**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Технічний коледж**

**НУ «Львівська політехніка»**

**Відділення**

**Інформаційних технологій**

**та комп’ютерної техніки**

**Звіт до:**

Лабораторної роботи № 6

з предмету «НП з ООП»

**Тема:**

«**Створення програм з GUI**»

**Підготував**

Студент групи – 31-ПЗ

Дячок Остап

**Львів 2021**

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

**Постановка задачі:**

**Варіант 8**

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.



**Код :**

Main.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["Bar code"], dbReader["Name"], dbReader["Weight(kg)"], dbReader["Cost($)"],dbReader["Count"]);

}

}

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.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!");

}

}

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^);

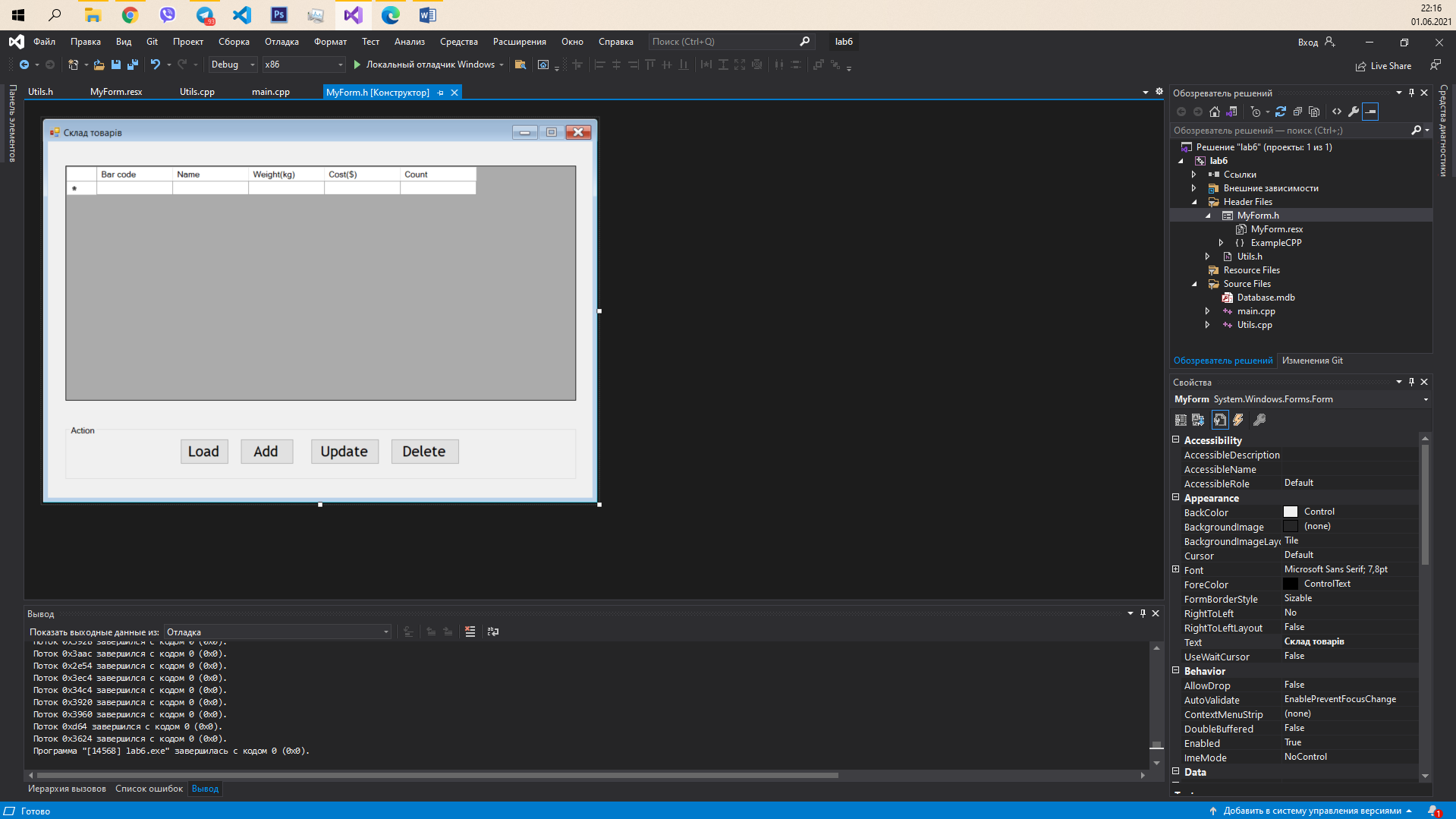
void Update(Field^);

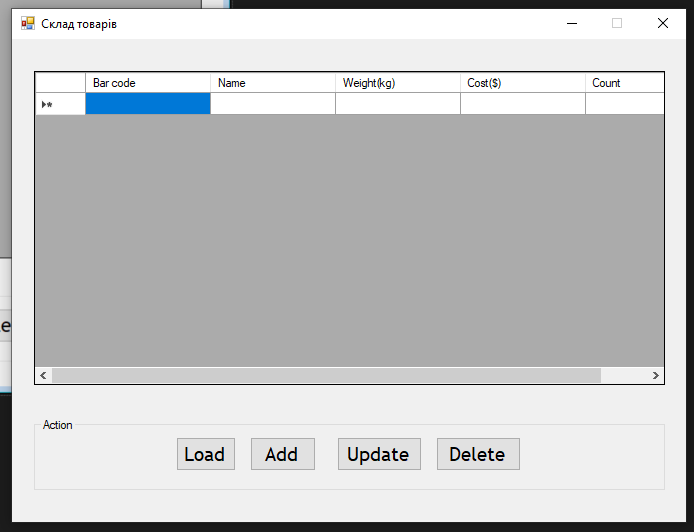
void Update();

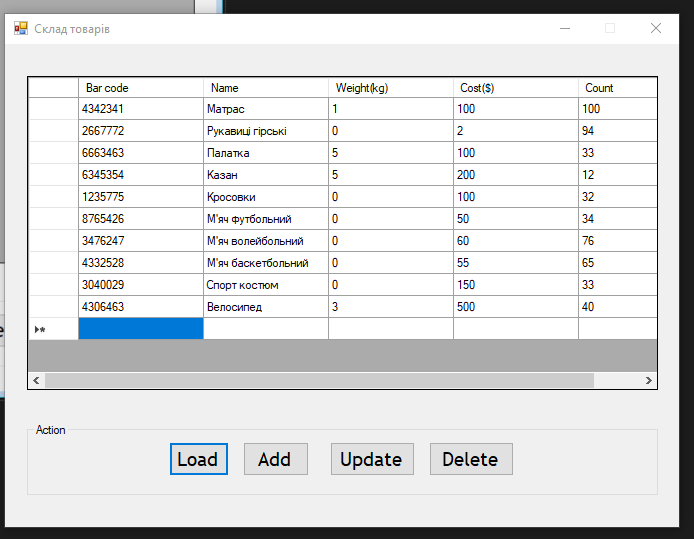
void Delete(Field^);

void Delete();

};







**Висновок:**

Завдяки виконанню лабораторної роботи 6 я ознайомився з ключовими поняттями, методами та засобами платформи .Net та технологіями Windows Forms та ADO.Net.