuStrip1 = (gcnew
System::Windows::Forms::ContextMenuStrip(this->components));
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->SuspendLayout();
    //
    // LoginTextBox
    //
    this->LoginTextBox->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->LoginTextBox->Location = System::Drawing::Point(105, 83);
    this->LoginTextBox->Name = L"LoginTextBox";
    this->LoginTextBox->Size = System::Drawing::Size(151, 32);
    this->LoginTextBox->TabIndex = 0;
    //
    // PasswordTextBox
    //
    this->PasswordTextBox->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->PasswordTextBox->Location = System::Drawing::Point(105,
130);
    this->PasswordTextBox->Name = L"PasswordTextBox";
    this->PasswordTextBox->PasswordChar = '*';
    this->PasswordTextBox->Size = System::Drawing::Size(151, 32);
    this->PasswordTextBox->TabIndex = 1;
    //
    // contextMenuStrip1
    //
    this->contextMenuStrip1->Name = L"contextMenuStrip1";
    this->contextMenuStrip1->Size = System::Drawing::Size(61, 4);
    //
    // label1
    //
    this->label1->AutoSize = true;
    this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Tai
Le", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->label1->Location = System::Drawing::Point(26, 86);
    this->label1->Name = L"label1";
    this->label1->Size = System::Drawing::Size(58, 23);
    this->label1->TabIndex = 3;
    this->label1->Text = L"Логін:";
    this->label1->Click += gcnew System::EventHandler(this,
&LoginForm::label1_Click);
    //
    // label2
    //
    this->label2->AutoSize = true;

```

```

        this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Tai
Le", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label2->Location = System::Drawing::Point(12, 133);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(77, 23);
        this->label2->TabIndex = 4;
        this->label2->Text = L"Пароль:";
        //
        // button1
        //
        this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft
Tai Le", 15.75F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->button1->Location = System::Drawing::Point(83, 186);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(123, 42);
        this->button1->TabIndex = 5;
        this->button1->Text = L"Увійти";
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gcnew System::EventHandler(this,
&LoginForm::button1_Click);
        //
        // LoginForm
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(284, 261);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->PasswordTextBox);
        this->Controls->Add(this->LoginTextBox);
        this->Name = L"LoginForm";
        this->Text = L"LoginForm";
        this->FormClosed += gcnew
System::Windows::Forms::FormClosedEventHandler(this,
&LoginForm::LoginForm_FormClosed);
        this->Load += gcnew System::EventHandler(this,
&LoginForm::LoginForm_Load);
        this->ResumeLayout(false);
        this->PerformLayout();
    }
}

#pragma endregion
private: System::Void label1_Click(System::Object^ sender,
System::EventArgs^ e) {
}
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    Account^ acc= this->repo->Login(LoginTextBox-
>Text, PasswordTextBox->Text);
    if (acc == nullptr) {
        MessageBox::Show("Невірний логін, або пароль");
    }
    else {
        MyForm^ myForm = gcnew MyForm();
        myForm->ParentForm = this;
        myForm->LoginAccount = acc;
    }
}

```

```

        myForm->Show();
    }
}
private: System::Void LoginForm_Load(System::Object^ sender,
System::EventArgs^ e)
{
}

private: System::Void LoginForm_FormClosed(System::Object^ sender,
System::Windows::Forms::FormClosedEventArgs^ e)
{
}

};

}

```

MyForm.h

```

#pragma once
#include "TransferForm.h"
#include "TopUpMobile.h"
#include "HistoryForm.h"
#include "Payments.h"
#include "CreditForm.h"
#include "DepositForm.h"
namespace BankApp {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Сводка для MyForm
/// </summary>
public ref class MyForm : public System::Windows::Forms::Form
{
private:
    Account^ loginAccount;
    AccountRepository^ repo = gcnew AccountRepository();
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button6;
private: System::Windows::Forms::Button^ button5;
    Form^ parrentForm;
public:
    MyForm(void)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
    }
    property Account^ LoginAccount
    {

```

```

        Account^ get() {
            return loginAccount;
        }
        void set(Account^acc) {
            loginAccount = acc;
        }
    }
    property Form^ ParrentForm
    {
        Form^ get() {
            return parrentForm;
        }
        void set(Form^val) {
            parrentForm = val;
            parrentForm->Hide();
        }
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
~MyForm()
{
    if (components)
    {
        delete components;
    }
}

protected:

private: System::Windows::Forms::GroupBox^ groupBox1;

protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)
{

```

```

System::ComponentModel::ComponentResourceManager^ resources =
(gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));
this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
this->button1 = (gcnew System::Windows::Forms::Button());
this->button2 = (gcnew System::Windows::Forms::Button());
this->button3 = (gcnew System::Windows::Forms::Button());
this->button4 = (gcnew System::Windows::Forms::Button());
this->button6 = (gcnew System::Windows::Forms::Button());
this->button5 = (gcnew System::Windows::Forms::Button());
this->SuspendLayout();
//
// groupBox1
//
this->groupBox1->Location = System::Drawing::Point(206, 12);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(635, 480);
this->groupBox1->TabIndex = 6;
this->groupBox1->TabStop = false;
this->groupBox1->Enter += gcnew System::EventHandler(this,
&MyForm::groupBox1_Enter);
//
// button1
//
this->button1->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->button1->ForeColor = System::Drawing::Color::Chocolate;
this->button1->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button1.Image")));
this->button1->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
this->button1->Location = System::Drawing::Point(12, 12);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(188, 74);
this->button1->TabIndex = 7;
this->button1->Text = L"Платежи";
this->button1-> TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click_1);
//
// button2
//
this->button2->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->button2->ForeColor = System::Drawing::Color::Chocolate;
this->button2->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button2.Image")));
this->button2->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
this->button2->Location = System::Drawing::Point(12, 332);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(188, 74);
this->button2->TabIndex = 8;
this->button2->Text = L"Історія";
this->button2-> TextAlign =
System::Drawing::ContentAlignment::MiddleRight;

```

```

        this->button2->UseVisualStyleBackColor = true;
        this->button2->Click += gcnew System::EventHandler(this,
&MyForm::button2_Click_1);
        //
        // button3
        //
        this->button3->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button3->ForeColor = System::Drawing::Color::Chocolate;
        this->button3->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button3.Image")));
        this->button3->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
        this->button3->Location = System::Drawing::Point(12, 252);
        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(188, 74);
        this->button3->TabIndex = 9;
        this->button3->Text = L"Поповнення мобільного";
        this->button3-> TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
        this->button3->UseVisualStyleBackColor = true;
        this->button3->Click += gcnew System::EventHandler(this,
&MyForm::button3_Click);
        //
        // button4
        //
        this->button4->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204));
        this->button4->ForeColor = System::Drawing::Color::Chocolate;
        this->button4->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button4.Image")));
        this->button4->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
        this->button4->Location = System::Drawing::Point(12, 92);
        this->button4->Name = L"button4";
        this->button4->Size = System::Drawing::Size(188, 74);
        this->button4->TabIndex = 10;
        this->button4->Text = L"Перекази";
        this->button4-> TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
        this->button4->UseVisualStyleBackColor = true;
        this->button4->Click += gcnew System::EventHandler(this,
&MyForm::button4_Click);
        //
        // button6
        //
        this->button6->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button6.BackgroundImage")));
        this->button6->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
        this->button6->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button6->ForeColor = System::Drawing::Color::Chocolate;
        this->button6->Location = System::Drawing::Point(12, 172);

```

```

        this->button6->Name = L"button6";
        this->button6->Size = System::Drawing::Size(188, 74);
        this->button6->TabIndex = 12;
        this->button6->Text = L"Кредит";
        this->button6-> TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
        this->button6->UseVisualStyleBackColor = true;
        this->button6->Click += gcnew System::EventHandler(this,
&MyForm::button6_Click);
        //
        // button5
        //
        this->button5->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
        this->button5->ForeColor = System::Drawing::Color::Chocolate;
        this->button5->Location = System::Drawing::Point(12, 413);
        this->button5->Name = L"button5";
        this->button5->Size = System::Drawing::Size(188, 68);
        this->button5->TabIndex = 13;
        this->button5->Text = L"Депозит";
        this->button5-> TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
        this->button5->UseVisualStyleBackColor = true;
        this->button5->Click += gcnew System::EventHandler(this,
&MyForm::button5_Click);
        //
        // MyForm
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::Font;
        this->ClientSize = System::Drawing::Size(853, 504);
        this->Controls->Add(this->button5);
        this->Controls->Add(this->button6);
        this->Controls->Add(this->button4);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->groupBox1);
        this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedDialog;
        this->MaximizeBox = false;
        this->MinimizeBox = false;
        this->Name = L"MyForm";
        this->Text = L"MyForm";
        this->Load += gcnew System::EventHandler(this,
&MyForm::MyForm_Load);
        this->ResumeLayout(false);

    }
#pragma endregion
public: void UploadCards()
{
    if (LoginAccount != nullptr)
    {
        LoginAccount = repo->Login(LoginAccount->Login, LoginAccount-
>Password);
        groupBox1->Controls->Clear();
        int y = 20;
        for (int i = 0; i < LoginAccount->Cards->Count; ++i)
        {

```

```

        GroupBox^ panel = LoginAccount->Cards[i]-
>CreateCardPanel(Point(50, y));
        y += 140;
        groupBox1->Controls->Add(panel);
    }
}

private: System::Void MyForm_Load(System::Object^ sender,
System::EventArgs^ e)
{
    UploadCards();
}
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
}

private: System::Void groupBox1_Enter(System::Object^ sender,
System::EventArgs^ e)
{
}

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e)
{
}

private: System::Void button4_Click(System::Object^ sender,
System::EventArgs^ e)
{
    TransferForm^ form = gcnew TransferForm();
    form->Cards = LoginAccount->Cards;
    form->>ShowDialog();
    UploadCards();
}
private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e)
{
    TopUpMobile^ form = gcnew TopUpMobile();
    form->Cards = loginAccount->Cards;
    form->PhoneNumber = loginAccount->NumberPhone;
    form->>ShowDialog();
    UploadCards();
}
private: System::Void button2_Click_1(System::Object^ sender,
System::EventArgs^ e)
{
    HistoryForm^ form = gcnew HistoryForm();
    form->Cards = loginAccount->Cards;
    form->>ShowDialog();
}

private: System::Void button1_Click_1(System::Object^ sender,
System::EventArgs^ e)
{
    Payments^ pay = gcnew Payments();
    pay->Cards = loginAccount->Cards;
    pay->>ShowDialog();
    UploadCards();
    repo->Save();
}
private: System::Void button6_Click(System::Object^ sender,
System::EventArgs^ e)
{
}

```

```

    {
        CreditForm^ form = gcnew CreditForm();
        form->Cards = loginAccount->Cards;
        form->>ShowDialog();
        UploadCards();
        repo->Save();
    }

    private: System::Void button5_Click(System::Object^ sender,
System::EventArgs^ e)
{
    DepositForm^ form = gcnew DepositForm();
    form->LoginAccount = loginAccount;
    form->>ShowDialog();
    UploadCards();
    repo->Save();
}
};

}

```

Payments.h

```

#pragma once
namespace BankApp {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections::Generic;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для Payments
    /// </summary>
    public ref class Payments : public System::Windows::Forms::Form
    {
        HistoryRepository^ repo = gcnew HistoryRepository();
        List<Card^>^ cards;

    public:

        Payments(void)
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }
        property List<Card^>^ Cards
        {
            List<Card^>^ get() {
                return cards;
            }
            void set(List<Card^>^ cards) {
                this->cards = cards;
            }
        }
    }

    protected:

```

```

/// <summary>
/// Освободить все используемые ресурсы.
/// </summary>
~Payments()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::RadioButton^ radioButton1;
protected:
private: System::Windows::Forms::RadioButton^ radioButton2;
private: System::Windows::Forms::RadioButton^ radioButton3;
private: System::Windows::Forms::RadioButton^ radioButton4;
private: System::Windows::Forms::RadioButton^ radioButton5;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Button^ button1;

protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
void InitializeComponent(void)
{
    this->radioButton1 = (gcnew
System::Windows::Forms::RadioButton());
    this->radioButton2 = (gcnew
System::Windows::Forms::RadioButton());
    this->radioButton3 = (gcnew
System::Windows::Forms::RadioButton());
    this->radioButton4 = (gcnew
System::Windows::Forms::RadioButton());
    this->radioButton5 = (gcnew
System::Windows::Forms::RadioButton());
    this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->SuspendLayout();
    //
    // radioButton1
    //
    this->radioButton1->AutoSize = true;
}

```

```

    this->radioButton1->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->radioButton1->Location = System::Drawing::Point(60, 69);
    this->radioButton1->Name = L"radioButton1";
    this->radioButton1->Size = System::Drawing::Size(98, 27);
    this->radioButton1->TabIndex = 0;
    this->radioButton1->TabStop = true;
    this->radioButton1->Text = L"Штрафи";
    this->radioButton1->UseVisualStyleBackColor = true;
//
// radioButton2
//
    this->radioButton2->AutoSize = true;
    this->radioButton2->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->radioButton2->Location = System::Drawing::Point(60, 36);
    this->radioButton2->Name = L"radioButton2";
    this->radioButton2->Size = System::Drawing::Size(193, 27);
    this->radioButton2->TabIndex = 1;
    this->radioButton2->TabStop = true;
    this->radioButton2->Text = L"Комунальні послуги";
    this->radioButton2->UseVisualStyleBackColor = true;
//
// radioButton3
//
    this->radioButton3->AutoSize = true;
    this->radioButton3->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->radioButton3->Location = System::Drawing::Point(60, 111);
    this->radioButton3->Name = L"radioButton3";
    this->radioButton3->Size = System::Drawing::Size(102, 27);
    this->radioButton3->TabIndex = 2;
    this->radioButton3->TabStop = true;
    this->radioButton3->Text = L"Інтернет";
    this->radioButton3->UseVisualStyleBackColor = true;
//
// radioButton4
//
    this->radioButton4->AutoSize = true;
    this->radioButton4->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->radioButton4->Location = System::Drawing::Point(60, 155);
    this->radioButton4->Name = L"radioButton4";
    this->radioButton4->Size = System::Drawing::Size(138, 27);
    this->radioButton4->TabIndex = 3;
    this->radioButton4->TabStop = true;
    this->radioButton4->Text = L"Телебачення";
    this->radioButton4->UseVisualStyleBackColor = true;
//
// radioButton5
//
    this->radioButton5->AutoSize = true;
    this->radioButton5->Font = (gcnew
System::Drawing::Font(L"Microsoft Tai Le", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

```

```

        static_cast<System::Byte>(0)));
this->radioButton5->Location = System::Drawing::Point(60, 198);
this->radioButton5->Name = L"radioButton5";
this->radioButton5->Size = System::Drawing::Size(147, 27);
this->radioButton5->TabIndex = 4;
this->radioButton5->TabStop = true;
this->radioButton5->Text = L"Благодійність";
this->radioButton5->UseVisualStyleBackColor = true;
//
// comboBox1
//
this->comboBox1->Font = (gcnew System::Drawing::Font(L"Microsoft
Tai Le", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Location = System::Drawing::Point(80, 262);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(193, 31);
this->comboBox1->TabIndex = 5;
//
// textBox1
//
this->textBox1->Font = (gcnew System::Drawing::Font(L"Microsoft
Tai Le", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
this->textBox1->Location = System::Drawing::Point(80, 311);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(193, 32);
this->textBox1->TabIndex = 6;
//
// label1
//
this->label1->AutoSize = true;
this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Tai
Le", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
this->label1->Location = System::Drawing::Point(6, 267);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(64, 21);
this->label1->TabIndex = 7;
this->label1->Text = L"Картка";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Tai
Le", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
this->label2->Location = System::Drawing::Point(22, 311);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(48, 21);
this->label2->TabIndex = 8;
this->label2->Text = L"Сума";
//
// button1
//
this->button1->BackColor = System::Drawing::Color::Gold;

```

```

        this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft
Tai Le", 15.75F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
        this->button1->ForeColor = System::Drawing::Color::Brown;
        this->button1->Location = System::Drawing::Point(53, 364);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(200, 36);
        this->button1->TabIndex = 9;
        this->button1->Text = L"Оплатити";
        this->button1->UseVisualStyleBackColor = false;
        this->button1->Click += gcnew System::EventHandler(this,
&Payments::button1_Click);
    //
    // Payments
    //
    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
    this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::Font;
    this->ClientSize = System::Drawing::Size(322, 412);
    this->Controls->Add(this->button1);
    this->Controls->Add(this->label2);
    this->Controls->Add(this->label1);
    this->Controls->Add(this->textBox1);
    this->Controls->Add(this->comboBox1);
    this->Controls->Add(this->radioButton5);
    this->Controls->Add(this->radioButton4);
    this->Controls->Add(this->radioButton3);
    this->Controls->Add(this->radioButton2);
    this->Controls->Add(this->radioButton1);
    this->Name = L"Payments";
    this->Text = L"Payments";
    this->Load += gcnew System::EventHandler(this,
&Payments::Payments_Load);
    this->ResumeLayout(false);
    this->PerformLayout();

}
#pragma endregion
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    try
    {
        int index = comboBox1->SelectedIndex;
        double sum = Convert::.ToDouble(textBox1->Text);
        if (sum <= Cards[index]->Balance && sum > 0)
        {
            Cards[index]->Balance -= sum;
            String^ message = CreateHistoryMessage();
            History^ history = gcnew History();
            history->Operation = message;
            history->CardNumver = Cards[index]->CardNumber;
            history->Date = DateTime::Today;
            history->Sum = sum;
            repo->Create(history);
            MessageBox::Show("Оплату виконано успішно!", "Увага!",
MessageBoxButtons::OK);
            this->Close();
        }
    }
    catch(...)
```

```

    {
        MessageBox::Show("Під час транзакції виникла помилка! Перевірте
коректність даних", "Увага!", MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
}

private: System::Void Payments_Load(System::Object^ sender,
System::EventArgs^ e)
{
    if (Cards != nullptr)
    {
        for (int i = 0; i < Cards->Count; ++i)
        {
            comboBox1->Items->Add(Cards[i]->CardNumber);
        }
    }
}
private: String^ CreateHistoryMessage()
{
    String^ message=nullptr;
    if (radioButton1->Checked) {
        message = "Оплата штрафу";
    }
    else if(radioButton2->Checked)
    {
        message = "Оплата комунальних послуг";
    }
    else if (radioButton3->Checked) {
        message = "Оплата інтернету";
    }
    else if (radioButton4->Checked)
    {
        message = "Оплата телебачення";
    }
    else if (radioButton5->Checked)
    {
        message = "Переведення коштів на благодійність";
    }
    return message;
}
};

}

```

TransferForm.h

```

#pragma once
#include"MyForm.h"
namespace BankApp {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections::Generic;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Сводка для TransferForm
/// </summary>
public ref class TransferForm : public System::Windows::Forms::Form
{

```

```

List<Card^>^ cards;
AccountRepository^ accRepo = gcnew AccountRepository();
HistoryRepository^ hisRepo = gcnew HistoryRepository();
//MyForm^ parrentForm=gcnew MyForm();
public:
    /*property MyForm^ ParrentForm
    {
        MyForm^ get() {
            return parrentForm;
        }
        void set(MyForm^ val) {
            parrentForm = val;
            parrentForm->Hide();
        }
    }*/
}

property List<Card^>^ Cards
{
    List<Card^>^ get() {
        return cards;
    }
    void set(List<Card^>^ cards) {
        this->cards = cards;
    }
}
TransferForm(void)
{
    InitializeComponent();
    //
    //TODO: добавьте код конструктора
    //
}

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
~TransferForm()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Button^ button1;
protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте

```

```

/// содержимое этого метода с помощью редактора кода.
/// </summary>
void InitializeComponent(void)
{
    this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->textBox2 = (gcnew System::Windows::Forms::TextBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->label3 = (gcnew System::Windows::Forms::Label());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->SuspendLayout();
    //
    // comboBox1
    //
    this->comboBox1->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->comboBox1->FormattingEnabled = true;
    this->comboBox1->Location = System::Drawing::Point(62, 46);
    this->comboBox1->Name = L"comboBox1";
    this->comboBox1->Size = System::Drawing::Size(217, 31);
    this->comboBox1->TabIndex = 0;
    this->comboBox1->SelectedIndexChanged += gcnew
System::EventHandler(this, &TransferForm::comboBox1_SelectedIndexChanged);
    //
    // textBox1
    //
    this->textBox1->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->textBox1->Location = System::Drawing::Point(62, 125);
    this->textBox1->Multiline = true;
    this->textBox1->Name = L"textBox1";
    this->textBox1->Size = System::Drawing::Size(217, 29);
    this->textBox1->TabIndex = 1;
    this->textBox1->Text = L"0";
    this->textBox1->TextChanged += gcnew System::EventHandler(this,
&TransferForm::textBox1_TextChanged);
    //
    // textBox2
    //
    this->textBox2->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->textBox2->Location = System::Drawing::Point(62, 191);
    this->textBox2->Multiline = true;
    this->textBox2->Name = L"textBox2";
    this->textBox2->Size = System::Drawing::Size(217, 30);
    this->textBox2->TabIndex = 2;
    this->textBox2->Text = L"0";
    this->textBox2->TextChanged += gcnew System::EventHandler(this,
&TransferForm::textBox2_TextChanged);
    //
    // label1
    //
    this->label1->AutoSize = true;
    this->label1->Font = (gcnew System::Drawing::Font(L"Cambria",
14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,

```

```

        static_cast<System::Byte>(204)));
this->label1->Location = System::Drawing::Point(58, 21);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(131, 22);
this->label1->TabIndex = 3;
this->label1->Text = L"Оберіть карту";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Font = (gcnew System::Drawing::Font(L"Cambria", 12,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->label2->Location = System::Drawing::Point(58, 93);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(252, 19);
this->label2->TabIndex = 4;
this->label2->Text = L"Введіть номер карти отримувача";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Font = (gcnew System::Drawing::Font(L"Cambria",
14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->label3->Location = System::Drawing::Point(58, 166);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(53, 22);
this->label3->TabIndex = 5;
this->label3->Text = L"Сума";
//
// button1
//
this->button1->BackColor = System::Drawing::Color::ForestGreen;
this->button1->Font = (gcnew System::Drawing::Font(L"Cambria",
15.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->button1->Location = System::Drawing::Point(91, 237);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(149, 43);
this->button1->TabIndex = 6;
this->button1->Text = L"Перевести";
this->button1->UseVisualStyleBackColor = false;
this->button1->Click += gcnew System::EventHandler(this,
&TransferForm::button1_Click);
//
// TransferForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::Font;
this->ClientSize = System::Drawing::Size(351, 292);
this->Controls->Add(this->button1);
this->Controls->Add(this->label3);
this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Controls->Add(this->textBox2);
this->Controls->Add(this->textBox1);
this->Controls->Add(this->comboBox1);
this->MaximizeBox = false;
this->MinimizeBox = false;

```

```

        this->Name = L"TransferForm";
        this->Text = L"TransferForm";
        this->Load += gcnew System::EventHandler(this,
&TransferForm::TransferForm_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }

#pragma endregion
private:
void DeleteBadSymbols(TextBox^textBox)
{
    try {
        int length = textBox->Text->Length;

        for (int i = 0; i < length; ++i)
        {
            char symbol = textBox->Text[i];
            if ((symbol < '0' || symbol> '9')&&symbol!='.')
            {
                textBox->Text = textBox->Text->Remove(i, 1);
            }
        }
    }
    catch (...) {}
}

private: System::Void comboBox1_SelectedIndexChanged(System::Object^
sender, System::EventArgs^ e) {
}

private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    String^ from = (String^)comboBox1->SelectedItem;
    String^ to = textBox1->Text;
    String^ sum = textBox2->Text;

    if (from != nullptr)
    {
        if (to != "0")
        {
            if (sum != "0")
            {
                try {
                    double val = Convert::.ToDouble(sum);
                    bool res = this->accRepo->TransferMoney(from, to,
val);
                    if (!res)
                    {
                        MessageBox::Show("Під час транзакції виникла
помилка! Перевірте коректність даних", "Увага!", MessageBoxButtons::OK,
MessageBoxIcon::Error);
                    }
                    else
                    {
                        MessageBox::Show("Транзакцію виконано успішно!",
"Увага!", MessageBoxButtons::OK);
                    }
                }
            }
        }
    }
}

DateTime date = DateTime::Today;

History^ newHistory = gcnew History();
newHistory->CardNumver = from;
newHistory->Date = date;
newHistory->Operation = "Переведення коштів на
карту " + to;
}

```

```

        newHistory->Sum = val;

        History^ newHist = gcnew History();
        newHist->CardNumver = to;
        newHist->Date = date;
        newHist->Operation = "Поповнення картки ";
        newHist->Sum = val;

        hisRepo->Create(newHistory);
        hisRepo->Create(newHist);
        this->Close();
    }

}

catch (...)
{
    MessageBox::Show("Під час транзакції виникла помилка!
Перевірте коректність даних", "Увага!", MessageBoxButtons::OK,
MessageBoxIcon::Error);
}

}

else MessageBox::Show("Вкажіть суму", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Warning);
}

else MessageBox::Show("Вкажіть карту
отримувача", "Увага!", MessageBoxButtons::OK, MessageBoxIcon::Warning);
}

else MessageBox::Show("Вкажіть карту", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Warning);
}

private: System::Void TransferForm_Load(System::Object^ sender,
System::EventArgs^ e)
{
    if(Cards != nullptr)
    {
        for (int i = 0; i < Cards->Count; ++i)
        {
            comboBox1->Items->Add(Cards[i]->CardNumber);
        }
    }
}

private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e)
{
    DeleteBadSymbols(textBox1);
}

private: System::Void textBox2_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
    DeleteBadSymbols(textBox2);
}
};

}
}

```

CreditForm.h

```

#pragma once

namespace BankApp {

    using namespace System;
    using namespace System::ComponentModel;

```

```

using namespace System::Collections::Generic;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Сводка для CreditForm
/// </summary>
public ref class CreditForm : public System::Windows::Forms::Form
{
    double maxCreditSum = 50000;
    List<Card^>^ cards;
    HistoryRepository^ repo = gcnew HistoryRepository();
public:
    CreditForm(void)
    {
        InitializeComponent();
        // 
        //TODO: добавьте код конструктора
        //
    }
    property List<Card^>^ Cards
    {
        List<Card^>^ get() {
            return cards;
        }
        void set(List<Card^>^ cards) {
            this->cards = cards;
        }
    }
}

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
~CreditForm()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::RadioButton^ radioButton1;
protected:
private: System::Windows::Forms::RadioButton^ radioButton2;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора – не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>

```

```

void InitializeComponent(void)
{
    radioButton1 = (gcnew
System::Windows::Forms::RadioButton());
    radioButton2 = (gcnew
System::Windows::Forms::RadioButton());
    button1 = (gcnew System::Windows::Forms::Button());
    textBox2 = (gcnew System::Windows::Forms::TextBox());
    comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    label1 = (gcnew System::Windows::Forms::Label());
    label2 = (gcnew System::Windows::Forms::Label());
    SuspendLayout();
    //
    // radioButton1
    //
    this->radioButton1->AutoSize = true;
    this->radioButton1->Font = (gcnew
System::Drawing::Font(L"Candara", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
    this->radioButton1->Location = System::Drawing::Point(81, 103);
    this->radioButton1->Name = L"radioButton1";
    this->radioButton1->Size = System::Drawing::Size(159, 27);
    this->radioButton1->TabIndex = 0;
    this->radioButton1->TabStop = true;
    this->radioButton1->Text = L"Видача кредиту";
    this->radioButton1->UseVisualStyleBackColor = true;
    //
    // radioButton2
    //
    this->radioButton2->AutoSize = true;
    this->radioButton2->Font = (gcnew
System::Drawing::Font(L"Candara", 14.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
    this->radioButton2->Location = System::Drawing::Point(81, 152);
    this->radioButton2->Name = L"radioButton2";
    this->radioButton2->Size = System::Drawing::Size(167, 27);
    this->radioButton2->TabIndex = 1;
    this->radioButton2->TabStop = true;
    this->radioButton2->Text = L"Погасити кредит";
    this->radioButton2->UseVisualStyleBackColor = true;
    this->radioButton2->CheckedChanged += gcnew
System::EventHandler(this, &CreditForm::radioButton2_CheckedChanged);
    //
    // button1
    //
    button1->Location = System::Drawing::Point(103, 254);
    button1->Name = L"button1";
    button1->Size = System::Drawing::Size(124, 43);
    button1->TabIndex = 2;
    button1->Text = L"Виконати";
    button1->UseVisualStyleBackColor = true;
    button1->Click += gcnew System::EventHandler(this,
&CreditForm::button1_Click);
    //
    // textBox2
    //
    textBox2->Font = (gcnew System::Drawing::Font(L"Comic Sans
MS", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
    textBox2->Location = System::Drawing::Point(131, 212);
}

```

```

this->textBox2->Multiline = true;
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(193, 36);
this->textBox2->TabIndex = 4;
//
// comboBox1
//
this->comboBox1->Font = (gcnew System::Drawing::Font(L"Comic Sans MS", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Location = System::Drawing::Point(66, 40);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(193, 34);
this->comboBox1->TabIndex = 5;
//
// label1
//
this->label1->AutoSize = true;
this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Tai Le", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label1->Location = System::Drawing::Point(12, 218);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(102, 21);
this->label1->TabIndex = 6;
this->label1->Text = L"Сума видачі";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Tai Le", 11.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label2->Location = System::Drawing::Point(66, 13);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(114, 19);
this->label2->TabIndex = 7;
this->label2->Text = L"Оберіть картку";
//
// CreditForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::Font;
this->ClientSize = System::Drawing::Size(336, 309);
this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Controls->Add(this->comboBox1);
this->Controls->Add(this->textBox2);
this->Controls->Add(this->button1);
this->Controls->Add(this->radioButton2);
this->Controls->Add(this->radioButton1);
this->Name = L"CreditForm";
this->Text = L"CreditForm";
this->Load += gcnew System::EventHandler(this,
&CreditForm::CreditForm_Load);
this->ResumeLayout(false);
this->PerformLayout();

```

```

        }
    #pragma endregion
    private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        try {
            int index = comboBox1->SelectedIndex;
            History^ badHistory=nullptr;
            List<History^>^ histories = repo-
>GetHistoriesByCardNumber(Cards[index]->CardNumber);
            if (histories->Count!=0) {
                for (int i = 0; i < histories->Count; ++i)
                {
                    if (histories[i]->IsCredit) { // перевірка
наявності кредиту у користувача
                        badHistory = histories[i];
                    }
                }
            }
            if (radioButton1->Checked)
            {
                if (badHistory!=nullptr)
                {
                    MessageBox::Show("Ми не можемо видати вам кредит, \nтак як
ви не погасили минулій борг", "Увага!", MessageBoxButtons::OK,
MessageBoxIcon::Error);
                }
                else
                {
                    double sum = Convert::.ToDouble(textBox2->Text);
                    if (sum <= maxCreditSum)
                    {
                        Cards[index]->Balance += sum;

                        History^ newHistory = gcnew History();
                        newHistory->CardNumver = Cards[index]->CardNumber;
                        newHistory->Date = DateTime::Today;
                        newHistory->Operation = "Видача кредиту ";
                        newHistory->Sum = sum;
                        newHistory->IsCredit = true;
                        repo->Create(newHistory);
                        MessageBox::Show("Кредит успішно видано!", "Увага!",
MessageBoxButtons::OK);
                    }
                    else {
                        MessageBox::Show("Завелика сума! Максимальна сума
кредиту "+maxCreditSum.ToString()+" грн", "Увага!", MessageBoxButtons::OK,
MessageBoxIcon::Error);
                    }
                }
            }
        }
        else if(radioButton2->Checked)
        {
            if (badHistory != nullptr)
            {
                if (Cards[index]->Balance >= badHistory->Sum)
                {
                    Cards[index]->Balance -= badHistory->Sum;
                    badHistory->IsCredit = false;

                    History^ newHistory = gcnew History();
                    newHistory->CardNumver = Cards[index]->CardNumber;

```

```

newHistory->Date = DateTime::Today;
newHistory->Operation = "Погашення кредиту ";
newHistory->Sum = badHistory->Sum;
newHistory->IsCredit = false;
repo->Create(newHistory);

MessageBox::Show("Кредит успішно погашено! :)",
"Увага!", MessageBoxButtons::OK);

}

else
{
    MessageBox::Show("Недостатньо коштів для погашення
кредиту!", "Увага!", MessageBoxButtons::OK, MessageBoxIcon::Error);
}

else
{
    MessageBox::Show("У вас немає кредитів!", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Error);
}

this->Close();
}

catch (Exception^e)
{
    MessageBox::Show(e->Message);
    MessageBox::Show("Не вдалося виконати операцію з
кредитом, перевірте коректність введених даних", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Error);
}

private: System::Void radioButton2_CheckedChanged(System::Object^ sender,
System::EventArgs^ e)
{
    label1->Visible = !radioButton2->Checked;
    textBox2->Visible = !radioButton2->Checked;
}

private: System::Void CreditForm_Load(System::Object^ sender,
System::EventArgs^ e) {
    if (Cards != nullptr)
    {
        for (int i = 0; i < Cards->Count; ++i)
        {
            comboBox1->Items->Add(Cards[i]->CardNumber);
        }
    }
}
};

}

```

TopUpMobile.h

```

#pragma once
namespace BankApp {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections::Generic;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

```

```

/// <summary>
/// Сводка для TopUpMobile
/// </summary>
public ref class TopUpMobile : public System::Windows::Forms::Form
{
    List<Card^>^ cards;
    AccountRepository^ accRepo = gcnew AccountRepository();
    HistoryRepository^ hisRepo = gcnew HistoryRepository();
    String^ phoneNumber;
public:
    TopUpMobile(void)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
    }
    property List<Card^>^ Cards
    {
        List<Card^>^ get() {
            return cards;
        }
        void set(List<Card^>^ cards) {
            this->cards = cards;
        }
    }
    property String^ PhoneNumber {
        String^ get() {
            return phoneNumber;
        }
        void set(String^val) {
            phoneNumber = val;
        }
    }
protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~TopUpMobile()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Button^ button1;
protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

```

```

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора – не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->textBox2 = (gcnew System::Windows::Forms::TextBox());
        this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->button1 = (gcnew System::Windows::Forms::Button());
        this->SuspendLayout();
        //
        // textBox1
        //
        this->textBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 15.75F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
        this->textBox1->Location = System::Drawing::Point(147, 122);
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(170, 31);
        this->textBox1->TabIndex = 0;
        this->textBox1->TextChanged += gcnew System::EventHandler(this,
&TopUpMobile::textBox1_TextChanged);
        //
        // textBox2
        //
        this->textBox2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 15.75F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
        this->textBox2->Location = System::Drawing::Point(147, 192);
        this->textBox2->Name = L"textBox2";
        this->textBox2->Size = System::Drawing::Size(170, 31);
        this->textBox2->TabIndex = 1;
        this->textBox2->Text = L"0";
        this->textBox2->TextChanged += gcnew System::EventHandler(this,
&TopUpMobile::textBox2_TextChanged);
        //
        // comboBox1
        //
        this->comboBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 15.75F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
        this->comboBox1->FormattingEnabled = true;
        this->comboBox1->Location = System::Drawing::Point(147, 59);
        this->comboBox1->Name = L"comboBox1";
        this->comboBox1->Size = System::Drawing::Size(170, 33);
        this->comboBox1->TabIndex = 2;
        this->comboBox1->SelectedIndexChanged += gcnew
System::EventHandler(this, &TopUpMobile::comboBox1_SelectedIndexChanged);
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));

```

```

        this->label1->Location = System::Drawing::Point(64, 68);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(77, 24);
        this->label1->TabIndex = 3;
        this->label1->Text = L"Картка:";

        //
        // label2
        //
        this->label2->AutoSize = true;
        this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 11.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                     static_cast<System::Byte>(204)));
        this->label2->Location = System::Drawing::Point(9, 130);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(132, 18);
        this->label2->TabIndex = 4;
        this->label2->Text = L"Номер телефону:";

        //
        // label3
        //
        this->label3->AutoSize = true;
        this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 11.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                     static_cast<System::Byte>(204)));
        this->label3->Location = System::Drawing::Point(6, 200);
        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(135, 18);
        this->label3->TabIndex = 5;
        this->label3->Text = L"Сума поповнення:";

        //
        // button1
        //
        this->button1->Location = System::Drawing::Point(134, 253);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(122, 50);
        this->button1->TabIndex = 6;
        this->button1->Text = L"Поповнити";
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gcnew System::EventHandler(this,
&TopUpMobile::button1_Click);

        //
        // TopUpMobile
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoSizeMode = System::Windows::Forms::AutoSizeMode::Font;
        this->ClientSize = System::Drawing::Size(413, 330);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->comboBox1);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Name = L"TopUpMobile";
        this->Text = L"TopUpMobile";
        this->Load += gcnew System::EventHandler(this,
&TopUpMobile::TopUpMobile_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }

#pragma endregion

```

```

private : void DeleteBadSymbols(TextBox^ textBox)
{
    try {
        int length = textBox->Text->Length;

        for (int i = 0; i < length; ++i)
        {
            char symbol = textBox->Text[i];
            if ((symbol < '0' || symbol > '9') && symbol != '.')
            {
                textBox->Text = textBox->Text->Remove(i, 1);
            }
        }
    }
    catch (...) {}
}

private: System::Void TopUpMobile_Load(System::Object^ sender,
System::EventArgs^ e)
{
    textBox1->Text = PhoneNumber;
    if (Cards != nullptr)
    {
        for (int i = 0; i < Cards->Count; ++i)
        {
            comboBox1->Items->Add(Cards[i]->CardNumber);
        }
    }
}
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    String^ from = (String^)comboBox1->SelectedItem;
    String^ phone = textBox1->Text;
    String^ sum = textBox2->Text;

    if (from != nullptr)
    {
        if (phone != "")
        {
            if (sum != "0")
            {
                try {
                    double val = Convert::.ToDouble(sum);

                    Card^ card;
                    for (int i = 0; i < Cards->Count; ++i)
                    {
                        if (Cards[i]->CardNumber == from) {
                            card = Cards[i];
                        }
                    }
                    if (val <= card->Balance) {

                        card->Balance -= val;
                        MessageBox::Show("Мобільний
поповнено!", "Увага!", MessageBoxButtons::OK);
                }
            }
        }
    }
}

History^ newHistory = gcnew
History();
newHistory->CardNumver = from;
newHistory->Date = DateTime::Today;

```

```

        newHistory->Operation = "Поповнення
мобільного ";

        newHistory->Sum = val;
hisRepo->Create(newHistory);

        this->Close();
    }
    else {
        MessageBox::Show("Під час
транзакції виникла помилка! Перевірте коректність даних", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
    catch(...)
{
    MessageBox::Show("Під час транзакції
виникла помилка! Перевірте коректність даних", "Увага!", MessageBoxButtons::OK,
MessageBoxIcon::Error);
}

    else MessageBox::Show("Вкажіть суму", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Warning);
}
else MessageBox::Show("Вкажіть номер телефону", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Warning);
}
else MessageBox::Show("Вкажіть карту", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Warning);
}

private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
    DeleteBadSymbols(textBox1);
}
private: System::Void textBox2_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
    DeleteBadSymbols(textBox2);
}
private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender,
System::EventArgs^ e) {
}
};

}

```

HistoryForm.h

```

#pragma once

namespace BankApp {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections::Generic;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Сводка для HistoryForm
/// </summary>

```

```

public ref class HistoryForm : public System::Windows::Forms::Form
{
    HistoryRepository^ repo=gcnew HistoryRepository();
    List<Card^>^ cards;
public:
    property List<Card^>^ Cards
    {
        List<Card^>^ get() {
            return cards;
        }
        void set(List<Card^>^ cards) {
            this->cards = cards;
        }
    }
    HistoryForm(void)
    {
        InitializeComponent();
        // 
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
~HistoryForm()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::ListBox^ listBox1;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::Label^ label1;
protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора – не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
void InitializeComponent(void)
{
    this->listBox1 = (gcnew System::Windows::Forms::ListBox());
    this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->SuspendLayout();
    //
    // listBox1
    //
    this->listBox1->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 9.75F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
}

```

```

        this->listBox1->FormattingEnabled = true;
        this->listBox1->ItemHeight = 16;
        this->listBox1->Location = System::Drawing::Point(28, 87);
        this->listBox1->Name = L"listBox1";
        this->listBox1->Size = System::Drawing::Size(516, 244);
        this->listBox1->TabIndex = 0;
        this->listBox1->SelectedIndexChanged += gcnew
System::EventHandler(this, &HistoryForm::listBox1_SelectedIndexChanged);
        //
        // comboBox1
        //
        this->comboBox1->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
        this->comboBox1->FormattingEnabled = true;
        this->comboBox1->Location = System::Drawing::Point(233, 46);
        this->comboBox1->Name = L"comboBox1";
        this->comboBox1->Size = System::Drawing::Size(181, 32);
        this->comboBox1->TabIndex = 1;
        this->comboBox1->SelectedIndexChanged += gcnew
System::EventHandler(this, &HistoryForm::comboBox1_SelectedIndexChanged);
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
        this->label1->Location = System::Drawing::Point(164, 49);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(63, 24);
        this->label1->TabIndex = 2;
        this->label1->Text = L"Kapta";
        //
        // HistoryForm
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::Font;
        this->ClientSize = System::Drawing::Size(575, 363);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->comboBox1);
        this->Controls->Add(this->listBox1);
        this->Name = L"HistoryForm";
        this->Text = L"HistoryForm";
        this->Load += gcnew System::EventHandler(this,
&HistoryForm::HistoryForm_Load);
        this->ResumeLayout(false);
        this->PerformLayout();
    }

}
#pragma endregion
private: System::Void comboBox1_SelectedIndexChanged(System::Object^
sender, System::EventArgs^ e)
{
    listBox1->Items->Clear();
    String^ card = (String^)comboBox1->SelectedItem;
    List<History^>^ hist = repo->GetHistoriesByCardNumber(card);
    for (int i = 0; i < hist->Count; ++i)
    {

```

```

        listBox1->Items->Add(hist[i]->Date->ToString() + " " + hist[i]-
>Operation + " " + hist[i]->Sum + " грн");
    }
}
private: System::Void listBox1_SelectedIndexChanged(System::Object^
sender, System::EventArgs^ e)
{
}
private: System::Void HistoryForm_Load(System::Object^ sender,
System::EventArgs^ e)
{
    if (Cards != nullptr)
    {
        for (int i = 0; i < Cards->Count; ++i)
        {
            comboBox1->Items->Add(Cards[i]->CardNumber);
        }
    }
}
};

}

```

DepositForm.h

```

#pragma once

namespace BankApp {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Сводка для DepositForm
/// </summary>
public ref class DepositForm : public System::Windows::Forms::Form
{
    Account^ loginAccount;
    AccountRepository^ accRepo = gcnew AccountRepository();
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::TextBox^ textBox2;
    HistoryRepository^ hisRepo = gcnew HistoryRepository();
    DepositRepository^ depRepo = gcnew DepositRepository();
public:
    DepositForm(void)
    {
        InitializeComponent();
        // 
        // TODO: добавьте код конструктора
        //
    }
    property Account^ LoginAccount
    {
        Account^ get()
        {
            return loginAccount;
        }
        void set(Account^ acc)

```

```

        {
            loginAccount = acc;
        }
    }
protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
~DepositForm()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::ComboBox^ comboBox1;
protected:
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::RadioButton^ radioButton1;
private: System::Windows::Forms::RadioButton^ radioButton2;
private: System::Windows::Forms::Button^ button1;

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора – не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
void InitializeComponent(void)
{
    this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->radioButton1 = (gcnew
System::Windows::Forms::RadioButton());
    this->radioButton2 = (gcnew
System::Windows::Forms::RadioButton());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->label3 = (gcnew System::Windows::Forms::Label());
    this->textBox2 = (gcnew System::Windows::Forms::TextBox());
    this->SuspendLayout();
    //
    // comboBox1
    //
    this->comboBox1->Font = (gcnew System::Drawing::Font(L"Comic Sans
MS", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
    this->comboBox1->FormattingEnabled = true;
    this->comboBox1->Location = System::Drawing::Point(99, 123);
    this->comboBox1->Name = L"comboBox1";
    this->comboBox1->Size = System::Drawing::Size(216, 34);
    this->comboBox1->TabIndex = 0;
    //
    // textBox1
    //
}

```

```

//  

>this->textBox1->Font = (gcnew System::Drawing::Font(L"Comic Sans  

MS", 14.25F, System::Drawing::FontStyle::Regular,  

System::Drawing::GraphicsUnit::Point,  

    static_cast<System::Byte>(204)));  

>this->textBox1->Location = System::Drawing::Point(99, 197);  

>this->textBox1->Multiline = true;  

>this->textBox1->Name = L"textBox1";  

>this->textBox1->Size = System::Drawing::Size(216, 39);  

>this->textBox1->TabIndex = 1;  

//  

// label1  

//  

>this->label1->AutoSize = true;  

>this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Tai  

Le", 14.25F, System::Drawing::FontStyle::Regular,  

System::Drawing::GraphicsUnit::Point,  

    static_cast<System::Byte>(0)));  

>this->label1->Location = System::Drawing::Point(99, 87);  

>this->label1->Name = L"label1";  

>this->label1->Size = System::Drawing::Size(138, 23);  

>this->label1->TabIndex = 2;  

>this->label1->Text = L"Оберіть картку";  

//  

// label2  

//  

>this->label2->AutoSize = true;  

>this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Tai  

Le", 14.25F, System::Drawing::FontStyle::Regular,  

System::Drawing::GraphicsUnit::Point,  

    static_cast<System::Byte>(0)));  

>this->label2->Location = System::Drawing::Point(99, 171);  

>this->label2->Name = L"label2";  

>this->label2->Size = System::Drawing::Size(52, 23);  

>this->label2->TabIndex = 3;  

>this->label2->Text = L"Сума";  

//  

// radioButton1  

//  

>this->radioButton1->AutoSize = true;  

>this->radioButton1->Font = (gcnew  

System::Drawing::Font(L"Microsoft Tai Le", 14.25F,  

System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  

    static_cast<System::Byte>(0)));  

>this->radioButton1->Location = System::Drawing::Point(99, 13);  

>this->radioButton1->Name = L"radioButton1";  

>this->radioButton1->Size = System::Drawing::Size(181, 27);  

>this->radioButton1->TabIndex = 4;  

>this->radioButton1->TabStop = true;  

>this->radioButton1->Text = L"Відкрити депозит";  

>this->radioButton1->UseVisualStyleBackColor = true;  

>this->radioButton1->CheckedChanged += gcnew  

System::EventHandler(this, &DepositForm::radioButton1_CheckedChanged);  

//  

// radioButton2  

//  

>this->radioButton2->AutoSize = true;  

>this->radioButton2->Font = (gcnew  

System::Drawing::Font(L"Microsoft Tai Le", 14.25F,  

System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  

    static_cast<System::Byte>(0)));  

>this->radioButton2->Location = System::Drawing::Point(99, 46);  

>this->radioButton2->Name = L"radioButton2";

```

```

this->radioButton2->Size = System::Drawing::Size(132, 27);
this->radioButton2->TabIndex = 5;
this->radioButton2->TabStop = true;
this->radioButton2->Text = L"Подивитися";
this->radioButton2->UseVisualStyleBackColor = true;
this->radioButton2->CheckedChanged += gcnew
System::EventHandler(this, &DepositForm::radioButton2_CheckedChanged);
//
// button1
//
this->button1->Font = (gcnew System::Drawing::Font(L"Candara",
14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->button1->Location = System::Drawing::Point(143, 330);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(124, 58);
this->button1->TabIndex = 6;
this->button1->Text = L"Виконати";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this,
&DepositForm::button1_Click);
//
// label3
//
this->label3->AutoSize = true;
this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Tai
Le", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label3->Location = System::Drawing::Point(99, 249);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(151, 23);
this->label3->TabIndex = 8;
this->label3->Text = L"Кількість місяців";
//
// textBox2
//
this->textBox2->Font = (gcnew System::Drawing::Font(L"Comic Sans
MS", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->textBox2->Location = System::Drawing::Point(99, 275);
this->textBox2->Multiline = true;
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(216, 39);
this->textBox2->TabIndex = 7;
//
// DepositForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::Font;
this->ClientSize = System::Drawing::Size(421, 400);
this->Controls->Add(this->label3);
this->Controls->Add(this->textBox2);
this->Controls->Add(this->button1);
this->Controls->Add(this->radioButton2);
this->Controls->Add(this->radioButton1);
this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Controls->Add(this->textBox1);
this->Controls->Add(this->comboBox1);

```

```

        this->Name = L"DepositForm";
        this->Text = L"DepositForm";
        this->Load += gcnew System::EventHandler(this,
&DepositForm::DepositForm_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }

#pragma endregion
    private: System::Void DepositForm_Load(System::Object^ sender,
System::EventArgs^ e)
{
    if (loginAccount->Cards != nullptr)
    {
        for (int i = 0; i < loginAccount->Cards->Count; ++i)
        {
            comboBox1->Items->Add(loginAccount->Cards[i]->CardNumber);
        }
    }
    private: System::Void radioButton2_CheckedChanged(System::Object^ sender,
System::EventArgs^ e)
{
    textBox1->Visible = !radioButton2->Checked;
    comboBox1->Visible = !radioButton2->Checked;
    textBox2->Visible = !radioButton2->Checked;

    label1->Visible = !radioButton2->Checked;
    label2->Visible = !radioButton2->Checked;
    label3->Visible = !radioButton2->Checked;

    if (radioButton1->Checked)
    {
        label1->Text = "Карта:";
        label2->Text = "Сума:";
        label3->Text = "Кількість місяців:";
    }
}

private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    try {
        if (radioButton1->Checked)
        {
            int index = comboBox1->SelectedIndex;
            double sum = Convert::.ToDouble(textBox1->Text);
            Deposit^ oldDepo = depRepo->GetDepositByLogin(loginAccount-
>Login);

            if (oldDepo == nullptr) {
                if (sum <= loginAccount->Cards[index]->Balance)
                {
                    Deposit^ depo = gcnew Deposit();
                    depo->Login = loginAccount->Login;
                    depo->Month = Convert::ToInt32(textBox2->Text);
                    depo->Sum = sum;
                    depo->CalculateInterestRate();
                    loginAccount->Cards[index]->Balance -= sum;
                    depRepo->Create(depo);
                    MessageBox::Show("Депозит відкрито!");
                    this->Close();
                }
            }
        }
    }
}

```

```

        else
        {
            MessageBox::Show("Недостатньо коштів!", "Увага!",
MessageBoxButtons::OK, MessageBoxIcon::Error);
        }
    else
    {
        MessageBox::Show("У вас вже є відкритий депозит",
"Увага!", MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
else if (radioButton2->Checked)
{
    Deposit^ oldDepo = depRepo->GetDepositByLogin(loginAccount-
>Login);
    if (oldDepo != nullptr)
    {
        label1->Visible = radioButton2->Checked;
        label2->Visible = radioButton2->Checked;
        label3->Visible = radioButton2->Checked;

        label1->Text = "Вкладена сума: " + oldDepo-
>Sum.ToString() + " грн";
        label2->Text = "Кількість місяців: " + oldDepo->Month;
        label3->Text = "Процентна ставка: " + oldDepo-
>InterestRate.ToString()+" %";
    }
    else
    {
        MessageBox::Show("У вас немає відкритого депозиту",
"Увага!", MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
}

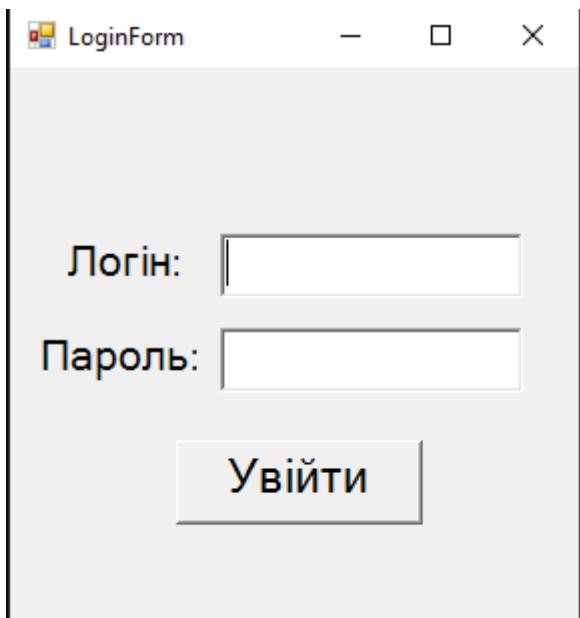
catch(...)
{
}
}
private: System::Void radioButton1_CheckedChanged(System::Object^ sender,
System::EventArgs^ e)
{
}
};

}
}

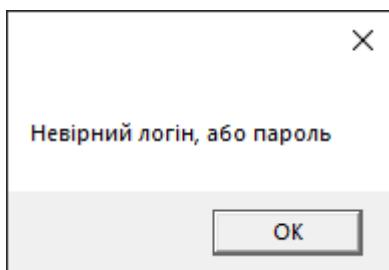
```

Результати розробки програми

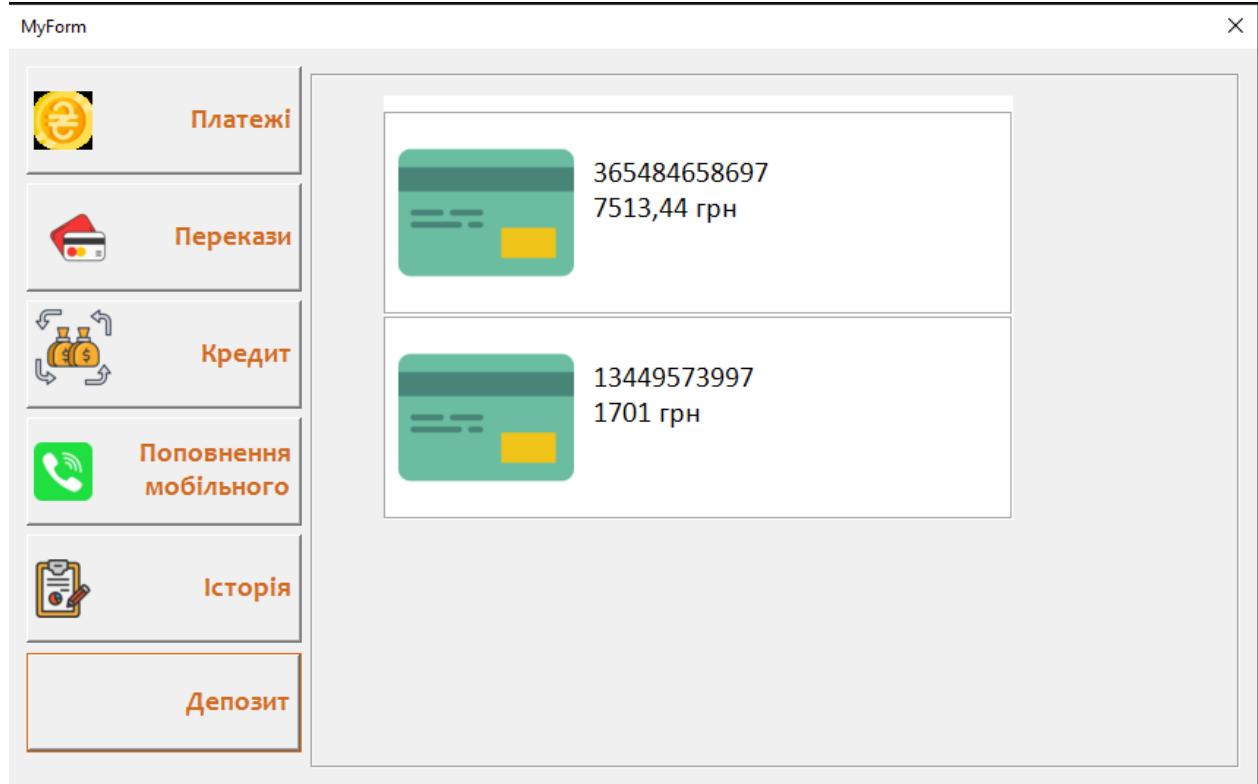
1. Вікно авторизації



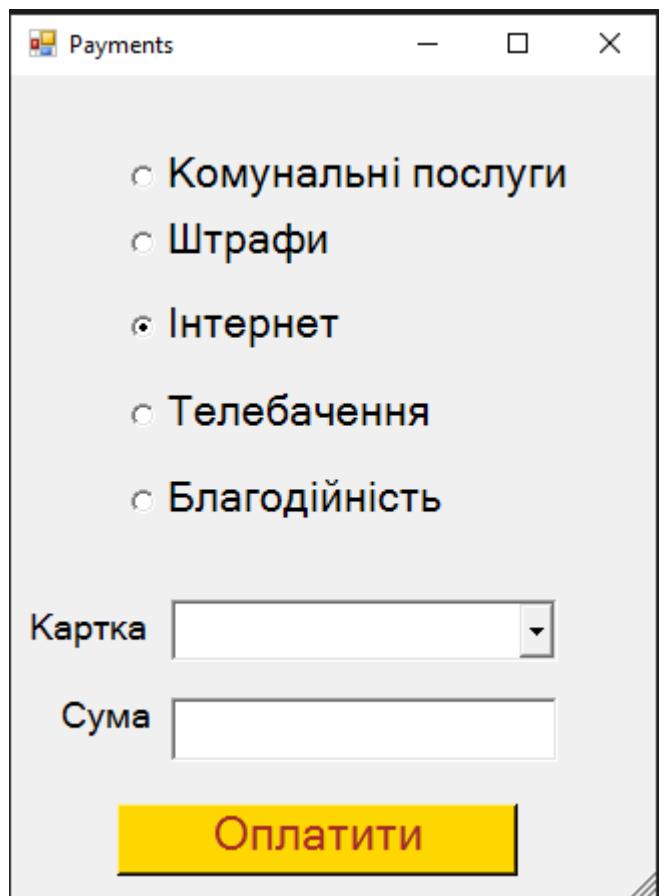
2. Повідомлення про неправильні дані



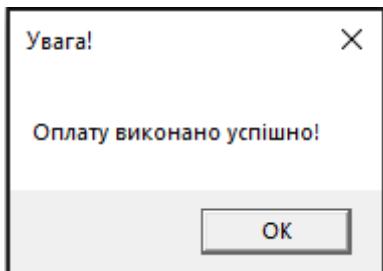
3. Головне меню програми



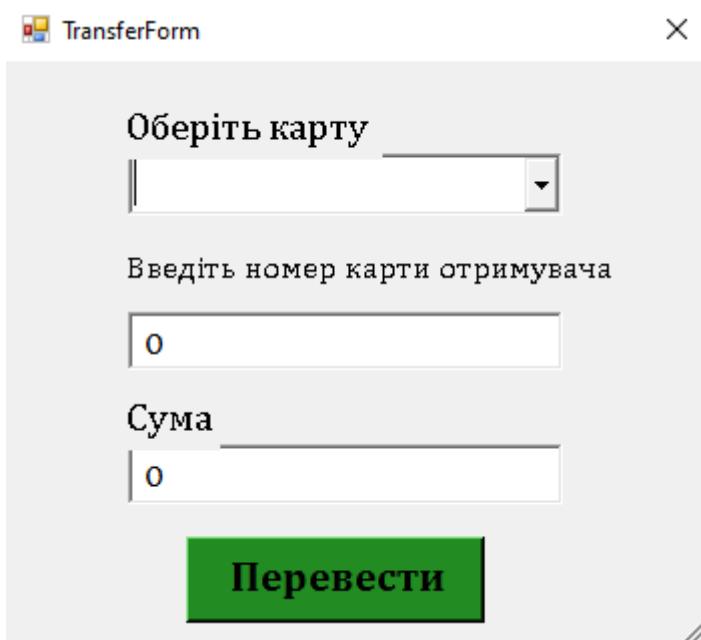
4. Вікно для виконання соціальних платежів



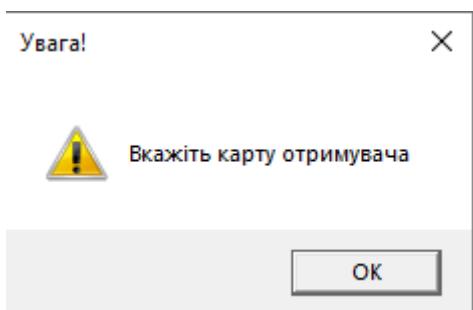
5. Повідомлення про успішну оплату соц. платежу



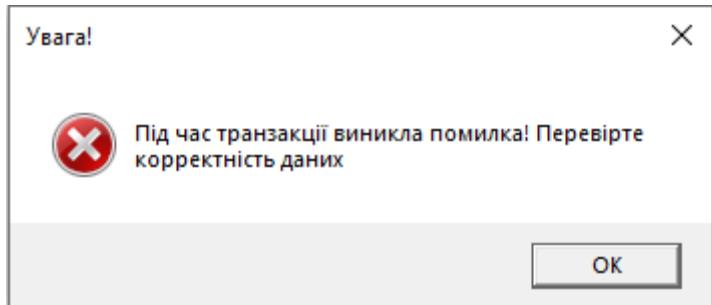
6. Вікно для переказу на іншу картку



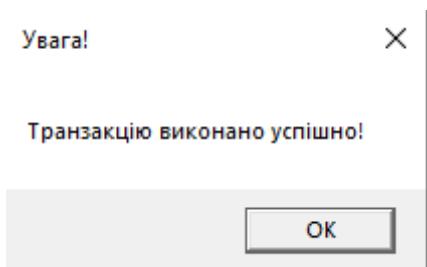
7. Нагадування ввести номер картки



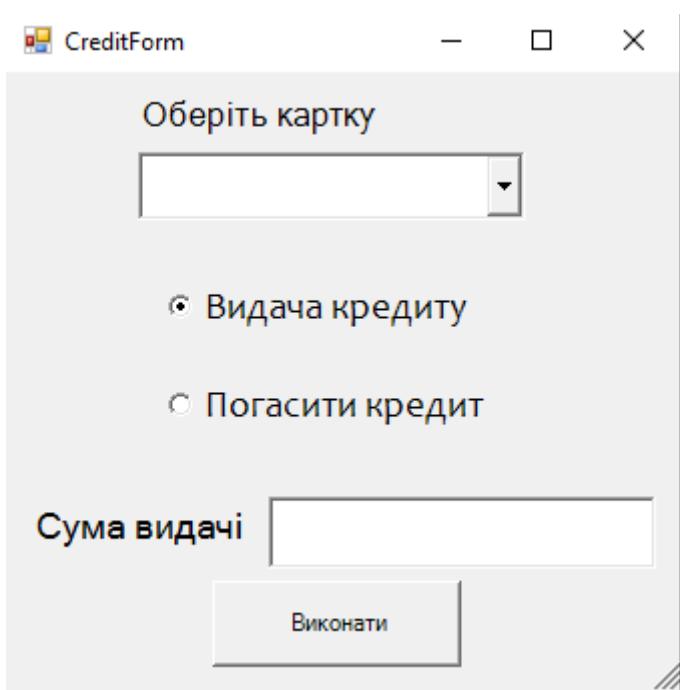
8. Повідомлення про неправильно введену картку



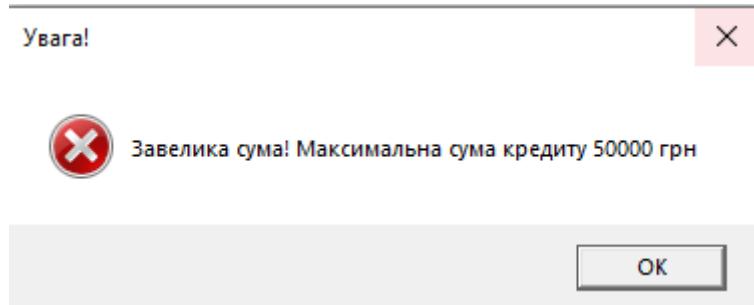
9. Повідомлення про успішний переказ



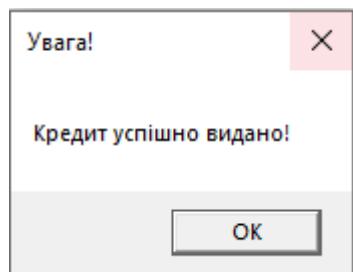
10. Вікно для видачі або погашення кредиту



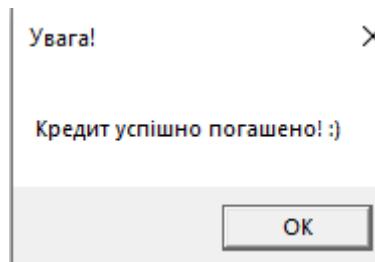
11. Повідомлення про обмеження кредиту



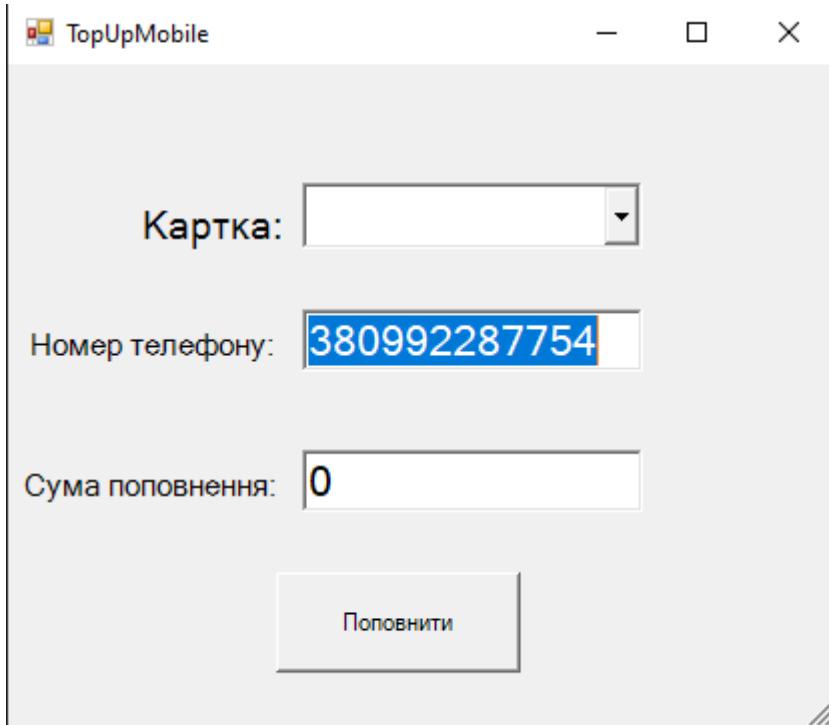
12. Повідомлення про успішну видачу кредиту



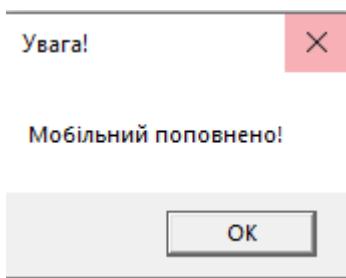
13. Повідомлення про успішне погашення кредиту



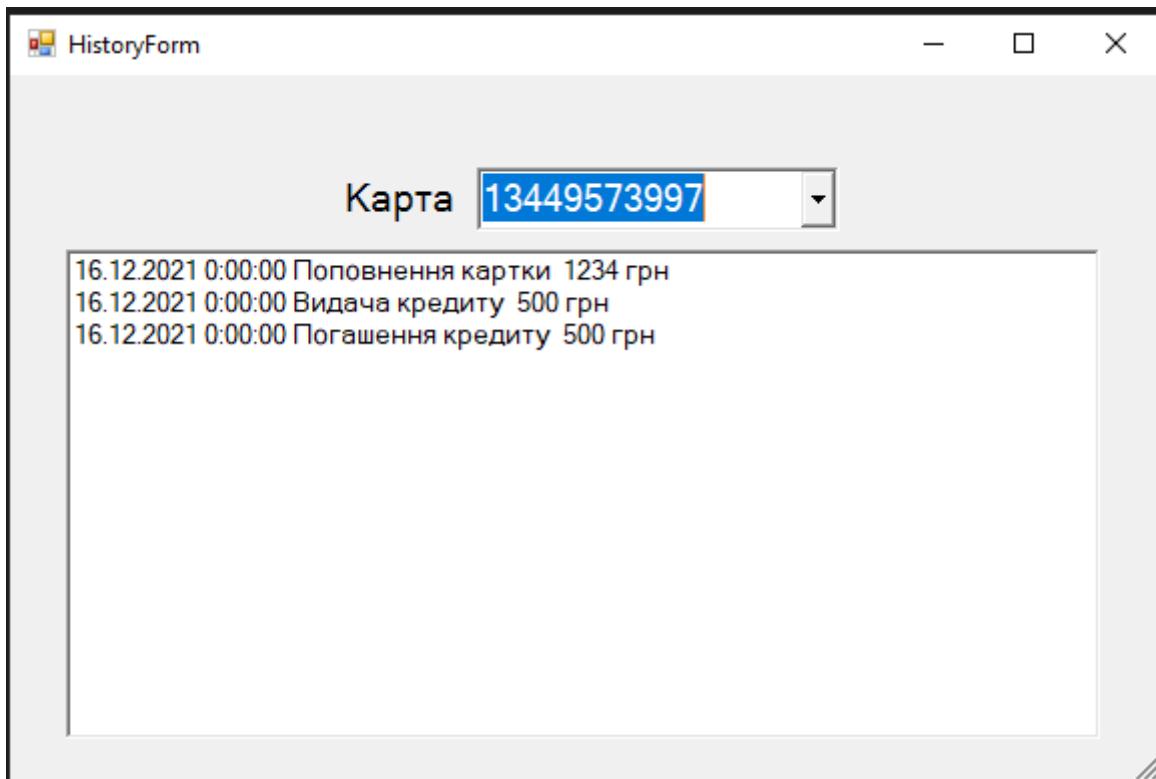
15. Вікно для поповнення мобільного телефону



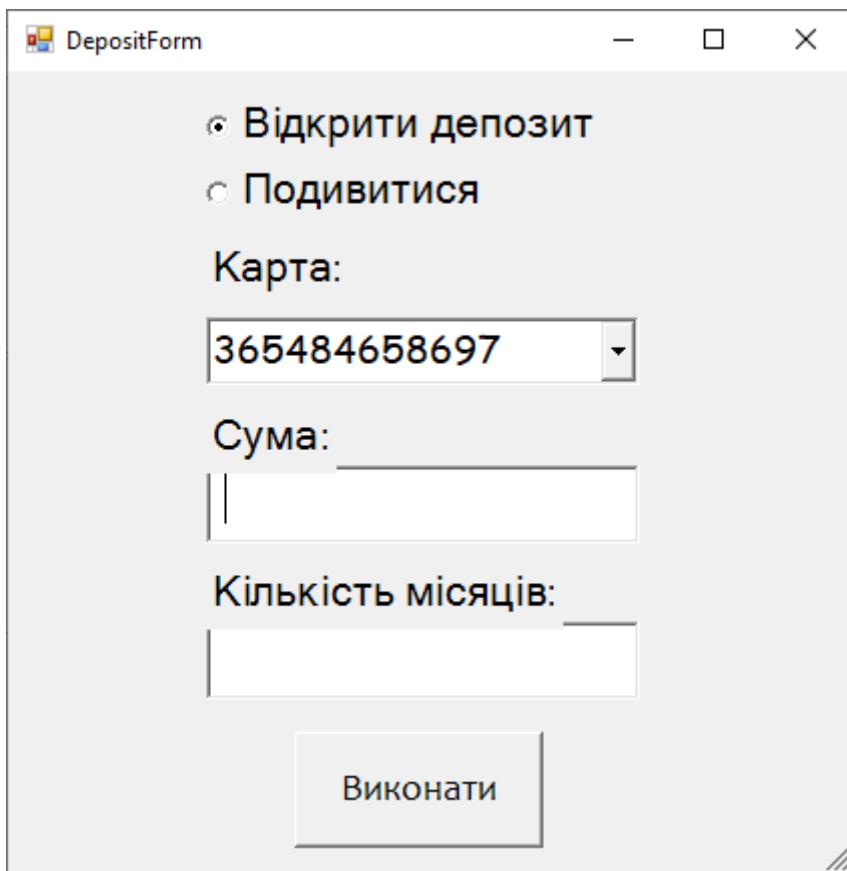
16. Повідомлення про успішне поповнення мобільного



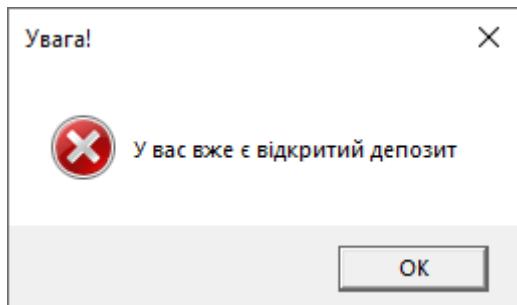
17. Вікно з історією транзакцій за картками



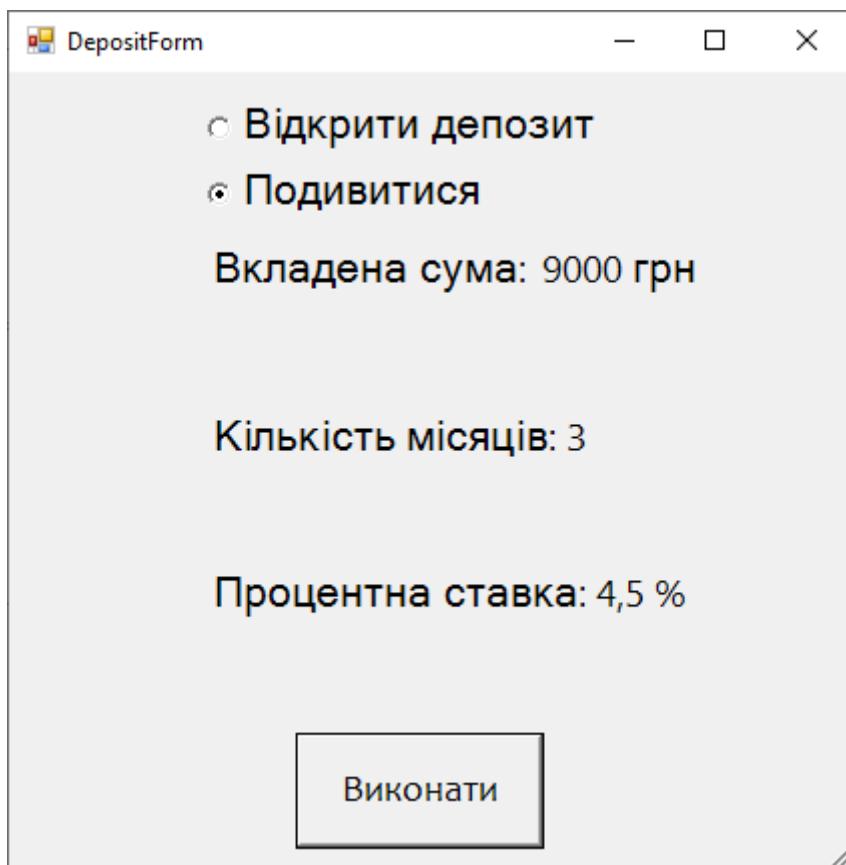
18. Вікно депозиту



19. Повідомлення про вже відкритий депозит



20. Інформація про вже відкритий депозит



Програмна документація

1. Файли json та інші ресурси

- accounts.json - файл для зберігання усіх аккаунтів
- histories.json - файл для зберігання усіх історій
- deposits.json - файл для зберігання усіх депозитів
- BankApp/icons - папка в якій знаходяться усі іконки для кнопок меню

2. Використовувані методи та класи

System - простір імен для використання типів даних .NET

System::ComponentModel::ComponentResourceManager - клас для перерахування ресурсів компонента

System::Collections::Generic::List<T> - клас для роботи зі списками

Newtonsoft::Json::JsonConvert - клас для роботи з JSON форматом

System::IO::File - клас для роботи з файлами

System::Windows::Forms::Form - клас для створення форми;

System::Windows::Forms::Button - клас для створення кнопки

System::Windows::Forms::GroupBox - клас для створення елемента групування

System::Windows::Forms::PictureBox - клас для створення картинки

System::Windows::Forms::ComboBox - клас для випадаючого списку

System::Windows::Forms::Label - клас для створення етикетки

System::Windows::Forms::Control - базовий клас для створення елементів

System::Windows::Forms::RadioButton - клас для перемикачів

System::Windows::Forms::TextBox - клас для текстового поля

System::Drawing::Point - клас для роботи з точкою(x;y);

System::Drawing::Color - структура для роботи з кольорами

System::Drawing::Bitmap - клас для використання точкового зображення

System::Drawing::Image - абстрактний клас для роботи із зображеннями

Add(T entity) - метод для додавання елементу у список

Remove(T entity)- метод для видалення елементу зі списку

JsonConvert::SerializeObject(Object^value) - метод для серіалізації даних у JSON

JsonConvert::DeserializeObject<T>(String^json)- метод для десеріалізації даних з JSON

File::WriteAllText(String^text)- метод для запису тексту у файл

InitializeComponent(void) - метод для ініціалізації компонентів форми

Form_Load() - метод, що викликається перед відображенням форми

button_Click() - метод, що викликається після натискання кнопки

3. Класи предметної області:

1)Card:

Поля:

- String^ cardNumber - номер карти
- int cvv - код безпеки картки
- int securCode - пін-код картки
- double balance - грошовий баланс на карті

Методи:

- GroupBox^ CreateCardPanel(Point p) - метод створення іконки для відображення

2)Інформації про картку на головній формі

Властивості:

- CardNumber
- CvV
- SecurCode
- Balance

3) History:

Поля:

- bool isCredit - якщо історія- це взяття кредиту, то значення true інакше false
- String[^] cardNumber - номер картки
- String[^] operation - назва операції, яка виконувалася
- DateTime[^] date - дата виконання операції
- double sum- сума яка при цьому використовувалася

Властивості:

- IsCredit
- CardNumver
- Operation
- Date
- IsCredit
- Sum

4) Account

Поля:

- String[^] login
- String[^] password
- String[^] numberPhone
- List<Card^{^>}>[^]cards

Властивості:

- Login
- Password
- NumberPhone
- Cards

5)Deposit

Поля:

- String^ login - логін користувача, який відкрив депозит
- DateTime today - сьогоднішня дата(дата відкриття депозиту)
- int month - кількість місяців
- double sum - внесена сума
- double interestRate - процентна ставка

4.Репозиторії

1)AccountRepository

Поля:

- List<Account^>^accounts - список усіх аккаунтів, що є у файлі accounts.json

Методи:

- Create(Account^entity) - метод створення нового аккаунту
- Delete(Account^entity) - метод видалення аккаунту
- List<Account^>^ GetAll() - метод, що завантажує усіх користувачів з файлу accounts.json
- Save() - метод збереження усіх користувачів у файл
- Account^ Login(String^login, String^password) - метод, що шукає користувача з таким логіном та паролем
- Card^ FindCard(String^number) - метод пошуку карти за номером
- bool TransferMoney(String^card1, String^card2, double sum) - метод переведення суми з однієї карти на іншу

2)DepositRepository

Поля:

- List<Deposit^>^deposits - список усіх депозитів, що є у файлі deposits.json

Методи:

- Create(Deposit^entity) - метод створення

- List<Deposit^>^ GetAll() - метод, що завантажує усі депозити з файлу deposits.json
- Save() - метод збереження у файл
- Deposit^ GetDepositByLogin(String^ login) - метод, що знаходить за логіном користувача його депозит

3) HistoryRepository

Поля:

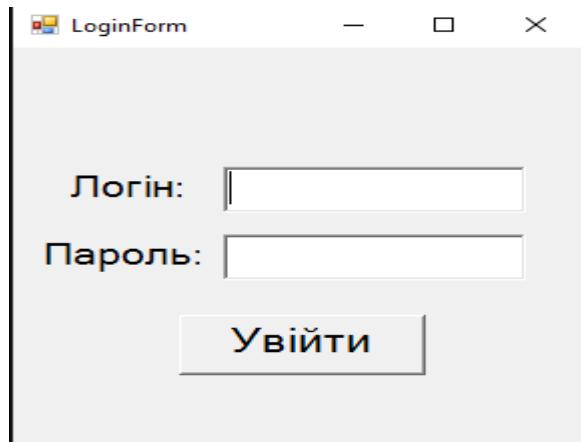
- List<History^>^ histories - список усіх історій, що є у файлі histories.json

Методи:

- Create(Account^ entity) - метод створення
- List<History^>^ GetAll() - метод, що завантажує усі депозити з файлу histories.json
- Save() - метод збереження у файл
- List<History^>^ GetHistoriesByCardNumber(String^ number) - метод, що повертає історії операцій над картою з таким-то номером

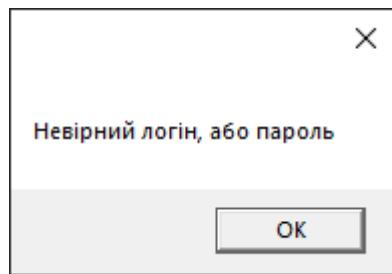
Керівництво користувача

1.На початку роботи з програмою вас зустрічає вікно авторизації (мал.1) у якому ви повинні у полі “Login” ввести свій логін, а у полі “Password” ввести пароль після цього натиснути кнопку “Увійти”.



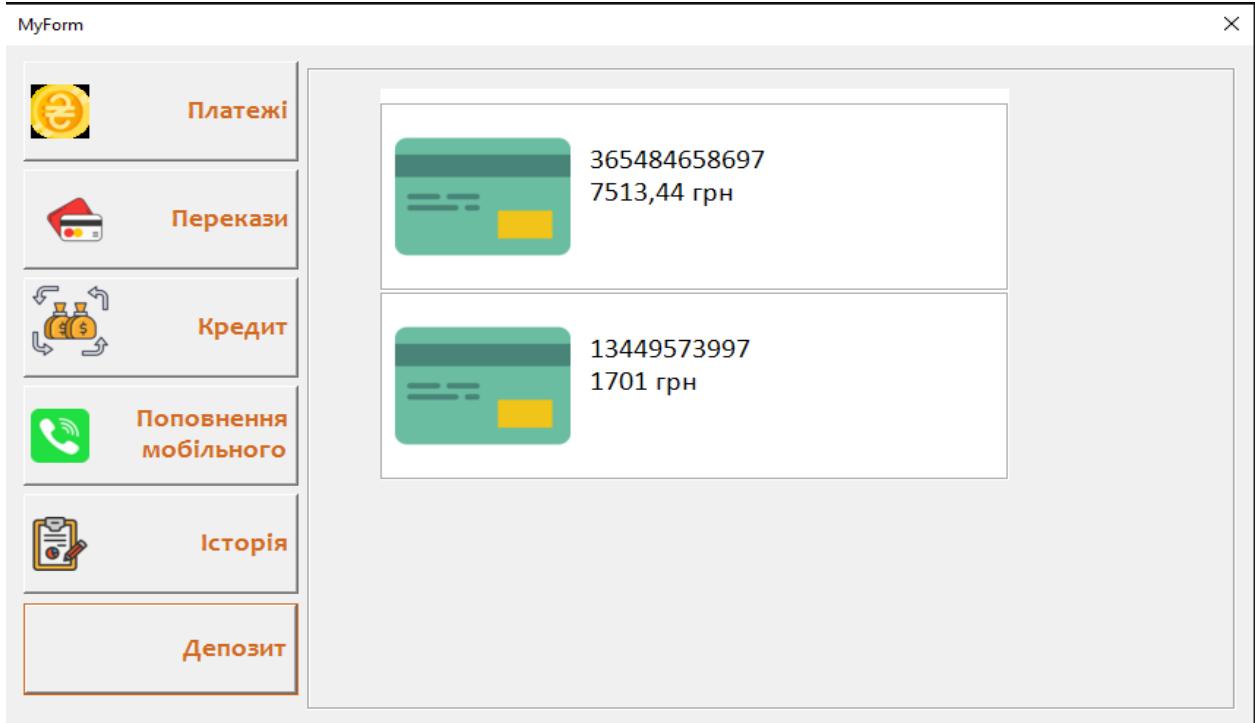
мал.1

У випадку, якщо вам видало помилку (мал.2) перевірте, будь ласка, правильність введених даних .



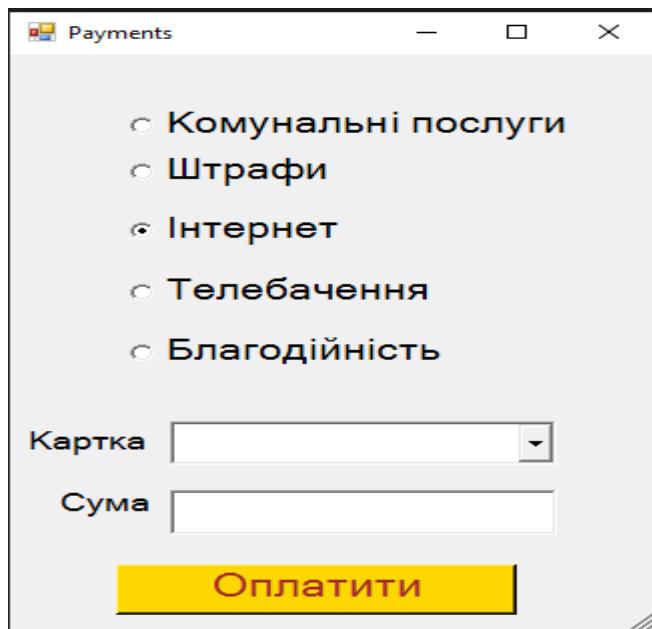
мал.2

2. Після входу у свій особистий кабінет, відкривається головне меню програми (мал.3) у якому можна вибрати потрібну вам послугу та подивитися баланс своїх карток.



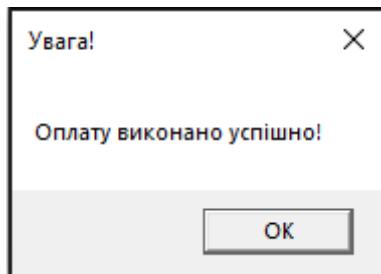
мал.3

3. У вкладці “Платежі” (мал.4) знаходяться різні соціальні послуги, щоб ними скористатися виберіть послугу, картку, введіть суму та натисніть кнопку “Оплатити”.



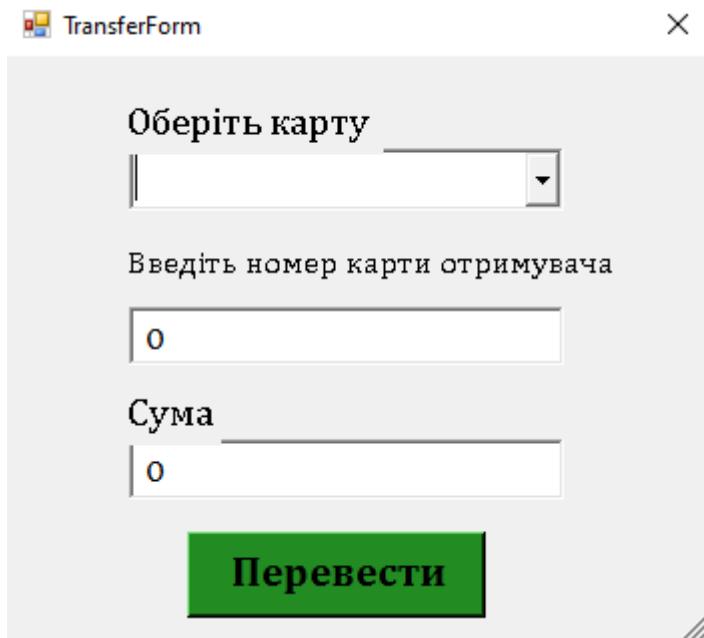
Мал.4

Після виконання операції з'явиться віконечко про успішну оплату (мал.5).



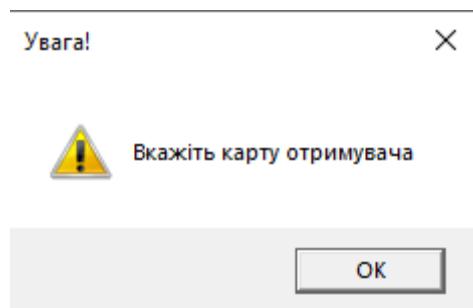
Мал.5

4. У вкладці “Перекази” (мал.6) можна виконати переказ на іншу картку, для цього потрібно обрати картку з якої спишуться кошти, ввести номер картки отримувача, суму та натиснути кнопку “Перевести”



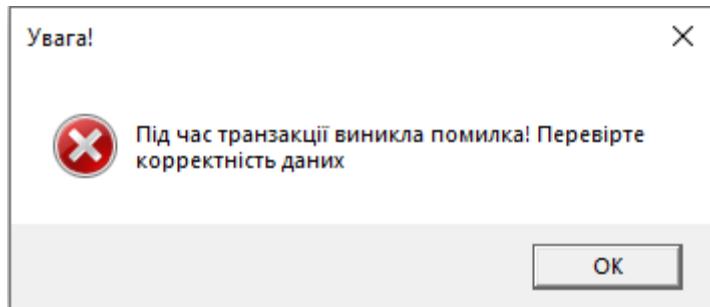
Мал.6

Якщо ви забули ввести номер карти отримувача у вас з'явиться повідомлення (мал.7).



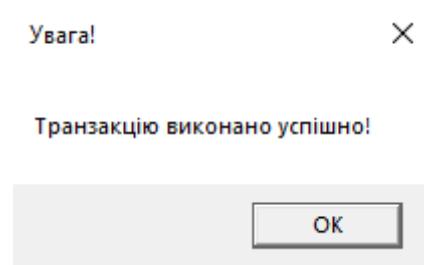
Мал.7

У випадку невірно введених даних картки отримувача з'явиться віконечко (мал.8), натисніть “ОК” та перевірте будь ласка правильність введених даних.



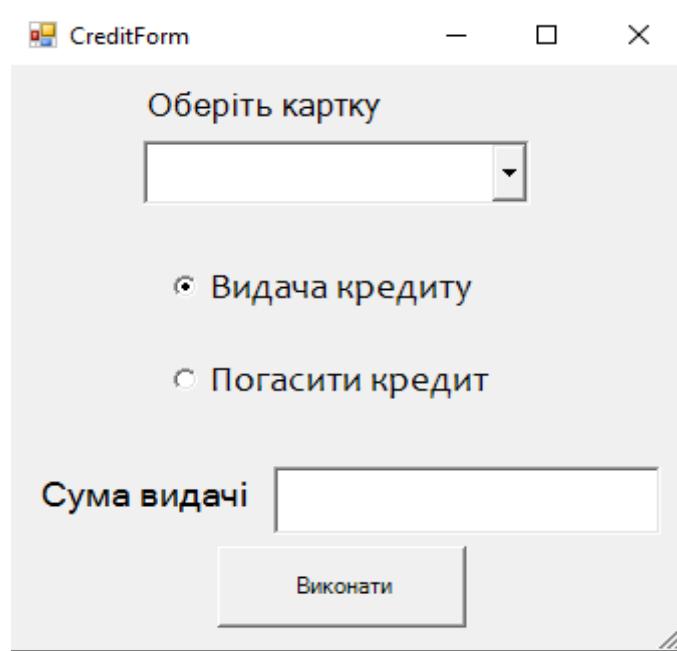
Мал.8

Після успішного виконання переказу з'явиться повідомлення (мал.9).



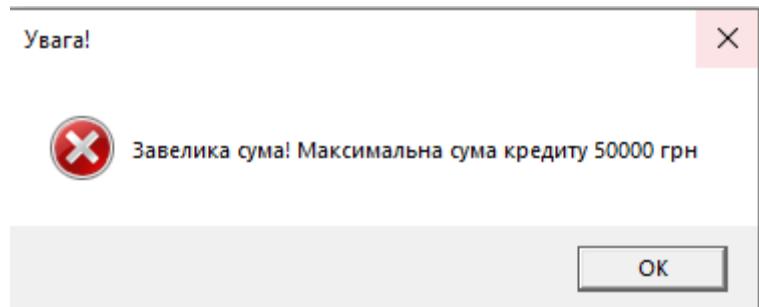
Мал.9

5. У вкладці “Кредит” (мал.10) ви можете отримати або погасити кредит. Для отримання кредиту оберіть картку, ведіть суму та натисніть кнопку “Виконати”. Для погашення виберіть кнопку “Погасити кредит” та натисніть “Виконати”.



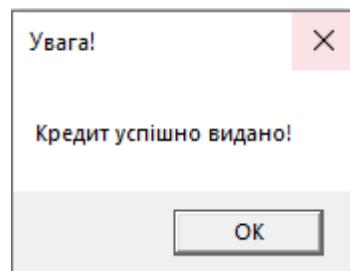
Мал.10

Обмеження на суму видачі кредиту 50000 грн., якщо сума перевищує це значення з'явиться помилка (мал.11).



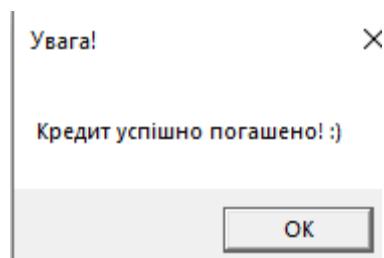
Мал.11

У випадку, якщо всі дані введено правильно з'явиться віконечко про успішну видачу кредиту(мал.12)



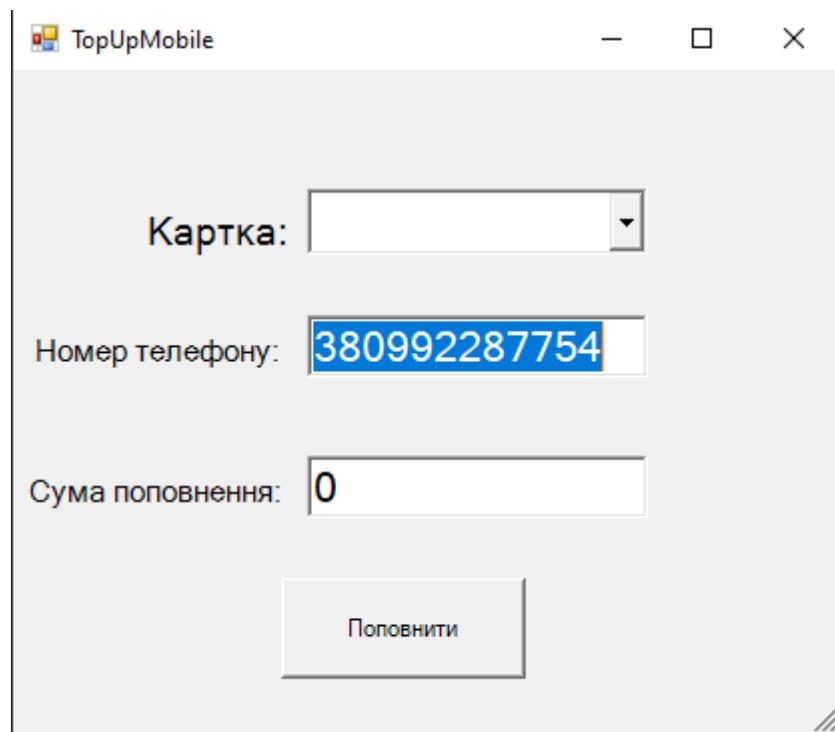
Мал.12

Після погашення кредиту з'явиться повідомлення про успішне погашення кредиту операцію (мал.13).



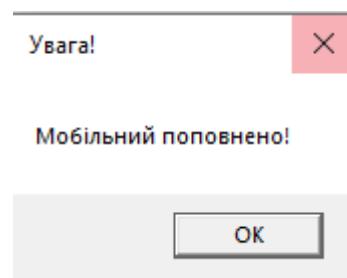
Мал.13

6. У вкладці “Поповнення мобільного” (мал.14) можна виконати операцію поповнення телефону, для цього потрібно обрати картку, ввести номер телефону, суму поповнення та натиснути на “Поповнити”



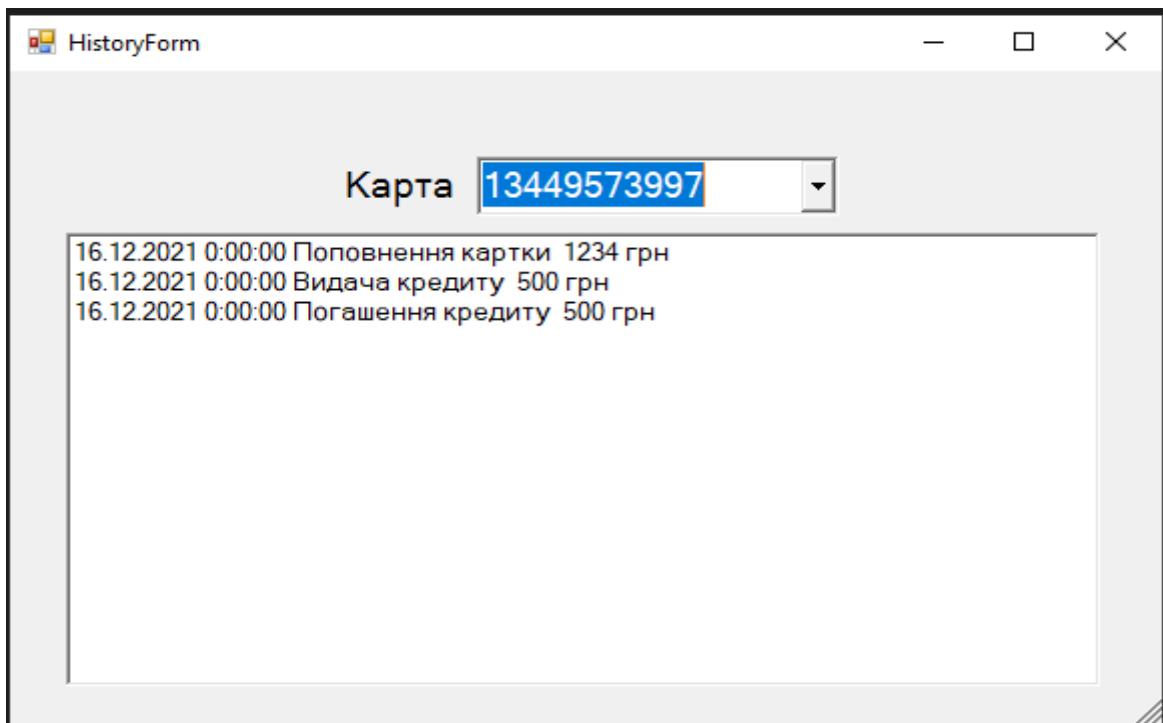
Мал.14

Після успішного поповнення з'явиться повідомлення (мал.15).



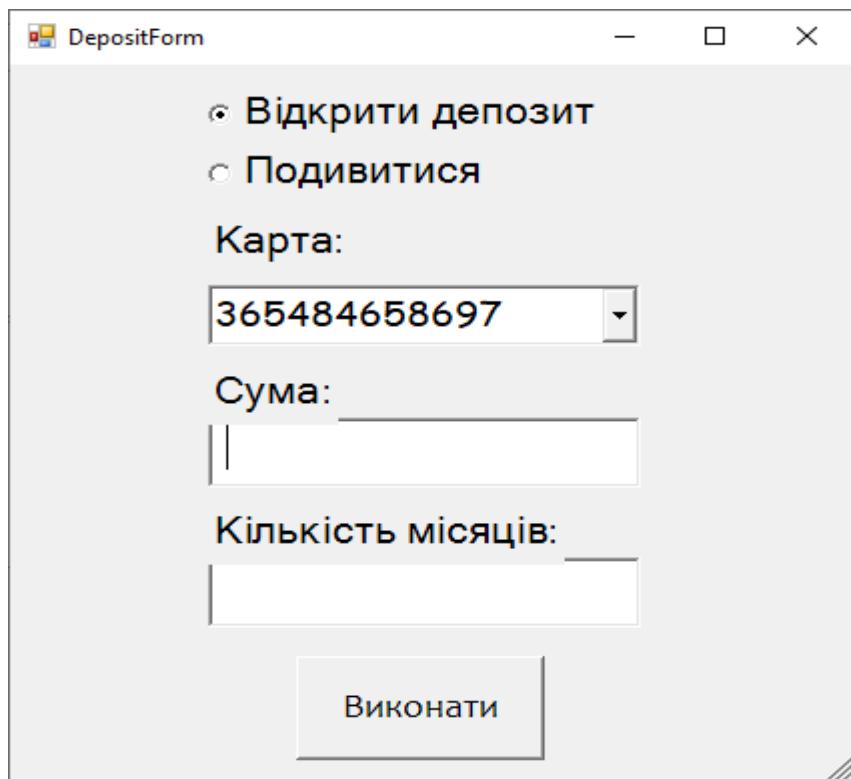
Мал.15

7. У вкладці “Історія” міститься інформація про останні операції з картками (мал.16).



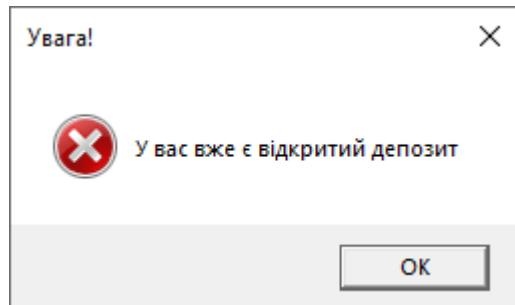
Мал.16

8. У вкладці “Депозит” (мал.17), можна відкрити депозит для цього оберіть картку, суму, кількість місяців та натисніть “Виконати”.



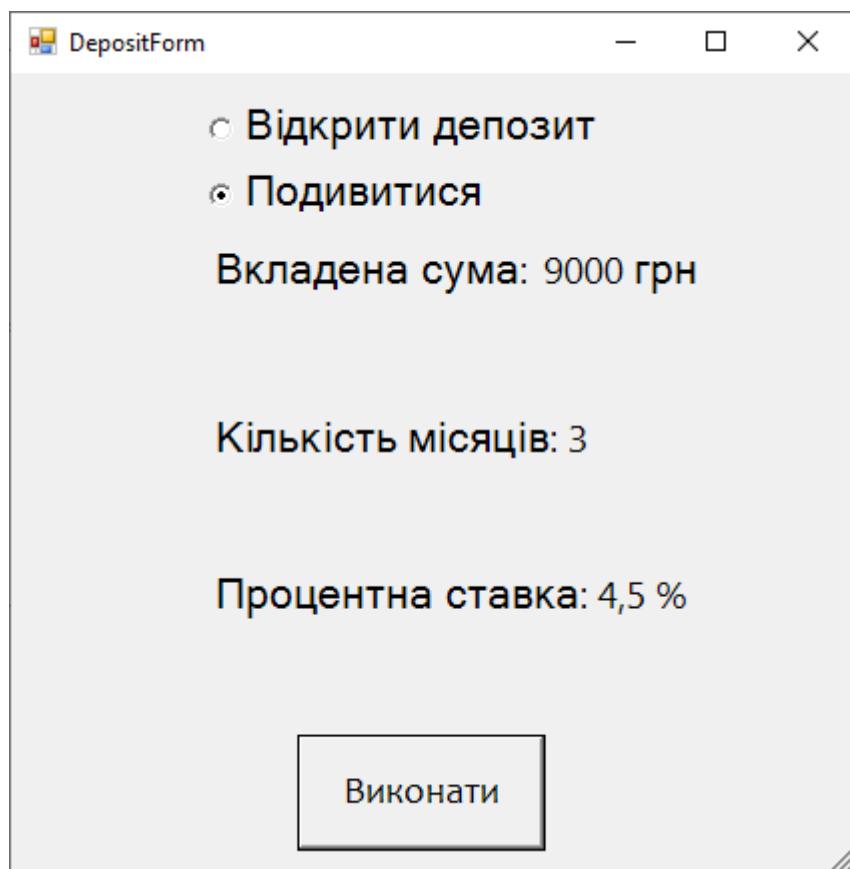
(Мал.17)

Якщо з'явилася помилка (мал.18) - це означає, що у вас уже є відкритий депозит.



Мал.18

Для перегляду вже відкритого депозиту виберіть функцію “Подивитися” після цього з'явиться вікно з інформацією (мал.19).



Мал.19

Результати тестувань

Тестування не було проведено у зв'язку з відсутністю тестувальника.

Висновок

У процесі виконання курсової роботи команда “4 Мушкетера inc.” створила автоматизовану систему клієнта банку. На початку роботи було проведено інтерв'ювання з замовником. Після цього Архітектор приступив до створення архітектури. Використавши архітектуру проекту програміст написав програмний код. Потім після перевірки ПЗ, документатор все скомпонував та написав поясннювальну записку.