**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ВСП – Технічний коледж**

**НУ «Львівська політехніка»**

Відділення

Інформаційних технологій

Та комп`ютерної техніки

**ЗВІТ**

Про виконання лабораторної роботи №6

З предмету «Навчальна практика з основ об'єктно-орієнтованого програмування»

Виконав

Студент групи 31 ПЗ

Тисляк А.О.

Перевірив викладач:

Чистяк В.В.

Львів 2021

**Лабораторна робота №6 «Створення програм з GUI»**

**Мета роботи:** ознайомитись ключовими поняттями, методами та засобами платформи .Net та технологій Windows Forms та ADO.Net

*Варіант 10*

**Завдання:** реалізувати графічний інтерфейс користувача для перегляду, додавання, оновлення та видалення інформації про об'єкт відповідного класу. Інформація має зберігатись у базі даних. База даних містить від 10 до 15 записів і містити 1 таблицю.

**Завдання мого варіанту:**



**Код програми:**

***MyForm.h***

#pragma once

namespace ExampleCPP {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

}

protected:

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::MenuStrip^ menuStrip1;

protected:

private: System::Windows::Forms::ToolStripMenuItem^ ActionToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ LoadToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ AddToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ UpdateToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ DeleteToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ AboutProgramToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ ExitToolStripMenuItem;

private: System::Windows::Forms::DataGridView^ dataGridView1;

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

private: System::Windows::Forms::Button^ button\_delete;

private: System::Windows::Forms::Button^ button\_update;

private: System::Windows::Forms::Button^ button\_add;

private: System::Windows::Forms::Button^ button\_download;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ surname;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ name;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ fathername;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ group;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ math;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ english;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ softdev;

private:

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

void InitializeComponent(void)

{

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->ActionToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->AboutProgramToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->ExitToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->LoadToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->AddToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->UpdateToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->DeleteToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());

this->button\_download = (gcnew System::Windows::Forms::Button());

this->button\_add = (gcnew System::Windows::Forms::Button());

this->button\_update = (gcnew System::Windows::Forms::Button());

this->button\_delete = (gcnew System::Windows::Forms::Button());

this->surname = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->name = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->fathername = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->group = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->math = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->english = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->softdev = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->menuStrip1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

this->groupBox1->SuspendLayout();

this->SuspendLayout();

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->ActionToolStripMenuItem,

this->AboutProgramToolStripMenuItem, this->ExitToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(675, 24);

this->menuStrip1->TabIndex = 0;

this->menuStrip1->Text = L"menuStrip1";

this->ActionToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(4) {

this->LoadToolStripMenuItem,

this->AddToolStripMenuItem, this->UpdateToolStripMenuItem, this->DeleteToolStripMenuItem

});

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

this->ActionToolStripMenuItem->Size = System::Drawing::Size(70, 20);

this->ActionToolStripMenuItem->Text = L"Action";

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

this->AboutProgramToolStripMenuItem->Size = System::Drawing::Size(101, 20);

this->AboutProgramToolStripMenuItem->Text = L"About program";

this->AboutProgramToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::AboutProgramToolStripMenuItem\_Click);

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

this->ExitToolStripMenuItem->Size = System::Drawing::Size(54, 20);

this->ExitToolStripMenuItem->Text = L"Exit";

this->ExitToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::ExitToolStripMenuItem\_Click);

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

this->LoadToolStripMenuItem->Size = System::Drawing::Size(180, 22);

this->LoadToolStripMenuItem->Text = L"Load";

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

this->AddToolStripMenuItem->Size = System::Drawing::Size(180, 22);

this->AddToolStripMenuItem->Text = L"Add";

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

this->UpdateToolStripMenuItem->Size = System::Drawing::Size(180, 22);

this->UpdateToolStripMenuItem->Text = L"Update";

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

this->DeleteToolStripMenuItem->Size = System::Drawing::Size(180, 22);

this->DeleteToolStripMenuItem->Text = L"Delete";

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(7) {

this->surname, this->name,

this->fathername, this->group,

this->math, this->english, this->softdev

});

this->dataGridView1->Location = System::Drawing::Point(12, 37);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->Size = System::Drawing::Size(445, 213);

this->dataGridView1->TabIndex = 1;

this->groupBox1->Controls->Add(this->button\_delete);

this->groupBox1->Controls->Add(this->button\_update);

this->groupBox1->Controls->Add(this->button\_add);

this->groupBox1->Controls->Add(this->button\_download);

this->groupBox1->Location = System::Drawing::Point(481, 37);

this->groupBox1->Name = L"groupBox1";

this->groupBox1->Size = System::Drawing::Size(182, 213);

this->groupBox1->TabIndex = 2;

this->groupBox1->TabStop = false;

this->groupBox1->Text = L"Action";

this->button\_download->Font = (gcnew System::Drawing::Font(L"Monotype Corsiva", 14.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button\_download->Location = System::Drawing::Point(17, 30);

this->button\_download->Name = L"button\_download";

this->button\_download->Size = System::Drawing::Size(140, 34);

this->button\_download->TabIndex = 0;

this->button\_download->Text = L"Load";

this->button\_download->UseVisualStyleBackColor = true;

this->button\_download->Click += gcnew System::EventHandler(this, &MyForm::button\_download\_Click);

this->button\_add->Font = (gcnew System::Drawing::Font(L"Monotype Corsiva", 14.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button\_add->Location = System::Drawing::Point(17, 70);

this->button\_add->Name = L"button\_add";

this->button\_add->Size = System::Drawing::Size(140, 34);

this->button\_add->TabIndex = 1;

this->button\_add->Text = L"Add";

this->button\_add->UseVisualStyleBackColor = true;

this->button\_add->Click += gcnew System::EventHandler(this, &MyForm::button\_add\_Click);

this->button\_update->Font = (gcnew System::Drawing::Font(L"Monotype Corsiva", 14.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button\_update->Location = System::Drawing::Point(17, 110);

this->button\_update->Name = L"button\_update";

this->button\_update->Size = System::Drawing::Size(140, 34);

this->button\_update->TabIndex = 2;

this->button\_update->Text = L"Update";

this->button\_update->UseVisualStyleBackColor = true;

this->button\_update->Click += gcnew System::EventHandler(this, &MyForm::button\_update\_Click);

this->button\_delete->Font = (gcnew System::Drawing::Font(L"Monotype Corsiva", 14.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button\_delete->Location = System::Drawing::Point(17, 150);

this->button\_delete->Name = L"button\_delete";

this->button\_delete->Size = System::Drawing::Size(140, 34);

this->button\_delete->TabIndex = 3;

this->button\_delete->Text = L"Delete";

this->button\_delete->UseVisualStyleBackColor = true;

this->button\_delete->Click += gcnew System::EventHandler(this, &MyForm::button\_delete\_Click);

this->surname->HeaderText = L"Surname";

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

this->name->HeaderText = L"Name";

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

this->fathername->HeaderText = L"Fathername";

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

this->group->HeaderText = L"Group";

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

this->math->HeaderText = L"Math";

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

this->english->HeaderText = L"English";

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

this->softdev->HeaderText = L"SoftDev";

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

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(675, 268);

this->Controls->Add(this->groupBox1);

this->Controls->Add(this->dataGridView1);

this->Controls->Add(this->menuStrip1);

this->MainMenuStrip = this->menuStrip1;

this->MaximizeBox = false;

this->MaximumSize = System::Drawing::Size(691, 307);

this->MinimumSize = System::Drawing::Size(691, 307);

this->Name = L"MyForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Working example WInForms c++ in ACCESS ";

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

this->groupBox1->ResumeLayout(false);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void ExitToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void AboutProgramToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void button\_download\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void button\_add\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void button\_update\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void button\_delete\_Click(System::Object^ sender, System::EventArgs^ e);

};

}

***Main6.cpp***

#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

using namespace System::Data::OleDb;

[STAThread]

int main(array<String^>^ arg) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

ExampleCPP::MyForm form;

Application::Run(% form);

}

System::Void ExampleCPP::MyForm::ExitToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

Application::Exit();

}

System::Void ExampleCPP::MyForm::AboutProgramToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

MessageBox::Show("This program shows simple work of Windows Forms c++ and ACCESS.", "Warning!");

return System::Void();

}

System::Void ExampleCPP::MyForm::button\_download\_Click(System::Object^ sender, System::EventArgs^ e)

{

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

dbConnection->Open();

String^ query = "SELECT \* FROM Table1";

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

OleDbDataReader^ dbReader = dbComand->ExecuteReader();

if (dbReader->HasRows == false) {

MessageBox::Show("Error in reading the DB!", "Error!");

}

else {

while (dbReader->Read()) {

dataGridView1->Rows->Add(dbReader["Surname"], dbReader["Name"], dbReader["Fathername"], dbReader["Group"], dbReader["Math"], dbReader["English"], dbReader["SoftDev"]);

}

}

dbReader->Close();

dbConnection->Close();

return System::Void();

}

System::Void ExampleCPP::MyForm::button\_add\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (dataGridView1->SelectedRows->Count != 1) {

MessageBox::Show("Choose one string for addition!", "Warning!");

return;

}

int index = dataGridView1->SelectedRows[0]->Index;

if (dataGridView1->Rows[index]->Cells[0]->Value == nullptr ||

dataGridView1->Rows[index]->Cells[1]->Value == nullptr ||

dataGridView1->Rows[index]->Cells[2]->Value == nullptr ||

dataGridView1->Rows[index]->Cells[3]->Value == nullptr) {

MessageBox::Show("Not all data entered!", "Warning!");

return;

}

String^ surname = dataGridView1->Rows[index]->Cells[0]->Value->ToString();

String^ name = dataGridView1->Rows[index]->Cells[1]->Value->ToString();

String^ fathername = dataGridView1->Rows[index]->Cells[2]->Value->ToString();

String^ group = dataGridView1->Rows[index]->Cells[3]->Value->ToString();

String^ math = dataGridView1->Rows[index]->Cells[4]->Value->ToString();

String^ english = dataGridView1->Rows[index]->Cells[5]->Value->ToString();

String^ softdev = dataGridView1->Rows[index]->Cells[6]->Value->ToString();

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

dbConnection->Open();

String^ query = "INSERT INTO Table1 VALUES ('" + surname + "', '" + name + "', '" + fathername + "', '" + group + "', '" + math + "', '" + english + "', '" + softdev + "')";

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

if (dbComand->ExecuteNonQuery() != 1)

MessageBox::Show("Error in request execution!", "Error!");

else

MessageBox::Show("Data is added!", "Done!");

dbConnection->Close();

return System::Void();

}

System::Void ExampleCPP::MyForm::button\_update\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (dataGridView1->SelectedRows->Count != 1) {

MessageBox::Show("Choose one string for addition!", "Warning!");

return;

}

int index = dataGridView1->SelectedRows[0]->Index;

if (dataGridView1->Rows[index]->Cells[0]->Value == nullptr ||

dataGridView1->Rows[index]->Cells[1]->Value == nullptr ||

dataGridView1->Rows[index]->Cells[2]->Value == nullptr ||

dataGridView1->Rows[index]->Cells[3]->Value == nullptr) {

MessageBox::Show("Not all data entered!", "Warning!");

return;

}

String^ surname = dataGridView1->Rows[index]->Cells[0]->Value->ToString();

String^ name = dataGridView1->Rows[index]->Cells[1]->Value->ToString();

String^ fathername = dataGridView1->Rows[index]->Cells[2]->Value->ToString();

String^ group = dataGridView1->Rows[index]->Cells[3]->Value->ToString();

String^ math = dataGridView1->Rows[index]->Cells[4]->Value->ToString();

String^ english = dataGridView1->Rows[index]->Cells[5]->Value->ToString();

String^ softdev = dataGridView1->Rows[index]->Cells[6]->Value->ToString();

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

dbConnection->Open();

String^ query = "UPDATE Table1 SET Surname='" + surname + "', Name = '" + name + "', Group='" + group + "', Math='" + math +

"', English='" + english + "', SoftDev='" + softdev + "' WHERE surname = '" + surname + "'";

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

if (dbComand->ExecuteNonQuery() != 1)

MessageBox::Show("Error in request execution!", "Error!");

else

MessageBox::Show("Data is changed!", "Done!");

dbConnection->Close();

return System::Void();

}

System::Void ExampleCPP::MyForm::button\_delete\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (dataGridView1->SelectedRows->Count != 1) {

MessageBox::Show("Choose one string for addition!", "Warning!");

return;

}

int index = dataGridView1->SelectedRows[0]->Index;

if (dataGridView1->Rows[index]->Cells[0]->Value == nullptr)

{

MessageBox::Show("Not all data entered!", "Warning!");

return;

}

String^ surname = dataGridView1->Rows[index]->Cells[0]->Value->ToString();

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

dbConnection->Open();

String^ query = "DELETE FROM Table1 WHERE surname = " + surname;

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

if (dbComand->ExecuteNonQuery() != 1)

MessageBox::Show("Error in request execution!", "Error!");

else {

MessageBox::Show("Data is deleted!", "Done!");

dataGridView1->Rows->RemoveAt(index);

}

dbConnection->Close();

return System::Void();

}

***Utils.h***

#pragma once

// Required libraries

using namespace System;

using namespace System::Collections::Generic;

using namespace System::Windows::Forms;

using namespace System::Data::OleDb;

// Managed Class - as a structure with data

ref class Field {

public:

int id;

String^ name;

};

// Managed Class

ref class Utils

{

// Fields

private:

List<Field^>^ fields;

// Methods

public:

Utils(); // default constructor - creates an empty object

~Utils(); // destructor - deletes

void Download();

void Insert();

void Insert(Field^);

void Update(Field^);

void Update();

void Delete(Field^);

void Delete();

};

***Utils.cpp***

#include "Utils.h"

Utils::Utils()

{

fields = gcnew List<Field^>();

}

Utils::~Utils()

{

fields->Clear();

}

void Utils::Download()

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb"; // connection string

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open(); // opening the connection

String^ query = "SELECT \* FROM [table\_name]";

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

OleDbDataReader^ dbReader = dbComand->ExecuteReader(); // reading the data

// Checking the data

if (dbReader->HasRows == false) {

MessageBox::Show("Data reading error!", "Error!");

}

else {

fields->Clear();

// We do the necessary actions with the data

while (dbReader->Read()) {

// For example:

Field^ buf = gcnew Field();

buf->id = Convert::ToInt32(dbReader["id"]);

buf->name = Convert::ToString(dbReader["Name"]);

fields->Add(buf);

}

}

// Closing the connection

dbReader->Close();

dbConnection->Close();

}

catch (OleDbException^ e) {

MessageBox::Show(e->ToString(), "Error!");

}

}

void Utils::Insert()

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb"; // connection string

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open(); // opening the connection

// Insert all data

bool err = false;

for (int i = 0; i < fields->Count; i++) {

String^ query = "INSERT INTO [table\_name] VALUES (" + fields[i]->id + ", '" + fields[i]->name + "')";

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

// Executing the request

if (dbComand->ExecuteNonQuery() != 1)

err = true;

}

if (err) {

MessageBox::Show("Error inserting!", "Error!");

}

else {

MessageBox::Show("All data inserting!!", "Ошибка!");

}

// Closing the connection

dbConnection->Close();

}

catch (OleDbException^ e) {

MessageBox::Show(e->ToString(), "Error!");

}

}

void Utils::Insert(Field^ f)

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb"; // connection string

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open(); // opening the connection

String^ query = "INSERT INTO [table\_name] VALUES (" + f->id + ", '" + f->name + "')";

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

// Executing the request

if (dbComand->ExecuteNonQuery() != 1) {

MessageBox::Show("Error inserting!", "Error!");

}

else {

MessageBox::Show("All data inserting!!", "Ошибка!");

}

// Closing the connection

dbConnection->Close();

}

catch (OleDbException^ e) {

MessageBox::Show(e->ToString(), "Error!");

}

}

void Utils::Update(Field^ f)

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open();

String^ query = "UPDATE [table\_name] SET Name='" + f->name + " WHERE id = " + f->id;

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

// Executing the request

if (dbComand->ExecuteNonQuery() != 1)

MessageBox::Show("Updating error!", "Error!");

else

MessageBox::Show("Data update!", "Done!");

// Closing the connection to the database

dbConnection->Close();

}

catch (OleDbException^ e) {

MessageBox::Show(e->ToString(), "Error!");

}

}

void Utils::Update()

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open();

// Update all data

bool err = false;

for each (Field ^ f in fields)

{

String^ query = "UPDATE [table\_name] SET Name='" + f->name + " WHERE id = " + f->id;

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

// Executing the request

if (dbComand->ExecuteNonQuery() != 1)

err = true;

}

if (err)

MessageBox::Show("Updating error!", "Error!");

else

MessageBox::Show("Data update!", "Done!");

// Closing the connection to the database

dbConnection->Close();

}

catch (OleDbException^ e) {

MessageBox::Show(e->ToString(), "Error!");

}

}

void Utils::Delete(Field^ f)

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open();

String^ query = "DELETE FROM [table\_name] WHERE id = " + f->id;

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

// Executing the request

if (dbComand->ExecuteNonQuery() != 1)

MessageBox::Show("Deletion error!", "Error!");

else {

MessageBox::Show("Data deleted!", "Done!");

}

// Closing the connection

dbConnection->Close();

}

catch (OleDbException^ e) {

MessageBox::Show(e->ToString(), "Error!");

}

}

void Utils::Delete()

{

try {

// Connecting to the database

String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data Source=Database.mdb";

OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString);

// Executing a query to the database

dbConnection->Open();

String^ query = "DELETE FROM [table\_name]"; // clear table

OleDbCommand^ dbComand = gcnew OleDbCommand(query, dbConnection);

// Executing the request

if (dbComand->ExecuteNonQuery() != 1)

MessageBox::Show("Deletion error!", "Error!");

else {

MessageBox::Show("Data deleted!", "Done!");

}

// Closing the connection

dbConnection->Close();

}

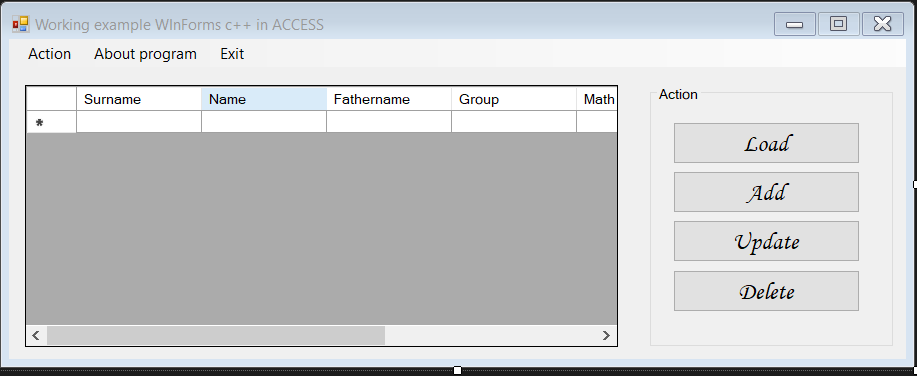
catch (OleDbException^ e) {

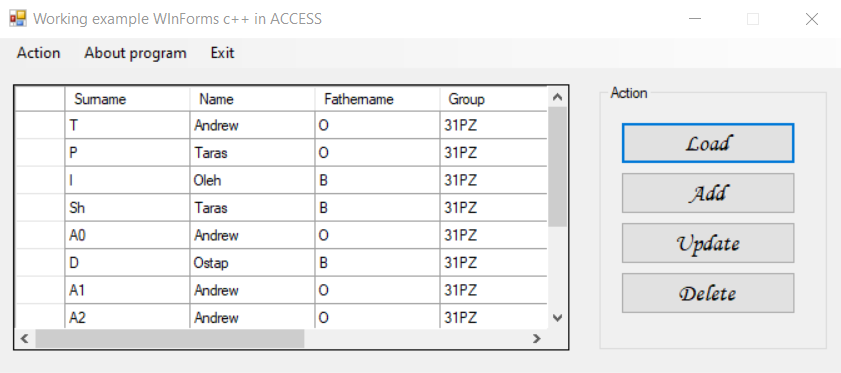
MessageBox::Show(e->ToString(), "Error!");

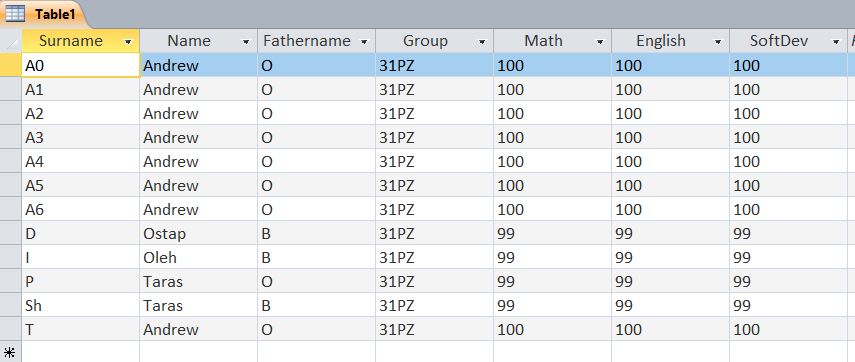
}

}

***Скрін-шоти роботи програми:***

******

******

******

**Висновок**

Отже, під час виконання лабораторної роботи №6, я ознайомився з ключовими поняттями, методами та засобами платформи .Net та технологій Windows Forms та ADO.Net.