

Тема. Створення настільного додатку на основі технології Microsoft .NET з використанням мови C++ та структурованих даних.

Мета. Ознайомитись із створенням додатків на основі технології Microsoft .NET та реалізувати програму з використанням даних типу структура на мові програмування C++.

Приклад виконання завдання

Завдання. Створити настільний додаток на основі технології Microsoft .NET з використанням мови CLR (Common Language Runtime) для розв'язку наступного завдання: для участі у конференції потрібно заповнити заявку: прізвище, ім'я, адресу, а також зазначити, чи потрібно бронювати номер у готелі. Якщо житло непотрібне, то записати свій контактний телефон, в іншому випадку – зазначити називу готелю.

Створити програму, яка б давала змогу ввести отримані дані у базу даних конференції.

Для того, щоб створити проект додатка на основі технології Windows Forms необхідно виконати наступні операції.

1 Вибрати в головному меню пункт Файл / Створити / Проект... (File / New / Project...).

2 У діалоговому вікні Створення проекту (New Project) в лівій частині вікна обрати Встановлені / Шаблони / Visual C++ / CLR (Installed / Templates / Visual C++ / CLR).

3 У діалоговому вікні Створення проекту (New Project) в центральній частині вікна обрати пункт Пустий проект CLR (Empty Project CLR).

4 Задати ім'я нового проекту у полі Ім'я: (Name:).

5 Задати місце збереження нового рішення у полі Розташування: (Location:).

6 Задати ім'я нового рішення у полі Ім'я рішення: (Solution name:) та натиснути кнопку OK.

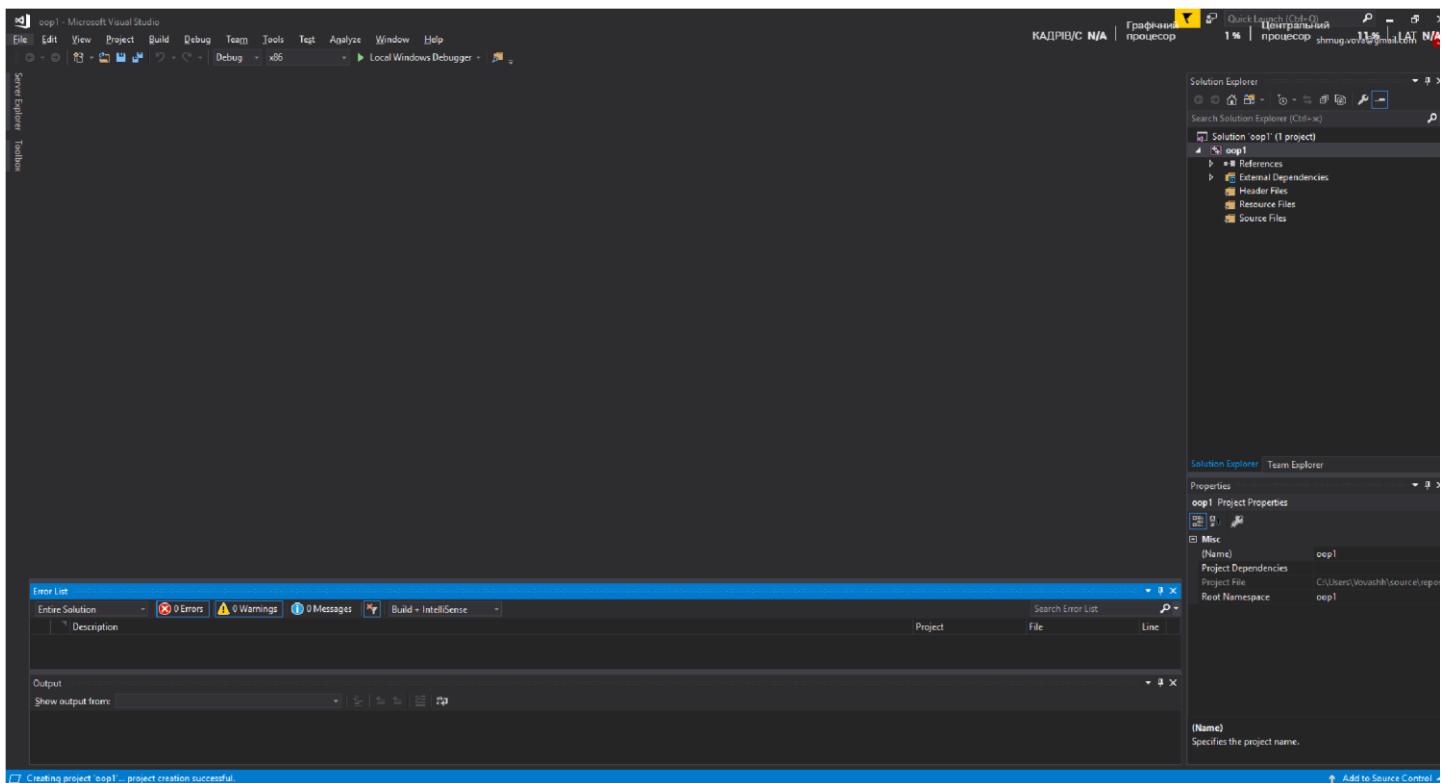


Рисунок 1.1 – Вікно середовища Visual Studio зі створеним пустим проектом

У результаті вище зазначених дій буде створений пустий проект CLR у якого відсутні будь-які елементи (див. Рис. 1.1).

CLR

Для можливості подальшого створення віконного додатку Windows за технологією Windows Forms у створений пустий проект CLR необхідно додати пусту форму (Form), яка буде вікном майбутнього додатку.

7 Обрати в меню пункт Проект / Додати новий елемент... (Project / Add new item...).

8 З'явиться діалогове вікно Додати новий елемент (Add New Item...), в лівій частині якого слід обрати пункт Встановлені / Visual C++ / UI (Installed / Visual C++ / UI).

9 Надалі в центральній частині діалогового вікна Додати новий елемент (Add New Item) обрати пункт Форма Windows Forms (Form Windows Forms).

10 У полі Ім'я: (Name:) задати ім'я нової форми (файл заголовку з розширенням .h).

11 У полі Розташування: (Location:) задається місце збереження файлу нової форми.

12 По завершенню вибору натиснути кнопку Додати (Add) для додавання нової форми у проект (див. Рис. 1.2).

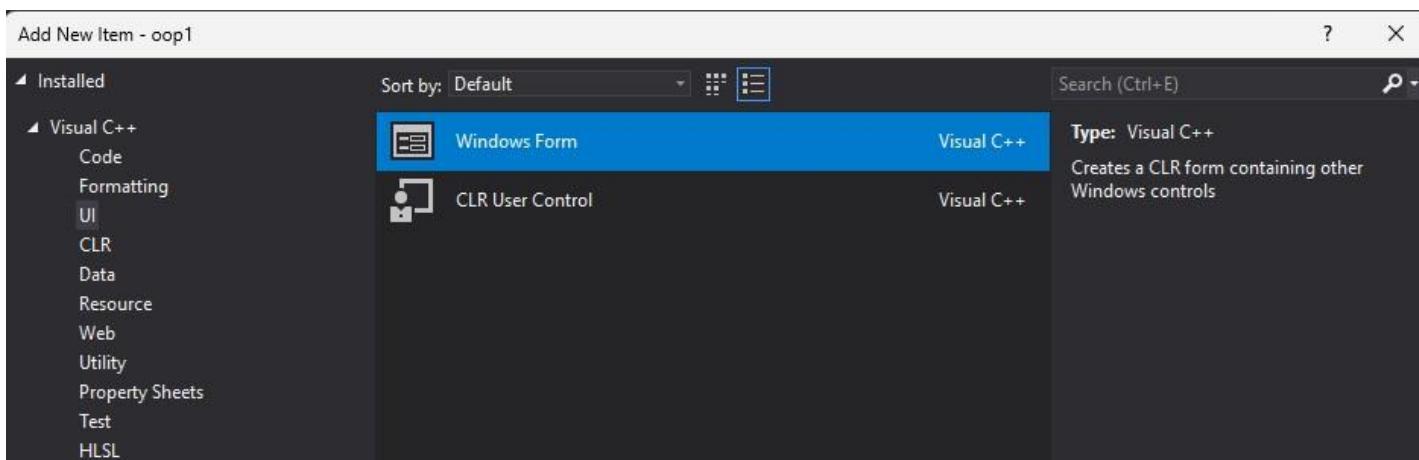


Рисунок 1.2 – Діалогове вікно додавання нового елементу до проекту CLR

13 Далі потрібно додати наступний програмний код у файл форми з розширенням .cpp (має таке ж ім'я, як і файл заголовків форми), який обирається в Оглядачі рішень (Solution Explorer).

```

1 #include "MyForm.h"
2 using namespace oop1;
3 [STAThreadAttribute]
4 int main(array<System::String ^> ^args)
5 {
6     Application::EnableVisualStyles();
7     Application::SetCompatibleTextRenderingDefault(false);
8     Application::Run(gcnew MyForm());
9     return 0;
10 }
```

Рисунок 1.3 - Код для cpp.

14 Обрати пункт меню Проект / Властивості: Проект1 (Project / Project1

Properties...), після чого з'явиться діалогове вікно Сторінки властивостей Проект1 (Project1 Property Pages).

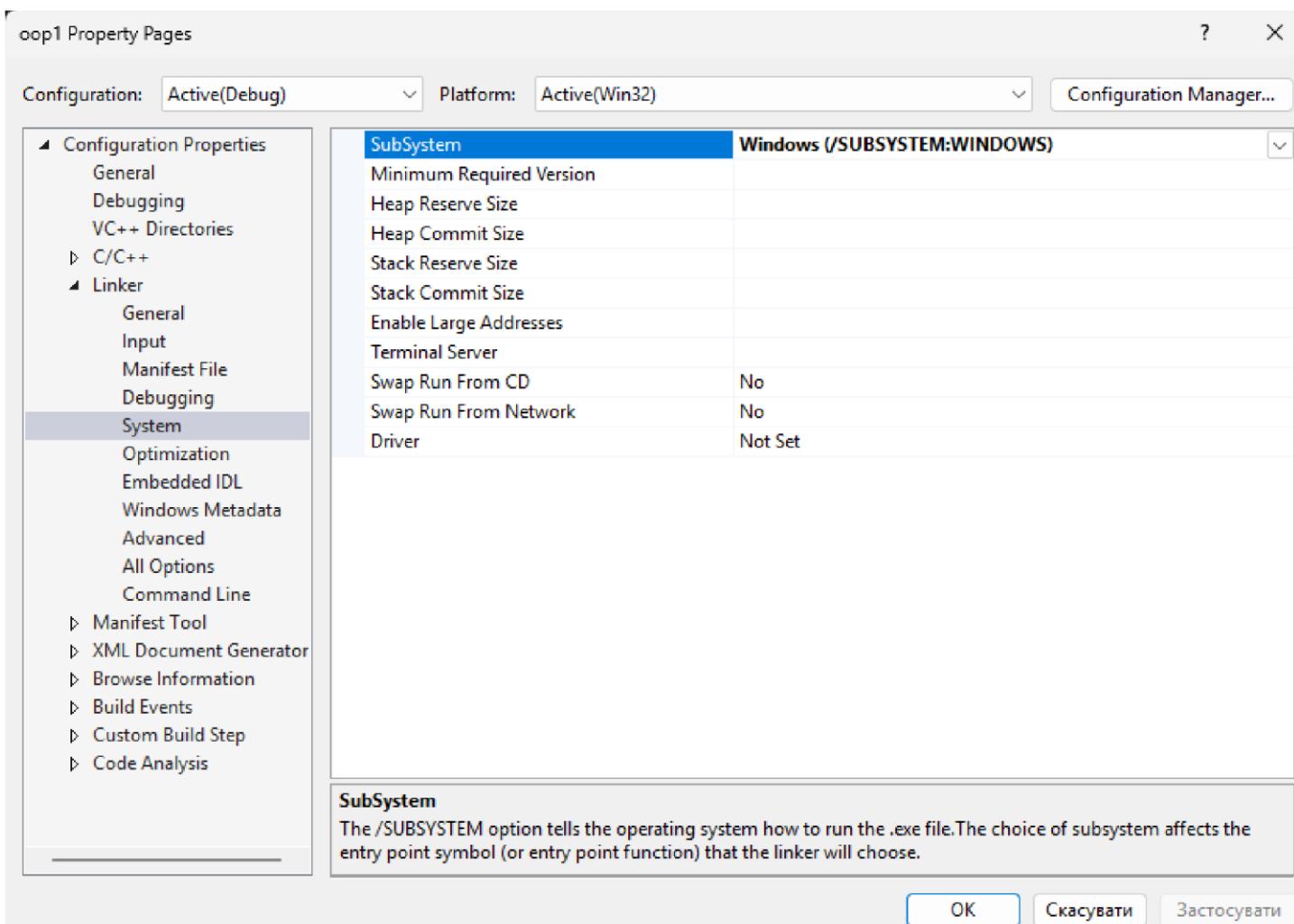


Рисунок 1.4 – Задавання значення поля Підсистема у властивостях проекту

15 У лівій частині діалогового вікна обираємо пункт Властивості конфігурації / Компонувальник / Система (Configuration Properties / Linker / System).

16 У правій частині діалогового вікна у полі Підсистема (SubSystem) задаємо Windows(/SUBSYSTEM:WINDOWS), як вказано на рисунку 3.3.

17 У лівій частині діалогового вікна обираємо пункт Властивості конфігурації / Компонувальник / Додатково (Configuration properties / Linker / Advanced).

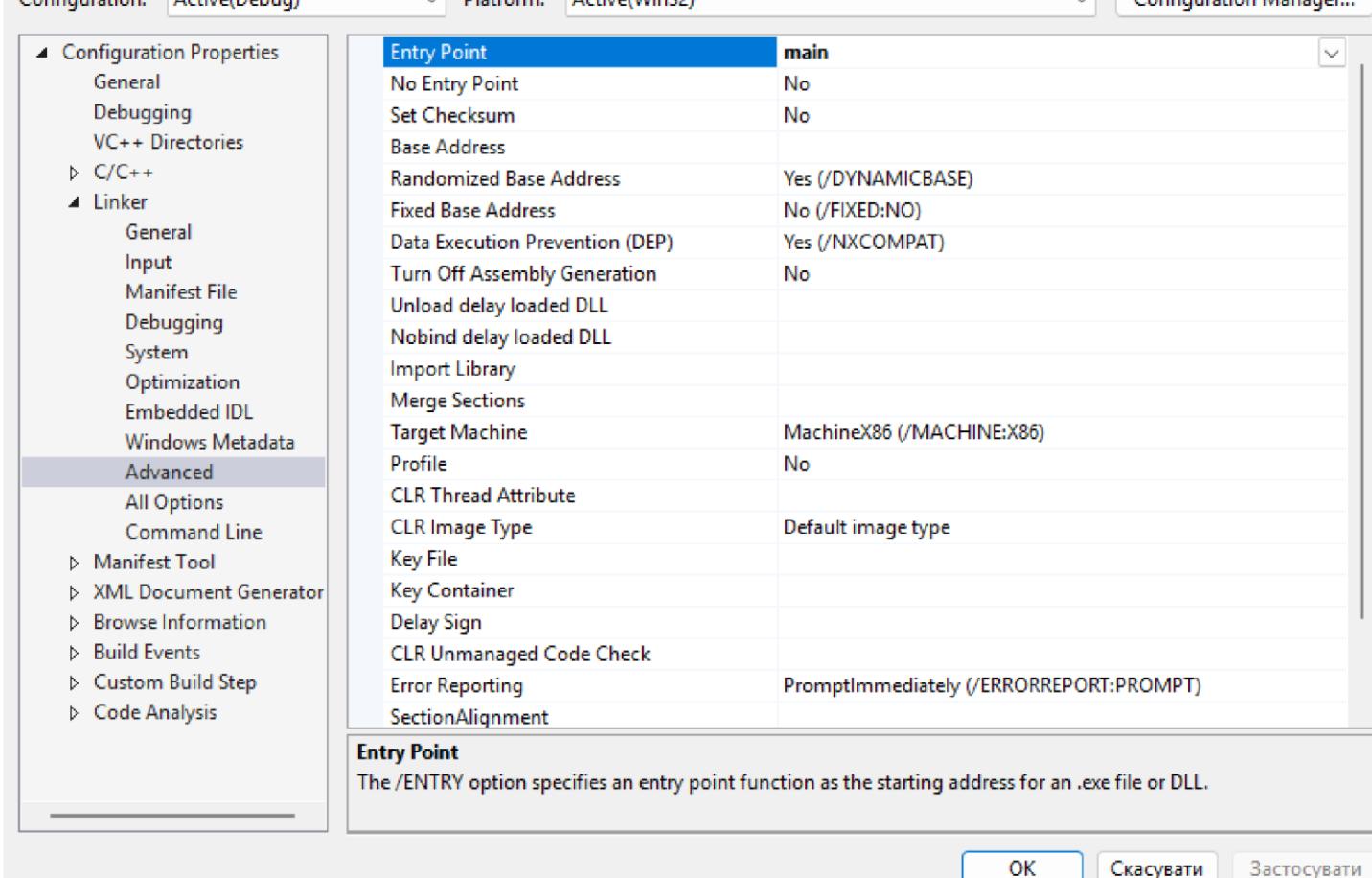


Рисунок 3.3 – Задавання значення поля Підсистема у властивостях проекту

19 У правій частині діалогового вікна у полі Точка входу (Entry Point) задаємо `main`, як вказано на рисунку 3.4 і натискаємо кнопку OK для застосування змін.

Рисунок 1.5 – Задавання значення поля Точка входу у властивостях проекту

Після цього ми отримаємо готовий до роботи додаток Windows Forms з пустою формою у ньому.

20 Створити об’єднання та структуру для розв’язку завдання. Оголосити масив об’єктів структури у файлі заголовків (*.h).

```
union gotel_tel
{
    char* gotel;
    char* tel;
};

public struct konfer
{
    char* Name;
    char* Surname;
    char* Adress;
    int br;
    gotel_tel bron;
};
konfer mas[n];
```

Рисунок 1.6 – Об’єднання структури

21 Підключити директиву для визначення значення n.

```
#define n 10
```

Рисунок 1.7 - Підключення директиви

Спроектувати форму для реєстрації та пошуку інформації про учасників конференції.

22 Використати глобальну змінну для підрахунку кількості введених елементів масиву об’єктів структури.

```
int qounity = 0;
```

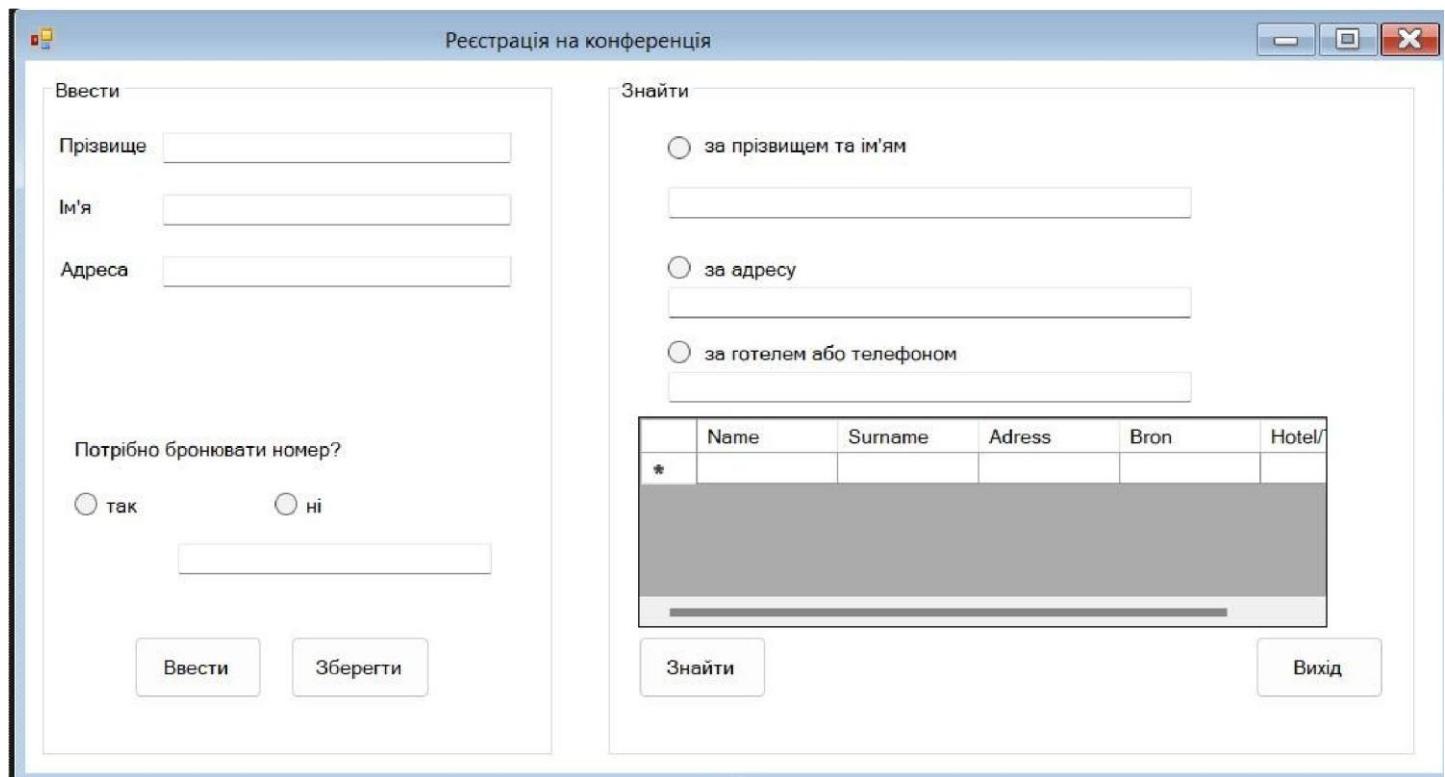
Рисунок 1.9 - Кількість введених об’єктів структури

23 Використати простір імен

```
using namespace System::Runtime::InteropServices;
```

Рисунок 1.10 – Ім'я

24 Запрограмувати кнопку «Ввести» для введення даних для масиву об'єктів структури.



```
this->label5->Text = "Введіть назву готелю";
}
private: System::Void radioButton2_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {
    if (this->radioButton2->Checked)
    {
        }
    this->label5->Text = "Введіть номер телефону";
}
```

25 Запрограмувати зміну тексту в компоненті label5 відносно вибору перемикачів.

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    if (qounity < n)
    {
        mas[qounity].Name = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox1->Text);
        mas[qounity].Surname = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox3->Text);
        mas[qounity].Adress = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox7->Text);

        if (this->radioButton1->Checked)
        {
            mas[qounity].br = 1;
            mas[qounity].bron.gotel = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox2->Text);
        }
        else if (this->radioButton2->Checked)
        {
            mas[qounity].br = 2;
            mas[qounity].bron.tel = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox2->Text);
        }

        qounity++;
        textBox1->Text = "";
        textBox3->Text = "";
        textBox7->Text = "";
        textBox2->Text = "";
    }
    else
    {
        MessageBox::Show("База учасників заповнена", "Попередження");
        button1->Enabled = false;
    }
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    FILE * file;
    file = fopen("konferenc.txt", "wt");
    char* s = "Прізвище\tІм'я\tАдресса\n";
    fputs(s, file);
    for (int j = 0; j < qounity; j++)
    {
        fprintf(file, "%s\t%s\t%s\t%s\t%s\n", mas[j].Name, mas[j].Surname, mas[j].Adress, mas[j].bron.gotel);
    }
    fclose(file);
}

```

Рисунок 1.12 - Код для label 15

26 Запрограмувати кнопку «Зберегти» для збереження даних у файл.

27 Створити відкриту функцію пошуку в масиві структур за прізвищем та ім'ям учасника конференції у класі MyForm. Знайдені дані зберегти у результуючий файл.

```

FILE* f;
f = fopen("result.txt", "wt");
char* s = "Прізвище\tІм'я\tАдреса\tБронь\tГотель\n";
fputs(s, f);
dataGridView1->Rows->Clear();
int k = 0;
for (int j = 0; j < quantity; j++)
{
    if (mas[j].br == 0 && strcmp(mas[j].bron.gotel, ht) == 0)
    {
        // запис у файл
        fprintf(f, "%s\t%s\t%s\t%s\t", mas[j].Name, mas[j].Surname, mas[j].Adress);
        fprintf(f, "%s\t");
        fprintf(f, "%s\n", mas[j].bron.gotel);

        // запис у dataGridView
        dataGridView1->RowCount = quantity;

        String^ s_name = gcnew String(mas[j].Name);
        this->dataGridView1->Rows[k]->Cells[0]->Value = s_name;

        String^ s_surname = gcnew String(mas[j].Surname);
        this->dataGridView1->Rows[k]->Cells[1]->Value = s_surname;

        String^ s_adress = gcnew String(mas[j].Adress);
        this->dataGridView1->Rows[k]->Cells[2]->Value = s_adress;

        this->dataGridView1->Rows[k]->Cells[3]->Value = "так";
        String^ s_gotel = gcnew String(mas[j].bron.gotel);
        this->dataGridView1->Rows[k]->Cells[4]->Value = s_gotel;

        k++;
    }

    if (mas[j].br == 2 && strcmp(mas[j].bron.tel, ht) == 0)
    {
        // запис у файл
        fprintf(f, "%s\t%s\t%s\t%s\t", mas[j].Name, mas[j].Surname, mas[j].Adress);
        fprintf(f, "%s\t");
        fprintf(f, "%s\n", mas[j].bron.tel);

        // запис у dataGridView
        String^ s_name = gcnew String(mas[j].Name);
        this->dataGridView1->Rows[k]->Cells[0]->Value = s_name;

        String^ s_surname = gcnew String(mas[j].Surname);
        this->dataGridView1->Rows[k]->Cells[1]->Value = s_surname;

        String^ s_adress = gcnew String(mas[j].Adress);
        this->dataGridView1->Rows[k]->Cells[2]->Value = s_adress;

        this->dataGridView1->Rows[k]->Cells[3]->Value = "ні";
        String^ s_gotel = gcnew String(mas[j].bron.gotel);
        this->dataGridView1->Rows[k]->Cells[4]->Value = s_gotel;

        k++;
    }
}
fclose(f);

```

Рисунок 1.14 - Реалізація Пошуку

28 Аналогічно реалізувати функцію пошуку за адресом учасника конференції.

```

void SearchAdress(char* adress) {
    FILE *f;
    f = fopen("result.txt", "wt");
    char* s = "Прізвище\tІм'я\tАдреса\tБронь\tГотель/Телефон";
    fputs(s, f);
    dataGridView1->Rows->Clear();
    int k = 0;
    for (int j = 0; j < qounity; j++) {
        if (strcmp(mas[j].Adress, adress) == 0) {
            // запис у файл
            fprintf(f, "%s\t%s\t%s\t%s\t", mas[j].Name, mas[j].Surname, mas[j].Adress);
            // запис у dataGridView
            dataGridView1->RowCount = qounity;
            String^ s_name = gcnew String(mas[j].Name);
            this->dataGridView1->Rows[k]->Cells[0]->Value = s_name;
            String^ s_surname = gcnew String(mas[j].Surname);
            this->dataGridView1->Rows[k]->Cells[1]->Value = s_surname;
            String^ s_adress = gcnew String(mas[j].Adress);
            this->dataGridView1->Rows[k]->Cells[2]->Value = s_adress;
            if (mas[j].br == 1) {
                fprintf(f, "Так\t\t");
                fprintf(f, "%s\n", mas[j].bron.gotel);
                this->dataGridView1->Rows[k]->Cells[3]->Value = "Так";
                String^ s_gotel = gcnew String(mas[j].bron.gotel);
                this->dataGridView1->Rows[k]->Cells[4]->Value = s_gotel;
            }
            else {
                fprintf(f, "Ні\t\t");
                fprintf(f, "%s\n", mas[j].bron.tel);
                this->dataGridView1->Rows[k]->Cells[3]->Value = "Ні";
                String^ s_tel = gcnew String(mas[j].bron.tel);
                this->dataGridView1->Rows[k]->Cells[4]->Value = s_tel;
            }
            k++;
        }
    }
}

```

Рисунок 1.15 - Пошук за адресом

29 Створити функцію пошуку в масиві структур за телефоном або готелем.

```

void SearchHotelTel(char* ht)
{
    FILE* f;
    f = fopen("result.txt", "wt");
    char* s = "Прізвище\тІм'я\тАдреса\тБронь\тГотель\нТелефон\n";
    fputs(s, f);
    dataGridView1->Rows->Clear();
    int k = 0;
    for (int j = 0; j < quantity; j++)
    {
        if (mas[j].br == 0 && strcmp(mas[j].bron.gotel, ht) == 0)
        {
            // запис у файл
            fprintf(f, "%s\t%s\t%s\t%s\t", mas[j].Name, mas[j].Surname, mas[j].Adress);
            fprintf(f, "%s\t");
            fprintf(f, "%s\n", mas[j].bron.gotel);

            // запис у dataGridView
            dataGridView1->RowCount = quantity;

            String* s_name = gcnew String(mas[j].Name);
            this->dataGridView1->Rows[k]->Cells[0]->Value = s_name;

            String* s_surname = gcnew String(mas[j].Surname);
            this->dataGridView1->Rows[k]->Cells[1]->Value = s_surname;

            String* s_adress = gcnew String(mas[j].Adress);
            this->dataGridView1->Rows[k]->Cells[2]->Value = s_adress;

            this->dataGridView1->Rows[k]->Cells[3]->Value = "зак";
            String* s_gotel = gcnew String(mas[j].bron.gotel);
            this->dataGridView1->Rows[k]->Cells[4]->Value = s_gotel;

            k++;
        }

        if (mas[j].br == 2 && strcmp(mas[j].bron.tel, ht) == 0)
        {
            // запис у файл
            fprintf(f, "%s\t%s\t%s\t%s\t", mas[j].Name, mas[j].Surname, mas[j].Adress);
            fprintf(f, "%s\t");
            fprintf(f, "%s\n", mas[j].bron.tel);

            // запис у dataGridView
            String* s_name = gcnew String(mas[j].Name);
            this->dataGridView1->Rows[k]->Cells[0]->Value = s_name;

            String* s_surname = gcnew String(mas[j].Surname);
            this->dataGridView1->Rows[k]->Cells[1]->Value = s_surname;

            String* s_adress = gcnew String(mas[j].Adress);
            this->dataGridView1->Rows[k]->Cells[2]->Value = s_adress;

            this->dataGridView1->Rows[k]->Cells[3]->Value = "ні";
            String* s_gotel = gcnew String(mas[j].bron.gotel);
            this->dataGridView1->Rows[k]->Cells[4]->Value = s_gotel;

            k++;
        }
    }
}

```

Рисунок 1.16 - Пошук за готелем або номером

30 Запрограмувати кнопку «Знайти» для пошуку даних їх збереження у результируючий файл та на формі у компоненті dataGridView1.

31 Запрограмувати кнопку «Вихід».

```

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    if (radioButton3->Checked) {
        char* str = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox4->Text);
        char* s1 = strtok(str, " ");
        char* s2 = strtok(NULL, " ");
        SearchNameSurname(s1, s2);
    }
}

if (radioButton4->Checked) {
    char* s = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox5->Text);
    SearchAddress(s);
}

if (radioButton5->Checked) {
    char* sht = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox6->Text);
    SearchHotelTel(sht);
}

```

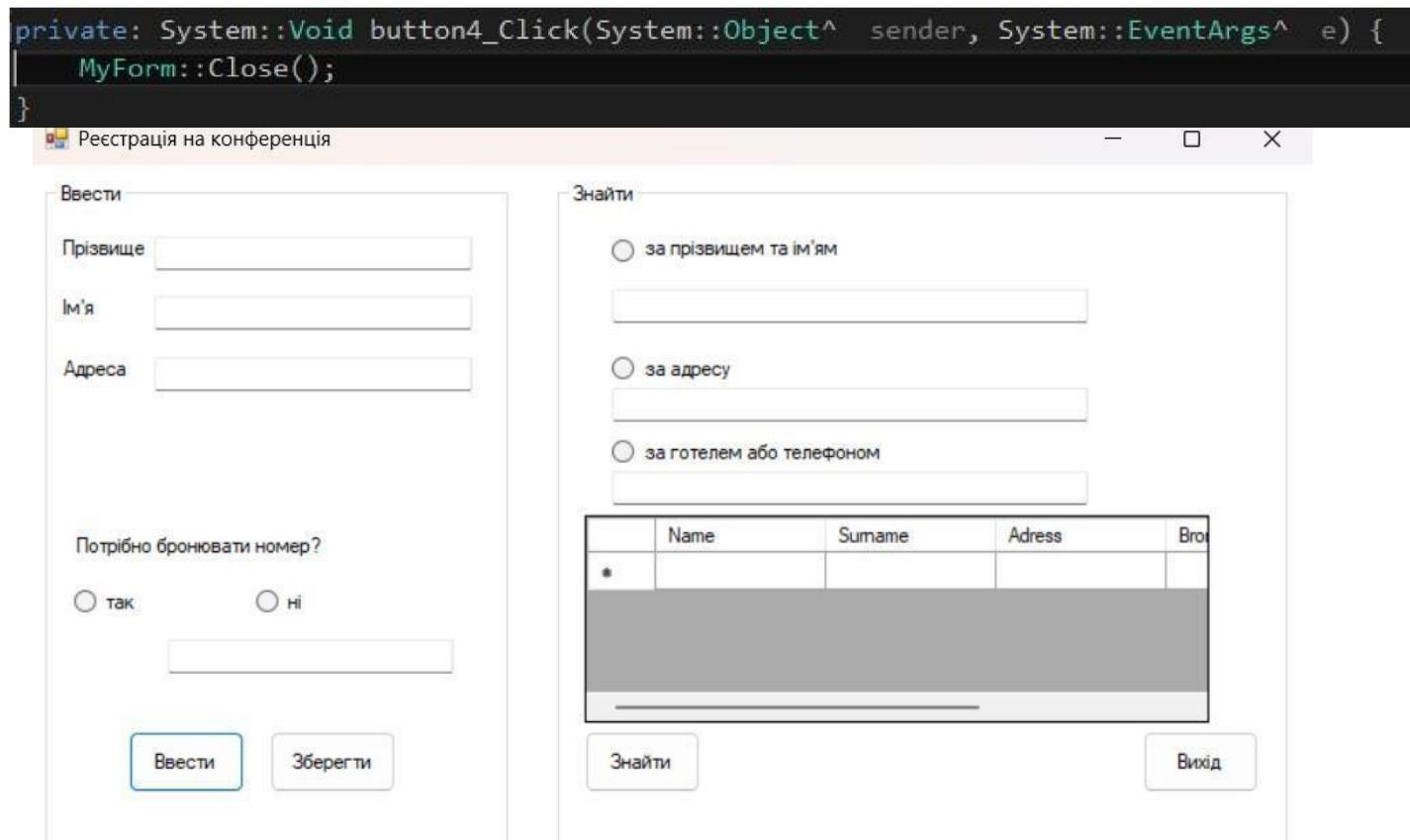


Рисунок 1.19 - Робоче поле

Висновок: Ознайомилась із створенням додатків на основі технології Microsoft .NET та реалізувала програму з використанням даних типу структура на мові програмування C++.