МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ

НАДВІРНЯНСЬКИЙ КОЛЕДЖ НТУ

**ЗВІТ  
З НАВЧАЛЬНОЇ ПРАКТИКИ**

**Розробка програми «Інтернет магазин»**

Виконав студент групи ІТ-31:

Оцінка:\_\_\_\_\_\_\_\_\_\_\_\_\_ Пицюк Володимир

Прийняв: Ілько М.В.

**Надвірна**

**2017**

**Зміст**

[Вступ 3](#_Toc501057501)

[1. Загальна структура і вимоги програми 4](#_Toc501057502)

[2. Опис методів і класів 6](#_Toc501057503)

[3. Результати роботи програми 7](#_Toc501057504)

[Висновки 8](#_Toc501057505)

[Перелік використаних джерел 9](#_Toc501057506)

## Вступ

Сьогодні світ переживає нову могутню революцію, здатну принципово змінити життя людей — їх роботу, дозвілля, способи об'єднання в співтовариства і навіть ставлення до са­мих себе. На відміну від попередніх технологічних рево­люцій, заснованих на матерії і енергії, ця фундаментальна зміна зачіпає наше розуміння часу, простору, відстані і знан­ня. В основі інформаційної революції лежить вибуховий роз­виток інформаційних технологій (IT), різноманітність і можливість вживання яких лімітовані лише винахідливістю самої людини. Сьогодні стала очевидною перевага інфор­маційної складової діяльності людей над всіма іншими її формами і компонентами. У сучасному світі знання й інформація породжують нові знання, їхні обсяги і вплив на продуктивний розвиток суспільства зростають у часі подібно сніжному кому. Цей виклик жадає від людства нових способів і засобів поширення і використання глобальних знань з метою подальшого прогресу, що і є головною властивістю суспільства знань і інформації. Пошук нової парадигми улаштування світу, що вимагає глобалізації знань і наукових досягнень, і став передумовою виникнення нової фази розвитку людства, пов'язаної з появою інформаційного суспільства і його найбільш розвиненої форми — суспільства знань.

*[База даних](http://lib.mdpu.org.ua/e-book/vstup/glos.htm" \l "База даних)* (БД) - упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД.

Бібліотеки візуальних компонентів (VCL) – це об’єктно орієнтовна бібліотека для розробки програмного забезпечення, розроблена компанією Borland. VCL представляє собою велику кількість готових до використання компонентів для роботи в самих різних областях програмування і значно

Метою навчальної практики з ООП є навчитись створювати програми з базами даних за допомогою Windows Forms ні мові програмування C++.

## 1. Загальна структура і вимоги програми

**Загальна структура програми**

**Програма для адміністрації бази даних –** це програма, за допомогою якої реалізується можливість маніпулювати набором даних: змінювати, видаляти, добавляти, сортувати, а також пошуку даних. В нашому випадку – це база даних **Microsoft Access.**

**Ця программа реалізує такий функціонал:**

1. Відтворення даних в таблиці
2. Редагування даних
3. Видалення даних
4. Створення нових записів
5. Пошук

**Вимоги до апаратного забезпечення.**

Дана програма призначена для редагування тексту.

Системні вимоги:

Операційні системи:

Windows XP (SP2)

Windows 7 з пакетом оновлень 1 (SP1) (x86 і x64)

Windows 8 (x86 і x64)

Windows 8,1 (x86 і x64)

Windows 10 (x86 і x64)

Windows Server 2008 R2 з пакетом оновлень 1 (SP1) (x64)

Windows Server 2012 (x64)

Windows Server 2012 R2 (x64)

Вимоги до обладнання:

Процесор з частотою 1,6 ГГц або більш потужніший.

1 ГБ ОЗУ (1,5 ГБ при використанні віртуальної машини).

4 ГБ вільного дискового простору

Жорсткий диск (5400 об / хв)

Відеоадаптер з підтримкою DirectX 9; мінімально допустима здатність екрану - 1024 x 768

Наявність .NET Framework 4.1

Дану програму компілювали з використанням Visual Studio 2015.

## 2. Опис методів і класів

В програмі використовується namespace PracticWork, головим класом є MyForm, а точкою входу для програми – функція main().

**Кнопки «Зберегти»**

Реалізована за допомогою елемента Button , при нажаті на який виконується подія button\_Click яка за допомогою методу Update зберігає змінені дані користувачем до таблиці.

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

oleDbDataAdapter1->Update(dataTable1);

}

**Кнопки «Пошук»**

Реалізована за допомогою елемента Button , при нажаті на який виконується подія button\_Click яка здійснює пошук в тексті.

## 3. Результати роботи програми

При першому запуску програми користувачу висвітлюється список всіх товарів, форма для редагування.

# Висновки

Під час практики було розроблено програму яка зберігає дані про інтернет магазини та товани в них, програма була виконана за допомогою CLR –компонентів. CLR -компоненти дали змогу значно скоротити код програми, що в свою чергу спростило розробку програми. В програмі було використано такі компоненти TextBox, label,PictureBox.

Вихідний код програми можна знайти на відритому репозиторію на github - <https://git.io/vbwmT>

# Перелік використаних джерел

* 1. MS Visual C++ 2010 в среде .NET
  2. Т. В. Ковалюк «Основи програмування»
  3. В.П. Аверкин, А.И. Бобровський, В. В. Веснич, В. Ф. Радушинский, А. Д. Хомоненко «Программирование на С++»
  4. «Искуство програмирования на С++»
  5. Зиборов В. В. – MS Visual C++ 2010 в [среде.NET](https://vk.com/away.php?to=http%3A%2F%2F%F1%F0%E5%E4%E5.NET). Библиотека программиста. — СПб.: Питер, 2012. — 320 с.: ил.
  6. Кравець П.О. Об'єктно-орієнтоване програмування: навч. посібник / П.О. Кравець. - Львів: Видавництво Львівської політехніки, 2012. - 624 с.
  7. Грицюк Ю.І., Рак Т.Є. Г 85 Програмування мовою С++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с.
  8. Сидорина Т. Самоучитель Microsoft Visual Studio C++ и MFC (+ CD-ROM). – СпБ: BHV, 2009. – 848 c.
  9. Давыдов В. Visual C++. Разработка Windows-приложений с помощью MFC и API-функций (+ CD-ROM). – СпБ: BHV, 2008. –576 c.
  10. Г. Шилдт. С++. Базовый курс. 3-е изд. М: Вильямс. – 2005, 624 с.
  11. Олафсен Ю., MFC и Visual C++ 6. Энциклопедия программиста (+ CD) / Ю. Олафсен, К. Скрайбнер, К. Дэвид Уайт. – М.: DiaSoft, 2004. – 992 с.
  12. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Обьектно-ориентированное програмирование – М.: МГТУ им. Н.Е. Баумана, 2001.-320с.
  13. Н. Культин. С/С++ в задачах и примерах. С.-Пб.: BHV. – 2005, 292 с.

**Додатки**

**Додаток 1**

Код програми

#pragma once

namespace AirCraft {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

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

{

public:

MyForm(void)

{

InitializeComponent();

dbPath = Application::StartupPath;

oleDbConnection1->ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data Source =" + dbPath + "\\aircraft.accdb";

oleDbConnection2->ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data Source =" + dbPath + "\\aircraft.accdb";

imFolder = dbPath + "\\images\\";

oleDbDataAdapter1->Fill(dataTable1);

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Data::OleDb::OleDbCommand^ oleDbSelectCommand1;

protected:

private: String^ imFolder; String^ dbPath;

…. //COMPONENTS

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Требуется переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Обязательный метод для поддержки конструктора - не изменяйте

/// содержимое данного метода при помощи редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));

this->oleDbConnection1 = (gcnew System::Data::OleDb::OleDbConnection());

this->oleDbSelectCommand1 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbInsertCommand1 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbDeleteCommand1 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbDataAdapter1 = (gcnew System::Data::OleDb::OleDbDataAdapter());

this->oleDbCommand1 = (gcnew System::Data::OleDb::OleDbCommand());

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

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

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

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

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

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

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

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

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

this->bindingSource1 = (gcnew System::Windows::Forms::BindingSource(this->components));

this->dataSet1 = (gcnew System::Data::DataSet());

this->dataTable1 = (gcnew System::Data::DataTable());

this->dataColumn1 = (gcnew System::Data::DataColumn());

this->dataColumn2 = (gcnew System::Data::DataColumn());

this->dataColumn3 = (gcnew System::Data::DataColumn());

this->dataColumn4 = (gcnew System::Data::DataColumn());

this->dataColumn5 = (gcnew System::Data::DataColumn());

this->dataColumn6 = (gcnew System::Data::DataColumn());

this->dataColumn7 = (gcnew System::Data::DataColumn());

this->dataColumn8 = (gcnew System::Data::DataColumn());

this->bindingNavigator1 = (gcnew System::Windows::Forms::BindingNavigator(this->components));

this->bindingNavigatorAddNewItem = (gcnew System::Windows::Forms::ToolStripButton());

this->bindingNavigatorCountItem = (gcnew System::Windows::Forms::ToolStripLabel());

this->bindingNavigatorDeleteItem = (gcnew System::Windows::Forms::ToolStripButton());

this->bindingNavigatorMoveFirstItem = (gcnew System::Windows::Forms::ToolStripButton());

this->bindingNavigatorMovePreviousItem = (gcnew System::Windows::Forms::ToolStripButton());

this->bindingNavigatorSeparator = (gcnew System::Windows::Forms::ToolStripSeparator());

this->bindingNavigatorPositionItem = (gcnew System::Windows::Forms::ToolStripTextBox());

this->bindingNavigatorSeparator1 = (gcnew System::Windows::Forms::ToolStripSeparator());

this->bindingNavigatorMoveNextItem = (gcnew System::Windows::Forms::ToolStripButton());

this->bindingNavigatorMoveLastItem = (gcnew System::Windows::Forms::ToolStripButton());

this->bindingNavigatorSeparator2 = (gcnew System::Windows::Forms::ToolStripSeparator());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->textBox4 = (gcnew System::Windows::Forms::TextBox());

this->textBox5 = (gcnew System::Windows::Forms::TextBox());

this->textBox6 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->textBox8 = (gcnew System::Windows::Forms::TextBox());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());

this->button1 = (gcnew System::Windows::Forms::Button());

this->oleDbSelectCommand2 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbConnection2 = (gcnew System::Data::OleDb::OleDbConnection());

this->oleDbInsertCommand2 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbUpdateCommand1 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbDeleteCommand2 = (gcnew System::Data::OleDb::OleDbCommand());

this->oleDbDataAdapter2 = (gcnew System::Data::OleDb::OleDbDataAdapter());

this->button2 = (gcnew System::Windows::Forms::Button());

this->textBox7 = (gcnew System::Windows::Forms::TextBox());

this->label7 = (gcnew System::Windows::Forms::Label());

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

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->bindingSource1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataSet1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataTable1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->bindingNavigator1))->BeginInit();

this->bindingNavigator1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

this->SuspendLayout();

//

// oleDbConnection1

//

this->oleDbConnection1->ConnectionString = L"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\"C:\\Users\\akidse\\Documents\\Visual S"

L"tudio 2013\\Projects\\AirCraft\\Debug\\aircraft.accdb\"";

//

// oleDbSelectCommand1

//

this->oleDbSelectCommand1->CommandText = L"SELECT aircraft.\* FROM aircraft";

this->oleDbSelectCommand1->Connection = this->oleDbConnection1;

//

// oleDbInsertCommand1

//

this->oleDbInsertCommand1->CommandText = L"INSERT INTO `aircraft` (`type`, `company`, `model`, `code\_name`, `country`, `year"

L"`, `image`) VALUES (\?, \?, \?, \?, \?, \?, \?)";

this->oleDbInsertCommand1->Connection = this->oleDbConnection1;

this->oleDbInsertCommand1->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(7) {

(gcnew System::Data::OleDb::OleDbParameter(L"type",

System::Data::OleDb::OleDbType::VarWChar, 0, L"type")), (gcnew System::Data::OleDb::OleDbParameter(L"company", System::Data::OleDb::OleDbType::VarWChar,

0, L"company")), (gcnew System::Data::OleDb::OleDbParameter(L"model", System::Data::OleDb::OleDbType::VarWChar, 0, L"model")),

(gcnew System::Data::OleDb::OleDbParameter(L"code\_name", System::Data::OleDb::OleDbType::VarWChar, 0, L"code\_name")), (gcnew System::Data::OleDb::OleDbParameter(L"country",

System::Data::OleDb::OleDbType::VarWChar, 0, L"country")), (gcnew System::Data::OleDb::OleDbParameter(L"year", System::Data::OleDb::OleDbType::Integer,

0, L"year")), (gcnew System::Data::OleDb::OleDbParameter(L"image", System::Data::OleDb::OleDbType::VarWChar, 0, L"image"))

});

//

// oleDbDeleteCommand1

//

this->oleDbDeleteCommand1->CommandText = L"DELETE FROM aircraft WHERE (id = \?)";

this->oleDbDeleteCommand1->Connection = this->oleDbConnection1;

this->oleDbDeleteCommand1->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(1) {

(gcnew System::Data::OleDb::OleDbParameter(L"id",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"id", System::Data::DataRowVersion::Original, nullptr))

});

//

// oleDbDataAdapter1

//

this->oleDbDataAdapter1->DeleteCommand = this->oleDbDeleteCommand1;

this->oleDbDataAdapter1->InsertCommand = this->oleDbInsertCommand1;

this->oleDbDataAdapter1->SelectCommand = this->oleDbSelectCommand1;

cli::array< System::Data::Common::DataColumnMapping^ >^ \_\_mcTemp\_\_1 = gcnew cli::array< System::Data::Common::DataColumnMapping^ >(8) {

(gcnew System::Data::Common::DataColumnMapping(L"id",

L"id")), (gcnew System::Data::Common::DataColumnMapping(L"type", L"type")), (gcnew System::Data::Common::DataColumnMapping(L"company",

L"company")), (gcnew System::Data::Common::DataColumnMapping(L"model", L"model")), (gcnew System::Data::Common::DataColumnMapping(L"code\_name",

L"code\_name")), (gcnew System::Data::Common::DataColumnMapping(L"country", L"country")), (gcnew System::Data::Common::DataColumnMapping(L"year",

L"year")), (gcnew System::Data::Common::DataColumnMapping(L"image", L"image"))

};

this->oleDbDataAdapter1->TableMappings->AddRange(gcnew cli::array< System::Data::Common::DataTableMapping^ >(1) {

(gcnew System::Data::Common::DataTableMapping(L"Table",

L"aircraft", \_\_mcTemp\_\_1))

});

this->oleDbDataAdapter1->UpdateCommand = this->oleDbCommand1;

this->oleDbDataAdapter1->RowUpdated += gcnew System::Data::OleDb::OleDbRowUpdatedEventHandler(this, &MyForm::oleDbDataAdapter1\_RowUpdated);

this->oleDbDataAdapter1->RowUpdating += gcnew System::Data::OleDb::OleDbRowUpdatingEventHandler(this, &MyForm::oleDbDataAdapter1\_RowUpdating);

this->oleDbDataAdapter1->FillError += gcnew System::Data::FillErrorEventHandler(this, &MyForm::oleDbDataAdapter1\_FillError);

//

// oleDbCommand1

//

this->oleDbCommand1->CommandText = L"UPDATE aircraft\r\nSET type = \?, company = \?, model = \?, code\_"

L"name = \?, country = \?, [year] = \?, [image] = \?\r\nWHERE (id = \?)";

this->oleDbCommand1->Connection = this->oleDbConnection1;

this->oleDbCommand1->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(8) {

(gcnew System::Data::OleDb::OleDbParameter(L"type",

System::Data::OleDb::OleDbType::WChar, 255, L"type")), (gcnew System::Data::OleDb::OleDbParameter(L"company", System::Data::OleDb::OleDbType::WChar,

255, L"company")), (gcnew System::Data::OleDb::OleDbParameter(L"model", System::Data::OleDb::OleDbType::WChar, 255, L"model")),

(gcnew System::Data::OleDb::OleDbParameter(L"code\_name", System::Data::OleDb::OleDbType::WChar, 255, L"code\_name")), (gcnew System::Data::OleDb::OleDbParameter(L"country",

System::Data::OleDb::OleDbType::WChar, 255, L"country")), (gcnew System::Data::OleDb::OleDbParameter(L"year", System::Data::OleDb::OleDbType::Integer,

0, L"year")), (gcnew System::Data::OleDb::OleDbParameter(L"image", System::Data::OleDb::OleDbType::WChar, 255, L"image")), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_id",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"id", System::Data::DataRowVersion::Original, nullptr))

});

//

// dataGridView1

//

this->dataGridView1->AutoGenerateColumns = false;

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

this->dataGridView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(8) {

this->idDataGridViewTextBoxColumn,

this->typeDataGridViewTextBoxColumn, this->companyDataGridViewTextBoxColumn, this->modelDataGridViewTextBoxColumn, this->codenameDataGridViewTextBoxColumn,

this->countryDataGridViewTextBoxColumn, this->yearDataGridViewTextBoxColumn, this->imageDataGridViewTextBoxColumn

});

this->dataGridView1->DataSource = this->bindingSource1;

this->dataGridView1->EditMode = System::Windows::Forms::DataGridViewEditMode::EditOnEnter;

this->dataGridView1->Location = System::Drawing::Point(12, 272);

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

this->dataGridView1->Size = System::Drawing::Size(725, 211);

this->dataGridView1->TabIndex = 0;

this->dataGridView1->CellEndEdit += gcnew System::Windows::Forms::DataGridViewCellEventHandler(this, &MyForm::dataGridView1\_CellEndEdit);

//

// idDataGridViewTextBoxColumn

//

this->idDataGridViewTextBoxColumn->DataPropertyName = L"id";

this->idDataGridViewTextBoxColumn->HeaderText = L"id";

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

//

// typeDataGridViewTextBoxColumn

//

this->typeDataGridViewTextBoxColumn->DataPropertyName = L"type";

this->typeDataGridViewTextBoxColumn->HeaderText = L"type";

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

//

// companyDataGridViewTextBoxColumn

//

this->companyDataGridViewTextBoxColumn->DataPropertyName = L"company";

this->companyDataGridViewTextBoxColumn->HeaderText = L"company";

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

//

// modelDataGridViewTextBoxColumn

//

this->modelDataGridViewTextBoxColumn->DataPropertyName = L"model";

this->modelDataGridViewTextBoxColumn->HeaderText = L"model";

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

//

// codenameDataGridViewTextBoxColumn

//

this->codenameDataGridViewTextBoxColumn->DataPropertyName = L"code\_name";

this->codenameDataGridViewTextBoxColumn->HeaderText = L"code\_name";

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

//

// countryDataGridViewTextBoxColumn

//

this->countryDataGridViewTextBoxColumn->DataPropertyName = L"country";

this->countryDataGridViewTextBoxColumn->HeaderText = L"country";

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

//

// yearDataGridViewTextBoxColumn

//

this->yearDataGridViewTextBoxColumn->DataPropertyName = L"year";

this->yearDataGridViewTextBoxColumn->HeaderText = L"year";

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

//

// imageDataGridViewTextBoxColumn

//

this->imageDataGridViewTextBoxColumn->DataPropertyName = L"image";

this->imageDataGridViewTextBoxColumn->HeaderText = L"image";

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

//

// bindingSource1

//

this->bindingSource1->DataMember = L"aircraft";

this->bindingSource1->DataSource = this->dataSet1;

//

// dataSet1

//

this->dataSet1->DataSetName = L"NewDataSet";

this->dataSet1->Tables->AddRange(gcnew cli::array< System::Data::DataTable^ >(1) { this->dataTable1 });

//

// dataTable1

//

this->dataTable1->Columns->AddRange(gcnew cli::array< System::Data::DataColumn^ >(8) {

this->dataColumn1, this->dataColumn2,

this->dataColumn3, this->dataColumn4, this->dataColumn5, this->dataColumn6, this->dataColumn7, this->dataColumn8

});

this->dataTable1->TableName = L"aircraft";

//

// dataColumn1

//

this->dataColumn1->ColumnName = L"id";

this->dataColumn1->DataType = System::Int32::typeid;

//

// dataColumn2

//

this->dataColumn2->ColumnName = L"type";

//

// dataColumn3

//

this->dataColumn3->ColumnName = L"company";

//

// dataColumn4

//

this->dataColumn4->ColumnName = L"model";

//

// dataColumn5

//

this->dataColumn5->ColumnName = L"code\_name";

//

// dataColumn6

//

this->dataColumn6->ColumnName = L"country";

//

// dataColumn7

//

this->dataColumn7->ColumnName = L"year";

this->dataColumn7->DataType = System::Int32::typeid;

//

// dataColumn8

//

this->dataColumn8->ColumnName = L"image";

//

// bindingNavigator1

//

this->bindingNavigator1->AddNewItem = this->bindingNavigatorAddNewItem;

this->bindingNavigator1->BindingSource = this->bindingSource1;

this->bindingNavigator1->CountItem = this->bindingNavigatorCountItem;

this->bindingNavigator1->DeleteItem = this->bindingNavigatorDeleteItem;

this->bindingNavigator1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(11) {

this->bindingNavigatorMoveFirstItem,

this->bindingNavigatorMovePreviousItem, this->bindingNavigatorSeparator, this->bindingNavigatorPositionItem, this->bindingNavigatorCountItem,

this->bindingNavigatorSeparator1, this->bindingNavigatorMoveNextItem, this->bindingNavigatorMoveLastItem, this->bindingNavigatorSeparator2,

this->bindingNavigatorAddNewItem, this->bindingNavigatorDeleteItem

});

this->bindingNavigator1->Location = System::Drawing::Point(0, 0);

this->bindingNavigator1->MoveFirstItem = this->bindingNavigatorMoveFirstItem;

this->bindingNavigator1->MoveLastItem = this->bindingNavigatorMoveLastItem;

this->bindingNavigator1->MoveNextItem = this->bindingNavigatorMoveNextItem;

this->bindingNavigator1->MovePreviousItem = this->bindingNavigatorMovePreviousItem;

this->bindingNavigator1->Name = L"bindingNavigator1";

this->bindingNavigator1->PositionItem = this->bindingNavigatorPositionItem;

this->bindingNavigator1->Size = System::Drawing::Size(748, 25);

this->bindingNavigator1->TabIndex = 1;

this->bindingNavigator1->Text = L"bindingNavigator1";

//

// bindingNavigatorAddNewItem

//

this->bindingNavigatorAddNewItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;

this->bindingNavigatorAddNewItem->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorAddNewItem.Image")));

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

this->bindingNavigatorAddNewItem->RightToLeftAutoMirrorImage = true;

this->bindingNavigatorAddNewItem->Size = System::Drawing::Size(23, 22);

this->bindingNavigatorAddNewItem->Text = L"Add new";

//

// bindingNavigatorCountItem

//

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

this->bindingNavigatorCountItem->Size = System::Drawing::Size(35, 22);

this->bindingNavigatorCountItem->Text = L"of {0}";

this->bindingNavigatorCountItem->ToolTipText = L"Total number of items";

//

// bindingNavigatorDeleteItem

//

this->bindingNavigatorDeleteItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;

this->bindingNavigatorDeleteItem->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorDeleteItem.Image")));

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

this->bindingNavigatorDeleteItem->RightToLeftAutoMirrorImage = true;

this->bindingNavigatorDeleteItem->Size = System::Drawing::Size(23, 22);

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

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

//

// bindingNavigatorMoveFirstItem

//

this->bindingNavigatorMoveFirstItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;

this->bindingNavigatorMoveFirstItem->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMoveFirstItem.Image")));

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

this->bindingNavigatorMoveFirstItem->RightToLeftAutoMirrorImage = true;

this->bindingNavigatorMoveFirstItem->Size = System::Drawing::Size(23, 22);

this->bindingNavigatorMoveFirstItem->Text = L"Move first";

//

// bindingNavigatorMovePreviousItem

//

this->bindingNavigatorMovePreviousItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;

this->bindingNavigatorMovePreviousItem->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMovePreviousItem.Image")));

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

this->bindingNavigatorMovePreviousItem->RightToLeftAutoMirrorImage = true;

this->bindingNavigatorMovePreviousItem->Size = System::Drawing::Size(23, 22);

this->bindingNavigatorMovePreviousItem->Text = L"Move previous";

//

// bindingNavigatorSeparator

//

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

this->bindingNavigatorSeparator->Size = System::Drawing::Size(6, 25);

//

// bindingNavigatorPositionItem

//

this->bindingNavigatorPositionItem->AccessibleName = L"Position";

this->bindingNavigatorPositionItem->AutoSize = false;

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

this->bindingNavigatorPositionItem->Size = System::Drawing::Size(50, 23);

this->bindingNavigatorPositionItem->Text = L"0";

this->bindingNavigatorPositionItem->ToolTipText = L"Current position";

//

// bindingNavigatorSeparator1

//

this->bindingNavigatorSeparator1->Name = L"bindingNavigatorSeparator1";

this->bindingNavigatorSeparator1->Size = System::Drawing::Size(6, 25);

//

// bindingNavigatorMoveNextItem

//

this->bindingNavigatorMoveNextItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;

this->bindingNavigatorMoveNextItem->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMoveNextItem.Image")));

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

this->bindingNavigatorMoveNextItem->RightToLeftAutoMirrorImage = true;

this->bindingNavigatorMoveNextItem->Size = System::Drawing::Size(23, 22);

this->bindingNavigatorMoveNextItem->Text = L"Move next";

//

// bindingNavigatorMoveLastItem

//

this->bindingNavigatorMoveLastItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;

this->bindingNavigatorMoveLastItem->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMoveLastItem.Image")));

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

this->bindingNavigatorMoveLastItem->RightToLeftAutoMirrorImage = true;

this->bindingNavigatorMoveLastItem->Size = System::Drawing::Size(23, 22);

this->bindingNavigatorMoveLastItem->Text = L"Move last";

//

// bindingNavigatorSeparator2

//

this->bindingNavigatorSeparator2->Name = L"bindingNavigatorSeparator2";

this->bindingNavigatorSeparator2->Size = System::Drawing::Size(6, 25);

//

// textBox1

//

….//BINDINGBUTTONS FOR NAVIGATORSEPARATOR2

//

// oleDbSelectCommand2

//

this->oleDbSelectCommand2->CommandText = L"SELECT \* FROM aircraft WHERE country LIKE \?";

this->oleDbSelectCommand2->Connection = this->oleDbConnection2;

this->oleDbSelectCommand2->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(1) {

(gcnew System::Data::OleDb::OleDbParameter(L"country",

System::Data::OleDb::OleDbType::WChar, 255, L"country"))

});

//

// oleDbConnection2

//

this->oleDbConnection2->ConnectionString = L"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\"C:\\Users\\akidse\\Documents\\Visual S"

L"tudio 2013\\Projects\\AirCraft\\AirCraft\\aircraft.accdb\"";

//

// oleDbInsertCommand2

//

this->oleDbInsertCommand2->CommandText = L"INSERT INTO `aircraft` (`type`, `company`, `model`, `code\_name`, `country`, `year"

L"`, `image`) VALUES (\?, \?, \?, \?, \?, \?, \?)";

this->oleDbInsertCommand2->Connection = this->oleDbConnection2;

this->oleDbInsertCommand2->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(7) {

(gcnew System::Data::OleDb::OleDbParameter(L"type",

System::Data::OleDb::OleDbType::VarWChar, 0, L"type")), (gcnew System::Data::OleDb::OleDbParameter(L"company", System::Data::OleDb::OleDbType::VarWChar,

0, L"company")), (gcnew System::Data::OleDb::OleDbParameter(L"model", System::Data::OleDb::OleDbType::VarWChar, 0, L"model")),

(gcnew System::Data::OleDb::OleDbParameter(L"code\_name", System::Data::OleDb::OleDbType::VarWChar, 0, L"code\_name")), (gcnew System::Data::OleDb::OleDbParameter(L"country",

System::Data::OleDb::OleDbType::VarWChar, 0, L"country")), (gcnew System::Data::OleDb::OleDbParameter(L"year", System::Data::OleDb::OleDbType::Integer,

0, L"year")), (gcnew System::Data::OleDb::OleDbParameter(L"image", System::Data::OleDb::OleDbType::VarWChar, 0, L"image"))

});

//

// oleDbUpdateCommand1

//

this->oleDbUpdateCommand1->CommandText = resources->GetString(L"oleDbUpdateCommand1.CommandText");

this->oleDbUpdateCommand1->Connection = this->oleDbConnection2;

this->oleDbUpdateCommand1->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(22) {

(gcnew System::Data::OleDb::OleDbParameter(L"type",

System::Data::OleDb::OleDbType::VarWChar, 0, L"type")), (gcnew System::Data::OleDb::OleDbParameter(L"company", System::Data::OleDb::OleDbType::VarWChar,

0, L"company")), (gcnew System::Data::OleDb::OleDbParameter(L"model", System::Data::OleDb::OleDbType::VarWChar, 0, L"model")),

(gcnew System::Data::OleDb::OleDbParameter(L"code\_name", System::Data::OleDb::OleDbType::VarWChar, 0, L"code\_name")), (gcnew System::Data::OleDb::OleDbParameter(L"country",

System::Data::OleDb::OleDbType::VarWChar, 0, L"country")), (gcnew System::Data::OleDb::OleDbParameter(L"year", System::Data::OleDb::OleDbType::Integer,

0, L"year")), (gcnew System::Data::OleDb::OleDbParameter(L"image", System::Data::OleDb::OleDbType::VarWChar, 0, L"image")), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_id",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"id", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_type",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"type", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_type",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"type", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_company",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"company", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_company",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"company", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_model",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"model", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_model",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"model", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_code\_name",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"code\_name", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_code\_name",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"code\_name", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_country",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"country", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_country",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"country", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_year",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"year", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_year",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"year", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_image",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"image", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_image",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"image", System::Data::DataRowVersion::Original, nullptr))

});

//

// oleDbDeleteCommand2

//

this->oleDbDeleteCommand2->CommandText = resources->GetString(L"oleDbDeleteCommand2.CommandText");

this->oleDbDeleteCommand2->Connection = this->oleDbConnection2;

this->oleDbDeleteCommand2->Parameters->AddRange(gcnew cli::array< System::Data::OleDb::OleDbParameter^ >(15) {

(gcnew System::Data::OleDb::OleDbParameter(L"Original\_id",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"id", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_type",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"type", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_type",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"type", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_company",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"company", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_company",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"company", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_model",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"model", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_model",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"model", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_code\_name",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"code\_name", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_code\_name",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"code\_name", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_country",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"country", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_country",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"country", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_year",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"year", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_year",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"year", System::Data::DataRowVersion::Original, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"IsNull\_image",

System::Data::OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::Input, static\_cast<System::Byte>(0), static\_cast<System::Byte>(0),

L"image", System::Data::DataRowVersion::Original, true, nullptr)), (gcnew System::Data::OleDb::OleDbParameter(L"Original\_image",

System::Data::OleDb::OleDbType::VarWChar, 0, System::Data::ParameterDirection::Input, false, static\_cast<System::Byte>(0),

static\_cast<System::Byte>(0), L"image", System::Data::DataRowVersion::Original, nullptr))

});

//

// oleDbDataAdapter2

//

this->oleDbDataAdapter2->DeleteCommand = this->oleDbDeleteCommand2;

this->oleDbDataAdapter2->InsertCommand = this->oleDbInsertCommand2;

this->oleDbDataAdapter2->SelectCommand = this->oleDbSelectCommand2;

cli::array< System::Data::Common::DataColumnMapping^ >^ \_\_mcTemp\_\_2 = gcnew cli::array< System::Data::Common::DataColumnMapping^ >(8) {

(gcnew System::Data::Common::DataColumnMapping(L"id",

L"id")), (gcnew System::Data::Common::DataColumnMapping(L"type", L"type")), (gcnew System::Data::Common::DataColumnMapping(L"company",

L"company")), (gcnew System::Data::Common::DataColumnMapping(L"model", L"model")), (gcnew System::Data::Common::DataColumnMapping(L"code\_name",

L"code\_name")), (gcnew System::Data::Common::DataColumnMapping(L"country", L"country")), (gcnew System::Data::Common::DataColumnMapping(L"year",

L"year")), (gcnew System::Data::Common::DataColumnMapping(L"image", L"image"))

};

this->oleDbDataAdapter2->TableMappings->AddRange(gcnew cli::array< System::Data::Common::DataTableMapping^ >(1) {

(gcnew System::Data::Common::DataTableMapping(L"Table",

L"aircraft", \_\_mcTemp\_\_2))

});

this->oleDbDataAdapter2->UpdateCommand = this->oleDbUpdateCommand1;

//

// button2

//

this->button2->Location = System::Drawing::Point(239, 227);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(112, 23);

this->button2->TabIndex = 19;

this->button2->Text = L"Шукати";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);

//

// textBox7

//

this->textBox7->Location = System::Drawing::Point(126, 229);

this->textBox7->Name = L"textBox7";

this->textBox7->Size = System::Drawing::Size(100, 20);

this->textBox7->TabIndex = 20;

//

// label7

//

this->label7->AutoSize = true;

this->label7->Location = System::Drawing::Point(9, 232);

this->label7->Name = L"label7";

this->label7->Size = System::Drawing::Size(90, 13);

this->label7->TabIndex = 21;

this->label7->Text = L"Пошук по країні:";

this->label7->Click += gcnew System::EventHandler(this, &MyForm::label7\_Click);

//

// MyForm

//

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

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

this->ClientSize = System::Drawing::Size(748, 514);

this->Controls->Add(this->label7);

this->Controls->Add(this->textBox7);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Controls->Add(this->pictureBox1);

this->Controls->Add(this->textBox8);

this->Controls->Add(this->label6);

this->Controls->Add(this->label5);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox6);

this->Controls->Add(this->textBox5);

this->Controls->Add(this->textBox4);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->bindingNavigator1);

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

this->Name = L"MyForm";

this->Text = L"AirCraft";

this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

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

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->bindingSource1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataSet1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataTable1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->bindingNavigator1))->EndInit();

this->bindingNavigator1->ResumeLayout(false);

this->bindingNavigator1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void MyForm\_FormClosing(System::Object^ sender, System::Windows::Forms::FormClosingEventArgs^ e)

{

oleDbDataAdapter1->Update(dataSet1->Tables["aircraft"]);

}

private: System::Void textBox8\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

String^ imageFile;

String^ msg;

if (textBox8->Text == String::Empty)

{

imageFile = imFolder + "noimage.png";

}

else

imageFile = imFolder + textBox8->Text;

try

{

msg = "";

pictureBox1->Image = System::Drawing::Bitmap::FromFile(imageFile);

}

catch (System::IO::FileNotFoundException^ e)

{

msg = "Файл не знайдено:" + imageFile;

pictureBox1->Image = nullptr;

pictureBox1->Refresh();

}

}

private: System::Void pictureBox1\_Click(System::Object^ sender, System::EventArgs^ e) {

openFileDialog1->Title = "Виберіть ілюстрацію";

openFileDialog1->InitialDirectory = imFolder;

openFileDialog1->Filter = "фото|\*.jpg| всі файли|\*.\*";

openFileDialog1->FileName = "";

if (openFileDialog1->ShowDialog() ==

System::Windows::Forms::DialogResult::OK)

{

bool r = openFileDialog1->FileName->ToLower()->Contains(

openFileDialog1->InitialDirectory->ToLower());

if (r == true)

{

textBox8->Text = openFileDialog1->SafeFileName;

}

else

{

try

{

System::IO::File::Copy(openFileDialog1->FileName,

imFolder + openFileDialog1->SafeFileName);

textBox8->Text = openFileDialog1->SafeFileName;

}

catch (System::Exception^ e)

{

System::Windows::Forms::DialogResult dr;

dr = MessageBox::Show(e->Message +

"Замінити його", "",

MessageBoxButtons::OKCancel,

MessageBoxIcon::Warning,

MessageBoxDefaultButton::Button2);

if (dr == System::Windows::Forms::DialogResult::OK)

{

System::IO::File::Copy(openFileDialog1->FileName,

imFolder + openFileDialog1->SafeFileName,

true);

textBox8->Text = openFileDialog1->SafeFileName;

}

}

}

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

oleDbDataAdapter1->Update(dataSet1);

}

private: System::Void oleDbDataAdapter1\_RowUpdating(System::Object^ sender, System::Data::OleDb::OleDbRowUpdatingEventArgs^ e) {

}

private: System::Void oleDbDataAdapter1\_RowUpdated(System::Object^ sender, System::Data::OleDb::OleDbRowUpdatedEventArgs^ e) {

}

private: System::Void bindingNavigatorDeleteItem\_Click(System::Object^ sender, System::EventArgs^ e) {

oleDbDataAdapter1->Update(dataSet1);

}

private: System::Void dataGridView1\_CellEndEdit(System::Object^ sender, System::Windows::Forms::DataGridViewCellEventArgs^ e) {

oleDbDataAdapter1->Update(dataSet1);

}

private: System::Void oleDbDataAdapter1\_FillError(System::Object^ sender, System::Data::FillErrorEventArgs^ e) {

}

private: System::Void label7\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

dataSet1->Clear();

oleDbDataAdapter2->SelectCommand->Parameters["country"]->Value = "%" + textBox7->Text + "%";

oleDbDataAdapter2->Fill(dataTable1);

}

};

}