

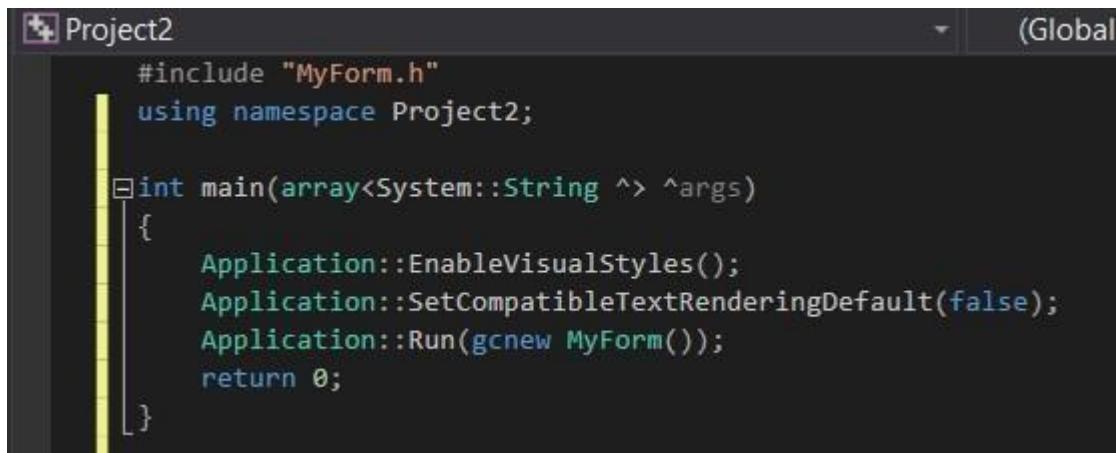
Тема. Створення настільного додатку на основі технології Microsoft .NET з використанням класів на мові C++

Мета. Ознайомитись із можливостями створення користувачьких класів у Visual Studio та їх використання у Windows Form додатках.

Хід роботи

Приклад

Завдання. Створити настільний додаток на основі технології Microsoft .NET з використанням мови CLR (Common Language Runtime) для розв'язку наступного завдання: визначити клас банківський рахунок, який повинен містити ім'я вкладника, адреса його проживання та телефон, номер рахунку, сума вкладу, дата вкладу, термін вкладу, відсоток, загальна сума. Реалізувати методи введення даних про вкладників, зарахування та зняття коштів з банківського рахунку.



```
#include "MyForm.h"
using namespace Project2;

int main(array<System::String ^> ^args)
{
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew MyForm());
    return 0;
}
```

Рисунок 2.1 – Код файлу MyForm.cpp

1 Створити клас для розв'язку завдання. Оголосити масив об'єктів класу у файлі заголовків (bank.h). Реалізувати у класі методи введення даних та пошуку (див Лістинг 2.1).

Лістинг2.1–Код файлу blank.h

```
#pragma once
#include "MyForm.h"
#include <stdlib.h>
```

```

#include <cstring>
#include <cstdio> #define n 10 class Bank {
private:
    char* Name;    char* Adress;    char* AccountNumber;
    double Suma;   char* Date;     int Term;      int Rate; public:
    void InsertDepositors(char* s, char* a, char* ac, double su, char*
d, int t, int r);    void RemoveMoney(char* s, char* ac, double m);
void SearchName(char* s);        void SearchAccount(char* a);    void
SearchNameNumber(char* s, char* a);
    void OutputFile(); }mas[n];
    void Bank::InsertDepositors(char* s, char* a, char* ac, double su,
char* d, int t,
    int r)
{
    Name = s;
    Adress = a;
    AccountNumber = ac;
    Suma = su;
    Date = d;
    Term = t;
    Rate = r; }
    void Bank::RemoveMoney(char* s, char* ac, double m) { if
(strcmp(Name, s) == 0 && strcmp(AccountNumber, ac) == 0)
{
    Suma -= m;
}
}
    void Bank::SearchName(char* s)
    { if (strcmp(Name, s)
== 0)
{
    FILE* f;
    f = fopen("search_name.txt", "wt");
    fprintf(f, "%s\t%s\t%s\t%.2f\t%s\t%i\t%i\n", AccountNumber, Name,
Adress, Suma, Date, Term, Rate);
    fclose(f);
}
}
    void Bank::SearchAccount(char* s)
    { if (strcmp(AccountNumber, s)
== 0)
{
    FILE* f;
    f = fopen("search_account.txt", "wt");
    fprintf(f, "%s\t%s\t%s\t%.2f\t%s\t%i\t%i\n", AccountNumber, Name,
Adress, Suma, Date, Term, Rate);
    fclose(f);
}
}
}

```

```

void Bank::SearchNameNumber(char* s, char* a)
{
    if (strcmp(AccountNumber, a) == 0 && strcmp(Name, s) ==
0)
    {
        FILE* f;
        f = fopen("search_name_account.txt", "wt");
        fprintf(f, "%s\t%s\t%s\t%.2f\t%s\t%i\t%i\n", AccountNumber, Name,
Adress, Suma, Date, Term, Rate);
        fclose(f);
    }
}
void Bank::OutputFile()
{
FILE* f = fopen("depositors.txt", "a");
fprintf(f, "%s\t%s\t%s\t%.2f\t%s\t%i\t%i\n", AccountNumber, Name, Adress,
Suma, Date, Term, Rate);    fclose(f);  }

```

2 Спроектував головну форму MyForm.h, додав на неї компоненти, написав код для функціонування програми (див Листинг 2.2 та Рис. 2.4).

Листинг 2.2 – Код файлу Myform.h

```

private:     System::Void      MyForm_Load(System::Object^      sender,
System::EventArgs^ e) {
}
private:     System::Void      label3_Click(System::Object^      sender,
System::EventArgs^ e) {
}
private:     System::Void
dataGridView2_CellContentClick(System::Object^      sender,
System::Windows::Forms::DataGridViewCellEventArgs^      e)
{
    tabControl1->Appearance =
    System::Windows::Forms::TabAppearance::FlatButtons;
    tabControl1->SizeMode
    tabControl1->TabStop = false;
}
private:     System::Void
dataGridView1_CellContentClick(System::Object^      sender,
System::Windows::Forms::DataGridViewCellEventArgs^      e)
{
    tabControl1->Appearance =
    System::Windows::Forms::TabAppearance::FlatButtons;
    tabControl1->SizeMode
    tabControl1->TabStop = false;
}

```

```

    private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) { this->Height = 400; tabControl1->SelectedTab
= tabPage2; }
    private: System::Void button6_Click(System::Object^ sender,
System::EventArgs^ e) {
        tabControl1->SelectedTab = tabPage1;
    }
    private: System::Void button7_Click(System::Object^ sender,
System::EventArgs^ e) {
        char* str1 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox1-
>Text);                                char* str2 = =
(char*)(void*)Marshal::StringToHGlobalAnsi(textBox2-
>Text);                                char* str3 = =
(char*)(void*)Marshal::StringToHGlobalAnsi(textBox3-
>Text);

        double str4;
        if (!Double::TryParse(textBox4->Text, str4)) {
            MessageBox::Show("Некоректне значення в полі для
числа з плаваючою точкою (textBox4).");
            return;
        }

        char* str5 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox5-
>Text);

        int str6;
        if (!Int32::TryParse(textBox6->Text, str6)) {
            MessageBox::Show("Некоректне значення в полі для цілого числа
(textBox6).");
            return;
        }

        int str7;
        if (!Int32::TryParse(comboBox1->Text, str7)) {
            MessageBox::Show("Некоректне значення в полі для цілого числа
(comboBox1).");
            return;
        }

        // Вставка в масив           mas[qountity].InsertDepositors(str1, str2,
str3, str4,
str5, str6, str7);
        // Запис до файлу          mas[qountity].OutputFile();
qountity++;

private: System::Void button9_Click(System::Object^ sender,

```

```
System::EventArgs^ e) {           char*             str1           =
(char*)(void*)Marshal::StringToHGlobalAnsi(textBox7-
>Text);
    char* str2 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox8-
>Text);   double str3 = Convert::.ToDouble(textBox9->Text);
    for (int i = 0; i < qountity; i++) {
        mas[i].RemoveMoney(str1, str2, str3);
    }

    FILE* f;
    f = fopen("depositors.txt", "w");
fclose(f);

    for (int i = 0; i < qountity; i++) {
        mas[i].OutputFile();
    }

}

private: System::Void button8_Click(System::Object^ sender,
System::EventArgs^ e) {
    tabControl1->SelectedTab = tabPage1;
}

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {      this->Height = 400;
    tabControl1->SelectedTab = tabPage3;
}

private: System::Void button4_Click(System::Object^ sender,
System::EventArgs^ e) {      this->Height = 400;
    tabControl1->SelectedTab = tabPage4;
}

private: System::Void button10_Click(System::Object^ sender,
System::EventArgs^ e) {
    tabControl1->SelectedTab = tabPage1;
}

private: System::Void button11_Click(System::Object^ sender,
System::EventArgs^ e) {           char*             str1           =
(char*)(void*)Marshal::StringToHGlobalAnsi(textBox10-
>Text);
    char* str2 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox11-
>Text);
    if (checkBox1->Checked) {           for (int
i = 0; i < qountity; i++) {
        mas[i].SearchName(str1);
    }

    dataGridView2->Rows->Clear();
    array<String^>^ lines = File::ReadAllLines("search_name.txt",
System::Text::Encoding::GetEncoding(1251));
}
```

```
        for each (String ^ str in lines) {
            DataGridViewRow^ row = gcnew DataGridViewRow();
            row->CreateCells(dataGridView2);
            for (int i = 0; i < str->Split('\t')->Length; i++) {
                row->Cells[i]->Value = str->Split('\t')[i];
            }
            dataGridView2->Rows->Add(row);
        }
    }

        if (checkBox2->Checked) {                                for (int i =
0; i < qountity; i++) {
            mas[i].SearchAccount(str2);
        }

        dataGridView2->Rows->Clear();
array<String^>^ lines =
File::ReadAllLines("search_account.txt",
System::Text::Encoding::GetEncoding(1251));

        for each (String ^ str in lines)
{
            DataGridViewRow^ row = gcnew DataGridViewRow();
            row->CreateCells(dataGridView2);
            for (int i = 0; i < str->Split('\t')->Length; i++) {
                row->Cells[i]->Value = str->Split('\t')[i];
            }
            dataGridView2->Rows->Add(row);
}
}
if (checkBox1->Checked && checkBox2->Checked) {

for (int i = 0; i < qountity; i++) {
    mas[i].SearchAccount(str2);
}

        dataGridView2->Rows->Clear();
array<String^>^ lines =
File::ReadAllLines("search_name_account.txt",
System::Text::Encoding::GetEncoding(1251));

        for each (String ^ str in lines) {
            DataGridViewRow^ row = gcnew DataGridViewRow();
            row->CreateCells(dataGridView2);
            for (int i = 0; i < str->Split('\t')->Length; i++) {
                row->Cells[i]->Value = str->Split('\t')[i];
            }
        }
}
```

```

        dataGridView2->Rows->Add(row);
    }
}

private: System::Void button5_Click(System::Object^ sender,
System::EventArgs^ e) {
    MyForm::Close();
}

private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->Height = 400;
    dataGridView1->Rows->Clear();
    // Читання даних з файлу depositors.txt    array<String^>^ lines
    = File::ReadAllLines("depositors.txt",
    System::Text::Encoding::GetEncoding(1251));
    for each (String ^ str in lines)
    {
        DataGridViewRow^ row = gcnew DataGridViewRow();
        row->CreateCells(dataGridView1);           for (int i = 0; i < str->Split('\t')-
>Length; i++)
            row->Cells[i]->Value = str->Split('\t')[i];
        dataGridView1->Rows->Add(row);
    }

    // Очищення файлу depositors.txt після завантаження даних
    FILE* f = fopen("depositors.txt", "w"); // Відкриття файлу в режимі
запису для очищення      if (f != nullptr) {
        fclose(f); // Закриваємо файл, щоб він був порожнім
    }
}

```

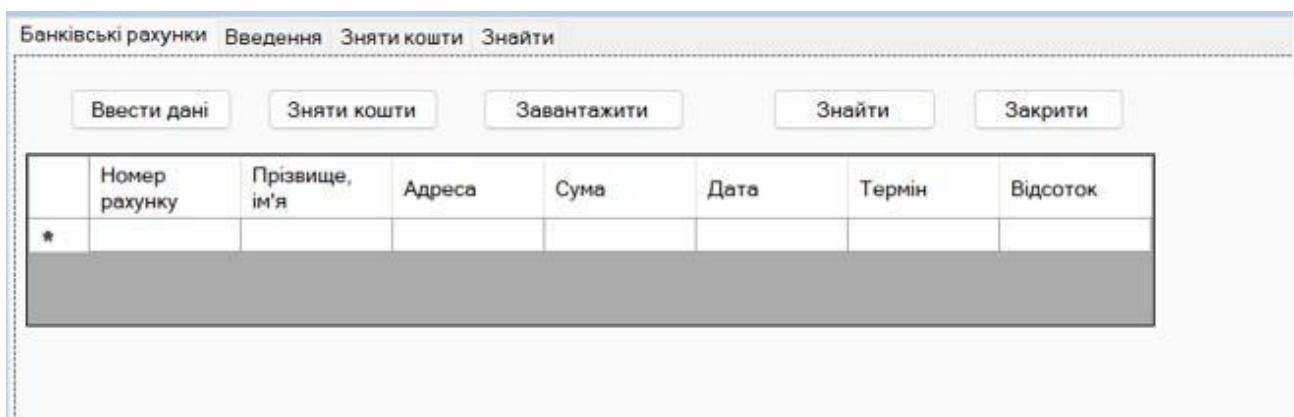


Рисунок 2.2 – Результат виконання програми

Висновок: на лабораторній роботі я ознайомилася із можливостями створення користувацьких класів у Visual Studio та їх використання у Windows Form додатках.