

## Лабораторна робота 7

Тема. Створення настільного додатку на основі технології Microsoft .NET з використанням перевизначення операцій в класах на мові C++.

Мета. Ознайомитись із можливостями створення та використання перевизначення операцій в класах у Visual Studio C++.

## Хід роботи

## 1. Приклад Завдання.

Створити настільний додаток на основі технології Microsoft .NET з використанням мови CLR (Common Language Runtime) для розв'язку наступного завдання: нехай дано клас для зберігання інформації про оцінки студентів з різних предметів. Реалізувати перевизначення операції інкременту в класі. Ознайомитись та запрограмувати експорт даних із компоненти dataGridView1 в Excel. Передбачити завантаження вхідних даних із текстового файлу та збереження результатів у текстовому файлі.

## Лістинг 1.1 – Код програми 1

```
#pragma once
#define
_CRT_SECURE_NO_WARNINGS #include <cstdio> #define n 5 class Teaching
{
    char* student;
    char* group; char* subject; char* teacher; char* date;
    int mark; public:
    void operator++(int);
    void Save(char* st, char* g, char* su, char* t, char* d, int m);
}mas[n];
void Teaching::Save(char* st, char* g, char* su, char* t, char* d, int
m) { student = st; group = g; subject = su; teacher = t;
date = d; mark = m;
}
void Teaching::operator++(int) { FILE* file = fopen("mark.txt", "a");
fprintf(file, "%s\t%s\t%s\t%s\t%s\t%i\n", student, group, subject, teacher,
date, mark);
fclose(file);
}
```

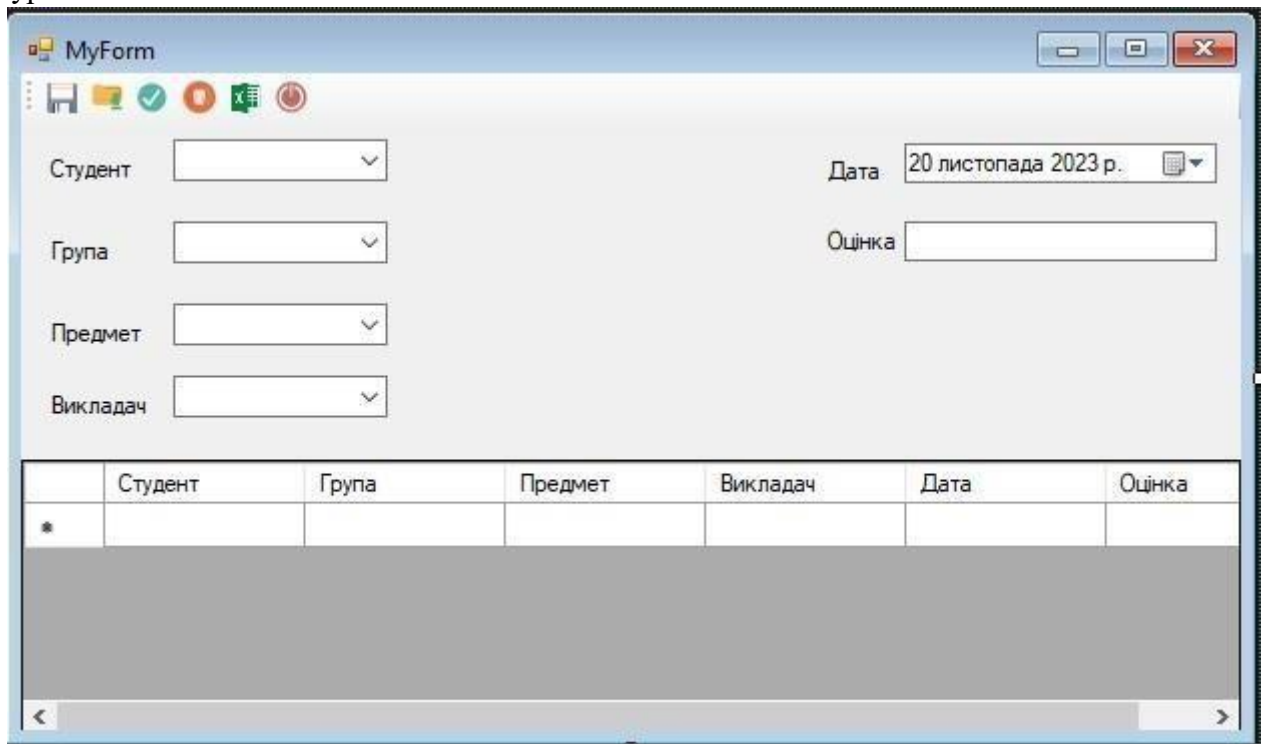


Рисунок 1.1 – Вигляд програми прикладу

## Лістинг 1.2 – Код меню

```

private: System::Void toolStripButton1_Click(System::Object^ sender,
System::EventArgs^ e) {          char* s1 =
(char*)(void*)Marshal::StringToHGlobalAnsi(comboBox1->Text);    char* s2
=
(char*)(void*)Marshal::StringToHGlobalAnsi(comboBox2->Text);    char* s3
=
(char*)(void*)Marshal::StringToHGlobalAnsi(comboBox3->Text);    char* s4
=
(char*)(void*)Marshal::StringToHGlobalAnsi(comboBox4->Text);    char* s5
=
(char*)(void*)Marshal::StringToHGlobalAnsi(dateTimePicker1-
>Value.ToShortDateString());    int s6 =
Convert::ToInt32(textBox1->Text);
    mas[qountity].Save(s1, s2, s3, s4, s5, s6);
    qountity++;
    comboBox1->Text = "";    comboBox2->Text = "";    comboBox3->Text = "";
comboBox4->Text = "";    textBox1->Clear();
} private: System::Void toolStripButton2_Click(System::Object^
sender, System::EventArgs^ e) {          for (int i = 0; i < qountity; i++) {
    mas[i]++;
}
}
private: System::Void toolStripButton3_Click(System::Object^ sender,
System::EventArgs^ e) {    dataGridView1->Rows->Clear();
array<String^>^ lines = File::ReadAllLines("mark.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);
        }
    }
    private: System::Void toolStripButton4_Click(System::Object^ sender,
System::EventArgs^ e) {
        for each (DataGridViewRow ^ Tw in
dataGridView1->Rows) {
            if (Tw->Selected == true) {
                dataGridView1>Rows->Remove(Tw);
            }
        }
        FILE* file = fopen("mark.txt", "w");
        for (int i = 0; i <= dataGridView1->Rows->Count - 1; i++)
        {
            for (int j = 0; j <= dataGridView1->Columns->Count - 1; j++)
            {
                String^ s = Convert::ToString(dataGridView1->Rows[i]->Cells[j]-
>Value);
                char* sLine =
(char*)(void*)Marshal::StringToHGlobalAnsi(s);
                fprintf(file, "%s", sLine);
                if (j !=
dataGridView1->Columns->Count - 1) {
                    fprintf(file, "\t");
                }
                fprintf(file, "\n");
            }
        }
        fclose(file);
    }
    private: System::Void toolStripButton5_Click(System::Object^ sender,
System::EventArgs^ e) {
        Microsoft::Office::Interop::Excel::Application^ ExcelApp = gcnew
Microsoft::Office::Interop::Excel::ApplicationClass();
        Microsoft::Office::Interop::Excel::Worksheet^ ExcelActiveWorkSheet;
        Microsoft::Office::Interop::Excel::Workbook^ ExcelWorkBook =
ExcelApp>Workbooks>Add(Type::Missing);
        ExcelWorkBook->Worksheets->Add(Type::Missing, Type::Missing, 1,
Type::Missing);
        safe_cast<Worksheet^>(ExcelApp->ActiveWorkbook-
>Sheets[1])>Select(Type::Missing);
        ExcelActiveWorkSheet = safe_cast<Worksheet^>(ExcelApp->ActiveSheet);
        for
        (int i = 0; i < dataGridView1->Rows->Count; i++) {
            ExcelActiveWorkSheet->Cells[1, i + 1] = dataGridView1->Columns[i]-
>HeaderText;
        }
        for (int i = 0; dataGridView1->Rows->Count; i++) {
            for (int j = 0; j <
dataGridView1->Columns->Count; j++) {

```

```
ExcelActiveWorkSheet->Cells[i + 2, j + 1] = dataGridView1-
>Rows[i]->Cells[j]>Value;
    }
}
ExcelApp->Visible = true;
ExcelApp->UserControl = true;
}
private: System::Void toolStripButton6_Click(System::Object^ sender,
System::EventArgs^ e) {
    Close();
}
private: System::Void comboBox1_MouseClick(System::Object^
sender, System::Windows::Forms::MouseEventArgs^ e) {    array<String^>^
lines = File::ReadAllLines("students.txt",
System::Text::Encoding::GetEncoding(1251));    comboBox1->Items-
>Clear();    for each
(String ^ str in lines) {        for (int i = 0; i < str-
>Split('\t')->Length; i++) {            comboBox1>Items-
>Add(str->Split('\t')[i]);                break;
        }
    }
}
private: System::Void comboBox2_MouseClick(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {    array<String^>^ lines =
File::ReadAllLines("group.txt",
System::Text::Encoding::GetEncoding(1251));    comboBox2->Items-
>Clear();    for each
(String ^ str in lines) {        for (int i = 0; i <
str>Split('\t')->Length; i++) {            comboBox2-
>Items->Add(str->Split('\t')[i]);                break;
        }
    }
}
private: System::Void comboBox3_MouseClick(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {    array<String^>^ lines =
File::ReadAllLines("subject.txt",
System::Text::Encoding::GetEncoding(1251));    comboBox3->Items->Clear();
    comboBox4>Items->Clear();    for each (String
^ str in lines) {        for (int i = 0; i < str->Split('\t')-
>Length; i++) {            comboBox3->Items->Add(str->Split('\t')[i]);
            i = i + 2;
            comboBox4->Items->Add(str->Split('\t')[i]);
            break;
        }
    }
}
```

	Студент	Група	Предмет	Викладач	Дата	Оцінка
*						

Рисунок 1.2 – Результат роботи програми

Висновок: на лабораторній роботі я ознайомилась із можливостями створення та використання перевизначення операцій в класах у Visual Studio C++.