Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра вычислительных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №8

по дисциплине конструирование алгоритмов и структур данных

Выполнил: ст. гр. 26/1

Минин К.С.

Проверил: доц. каф. ИТ

Полетайкин А.Н.

Краснодар,

2021

Тема: Формирование клиент-серверной ИС в MS Visual Studio

Цель: Получить практические навыки разработки, тестирования, и построения базовой версии приложения Windows для работы с базой данных.

## Задание

1. Создать в MS Visual Studio проект типа Windows Application.
2. Создать подключение к базе данных, разработанной при выполнении [лабораторной работы №5](#_Лабораторная_работа_№5). Обеспечить видимость соединения во всех формах приложения, а также его переносимость на другие ЭВМ.
3. В главной форме создать основное меню, состоящее из 5 пунктов:

* «Справочники» – ввод справочных данных;
* «Данные» – ввод оперативных данных;
* «Отчеты» – вывод результатов запросов;
* «О программе» – выдача справки и данных о приложении и разработчиках;
* «Выход» – выход из приложения.

1. Создать формы для работы со справочниками. Каждую форму связать с соответствующим пунктом главного меню «Справочники». В каждой форме обеспечить:

* операции с записями;
* сохранение изменений;
* возврат в главное окно.

1. Разработать статическую форму сведений о программе и связать ее с соответствующим пунктом меню.
2. Выполнить описание разработанных форм приложения в виде таблицы 1.

Таблица 1. Перечень разработанных компонентов приложения

|  |  |  |
| --- | --- | --- |
| №  пп | Имя формы | Описание |

**Ход работы**

**1 задание.**

Это задание было выполнить просто, ведь Visual Studio предоставляет много шаблонов (в том числе и приложения Windows Forms, которое я и создал), поэтому в пару кликов я создал пустое приложение.

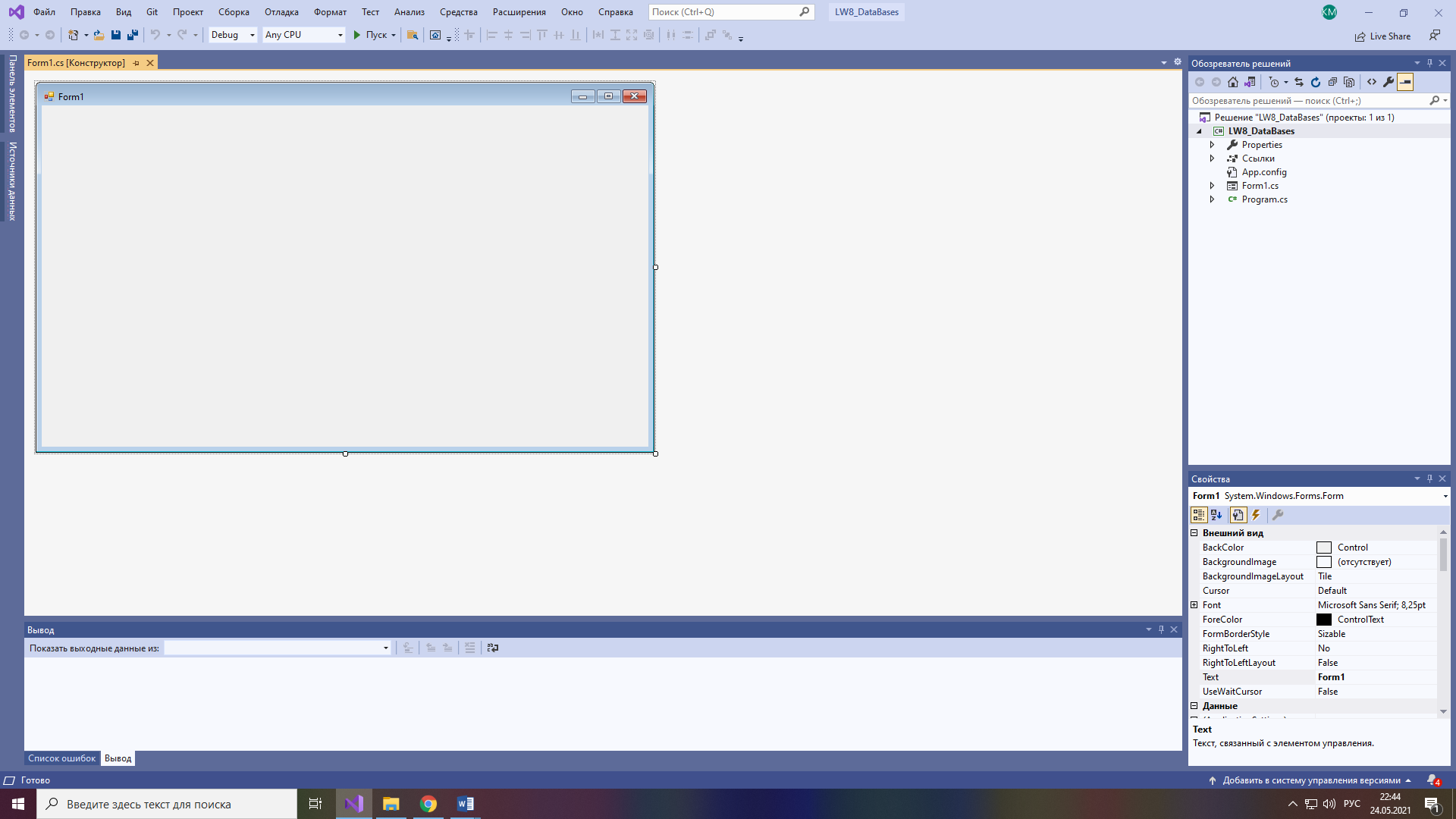


Рис. 1.1. Окно Visual studio с пустым проектом приложения.

**2 задание.**

Чтобы подключиться к SQL server мне понадобилось ввести имя сервера в мастере настройки подключения вручную (так как сам он не захотел находить мой локальный сервер), а затем выбрать базу данных, которую я хочу добавить. И вуаля! Результат на рисунке 2.1.

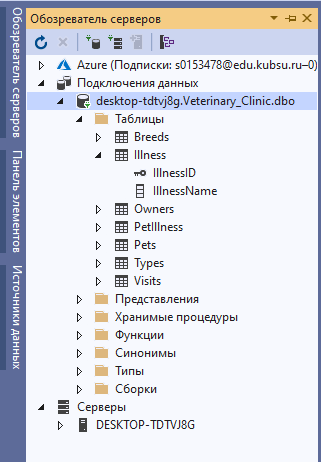


Рис. 2.1. Обозреватель серверов

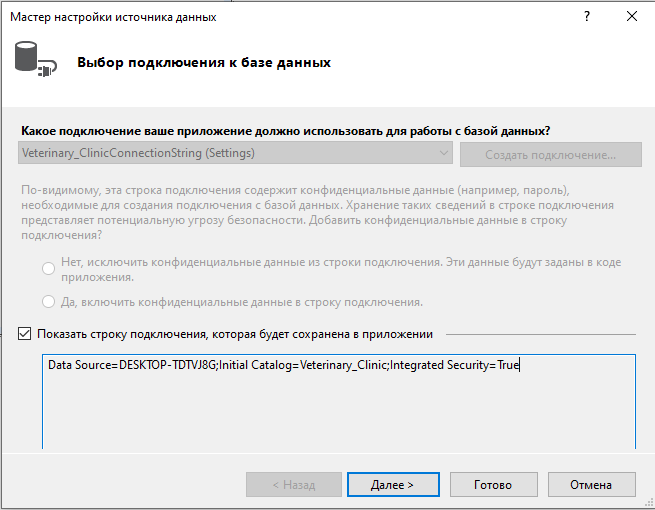


Рис 2.2. Окно мастера настройки подключения к БД.

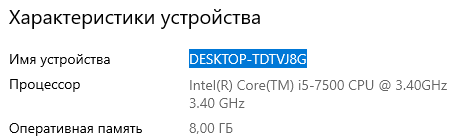


Рис 2.3. Окно параметров компьютера

Как видно из рисунков 2.2 и 2.3, мастер настройки подключения самостоятельно прописал имя компьютера в поле DataSource. Значит, переносимость приложения и работа его на другом ПК обеспечена

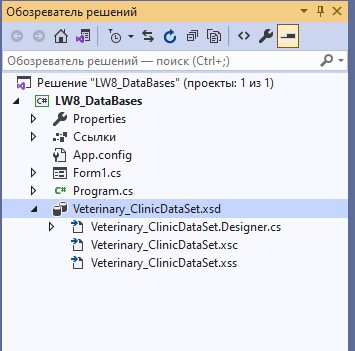


Рис 2.4. Успешное добавление БД к проекту.

**3 задание.**

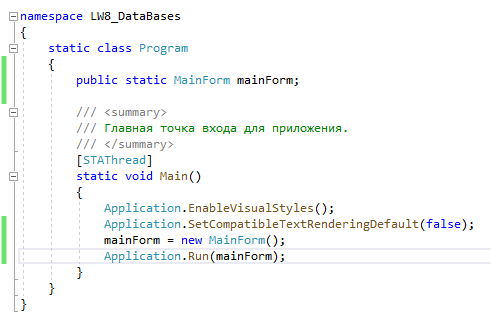


Рис 3.1. Изменения в файле Program.cs.

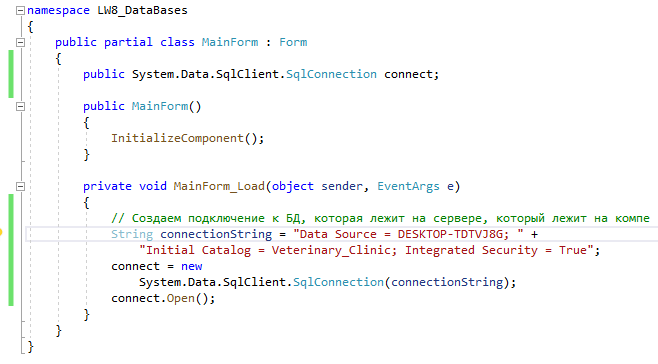


Рис 3.2. MainForm.cs.

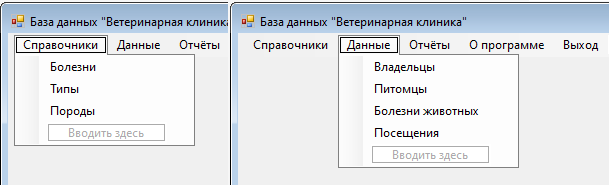


Рис 3.3. Окно главной формы.

**4, 5, 6 задание.**

Таблица 1. Перечень разработанных компонентов приложения

|  |  |  |
| --- | --- | --- |
| №  пп | Имя формы | Описание |
| 1 | MainForm | Главная и начальная форма приложения. Из нее осуществляется переход по всем остальным формам, которые составляют приложение |
| 2 | AboutProgram | Форма, содержащая информацию о приложении, разработчике |
| 3 | Reports | Содержит различные отчёты, которые извлекают различные данные из БД |
| 4 | Types | Форма для работы с таблицей «Виды» из БД |
| 5 | Breeds | Форма для работы с таблицей «Породы» из БД |
| 6 | Illness | Форма для работы с таблицей «Болезни» из БД |
| 7 | Owners | Форма для работы с таблицей «Владельцы» из БД. Пока что пустая. |
| 8 | Pets | Форма для работы с таблицей «Питомцы» из БД. Пока что пустая. |
| 9 | PetsIllness | Форма для работы с таблицей «Болезни питомцев» из БД. Пока что пустая. |
| 10 | Visits | Форма для работы с таблицей «Посещения» из БД. Пока что пустая. |

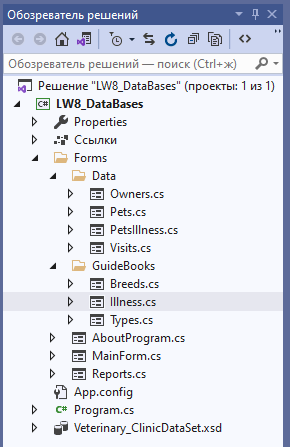


Рис 4.1. Обозреватель решений

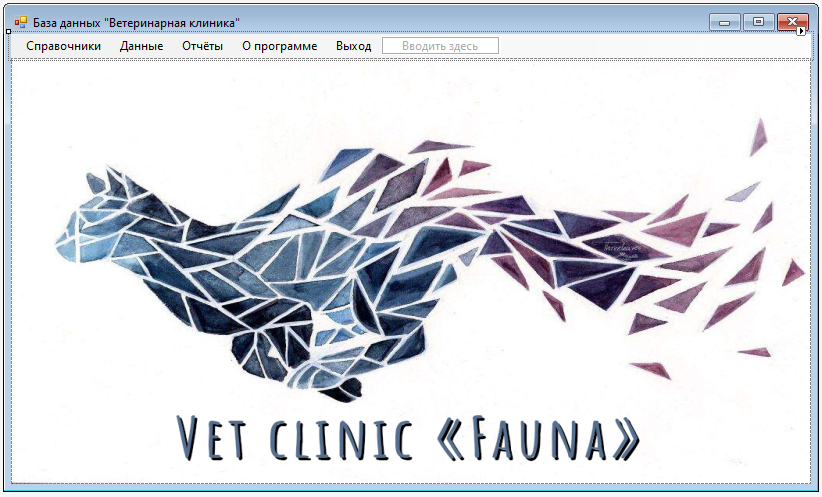


Рис 4.2. MainForm (изображение).

namespace LW8\_DataBases

{

public partial class MainForm : Form

{

public System.Data.SqlClient.SqlConnection connect;

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

// Создаем подключение к БД, которая лежит на сервере, который лежит на компе

String connectionString = "Data Source = DESKTOP-TDTVJ8G; " +

"Initial Catalog = Veterinary\_Clinic; Integrated Security = True";

connect = new

System.Data.SqlClient.SqlConnection(connectionString);

connect.Open();

}

private void typesToolStripMenuItem\_Click(object sender, EventArgs e)

{

Types typesForm = new Types();

typesForm.Show();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void illnessToolStripMenuItem\_Click(object sender, EventArgs e)

{

Illness illnessForm = new Illness();

illnessForm.Show();

}

private void breedsToolStripMenuItem\_Click(object sender, EventArgs e)

{

Breeds breedsForm = new Breeds();

breedsForm.Show();

}

private void ownersToolStripMenuItem\_Click(object sender, EventArgs e)

{

Owners ownersForm = new Owners();

ownersForm.Show();

}

private void petsToolStripMenuItem\_Click(object sender, EventArgs e)

{

Pets petsForm = new Pets();

petsForm.Show();

}

private void petsIllnessToolStripMenuItem\_Click(object sender, EventArgs e)

{

PetsIllness petsIllnessForm = new PetsIllness();

petsIllnessForm.Show();

}

private void visitsToolStripMenuItem\_Click(object sender, EventArgs e)

{

Visits visitsForm = new Visits();

visitsForm.Show();

}

private void reportsToolStripMenuItem\_Click(object sender, EventArgs e)

{

Reports reportsForm = new Reports();

reportsForm.Show();

}

private void aboutProgramToolStripMenuItem\_Click(object sender, EventArgs e)

{

AboutProgram aboutProgramForm = new AboutProgram();

aboutProgramForm.Show();

}

}

}

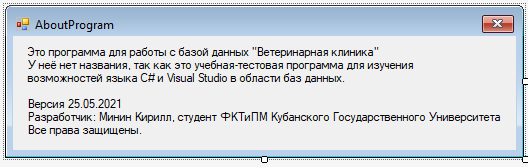


Рис 4.3. AboutProgram (изображение).

namespace LW8\_DataBases

{

public partial class AboutProgram : Form

{

public AboutProgram()

{

InitializeComponent();

}

}

}

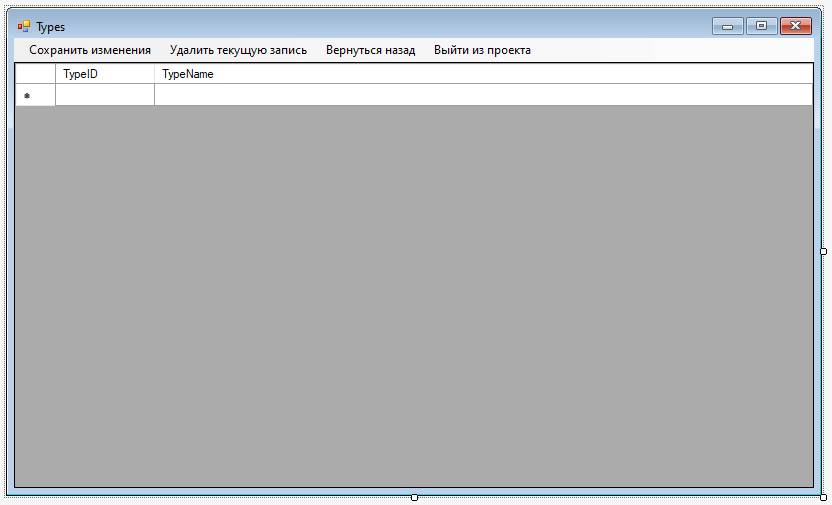


Рис 4.4. Types (изображение).

namespace LW8\_DataBases

{

public partial class Types : Form

{

public Types()

{

InitializeComponent();

}

private void saveChangesToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

// записываем данные из DataGridView в DataSet

dataGridView1.DataSource = veterinary\_ClinicDataSet.Illness;

// обновляем БД

typesTableAdapter.Update(veterinary\_ClinicDataSet);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ОШИБКА");

}

}

private void Types\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "veterinary\_ClinicDataSet.Types". При необходимости она может быть перемещена или удалена.

this.typesTableAdapter.Fill(this.veterinary\_ClinicDataSet.Types);

}

private void deleteCurrentNoteToolStripMenuItem\_Click(object sender, EventArgs e)

{

// в буферную таблицу записываем текущую таблицу из DataGrid

CurrencyManager curMan = (CurrencyManager)

dataGridView1.BindingContext[dataGridView1.DataSource];

// если таблица непустая, то

if (curMan.Count > 0)

{

curMan.RemoveAt(curMan.Position); // удаляем текущую позицию

// по логике, сохранение в БД нам здесь не нужно при удалении

// ведь у нас есть кнопка "сохранить изменения"

//illnessTableAdapter.Update(veterinary\_ClinicDataSet);

}

}

private void exitFromProjectToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void backToolStripMenuItem\_Click(object sender, EventArgs e)

{

// закрываем текущую форму

this.Close();

// выводим на 1 план главную форму

Program.mainForm.Activate();

}

}

}

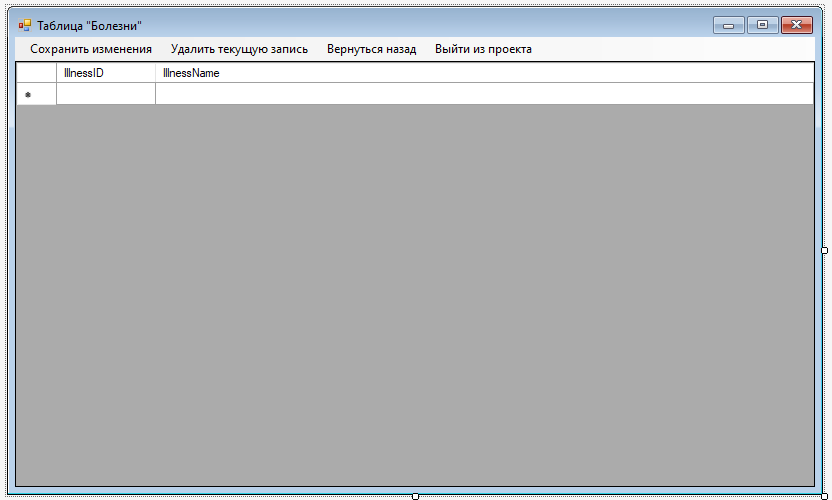


Рис 4.5. Illness (изображение).

namespace LW8\_DataBases

{

public partial class Illness : Form

{

public Illness()

{

InitializeComponent();

}

private void Illness\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "veterinary\_ClinicDataSet.Illness". При необходимости она может быть перемещена или удалена.

this.illnessTableAdapter.Fill(this.veterinary\_ClinicDataSet.Illness);

}

private void deleteCurrentNoteToolStripMenuItem\_Click(object sender, EventArgs e)

{

// в буферную таблицу записываем текущую таблицу из DataGrid

CurrencyManager curMan = (CurrencyManager)

dataGridView1.BindingContext[dataGridView1.DataSource];

// если таблица непустая, то

if (curMan.Count > 0)

{

curMan.RemoveAt(curMan.Position); // удаляем текущую позицию

// по логике, сохранение в БД нам здесь не нужно при удалении

// ведь у нас есть кнопка "сохранить изменения"

//illnessTableAdapter.Update(veterinary\_ClinicDataSet);

}

}

private void exitFromProjectToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void saveChangesToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

// записываем данные из DataGridView в DataSet

dataGridView1.DataSource = veterinary\_ClinicDataSet.Illness;

// обновляем БД

illnessTableAdapter.Update(veterinary\_ClinicDataSet);

}

catch(Exception ex)

{

MessageBox.Show(ex.Message, "ОШИБКА");

}

}

private void backToolStripMenuItem\_Click(object sender, EventArgs e)

{

// закрываем текущую форму

this.Close();

// выводим на 1 план главную форму

Program.mainForm.Activate();

}

}

}

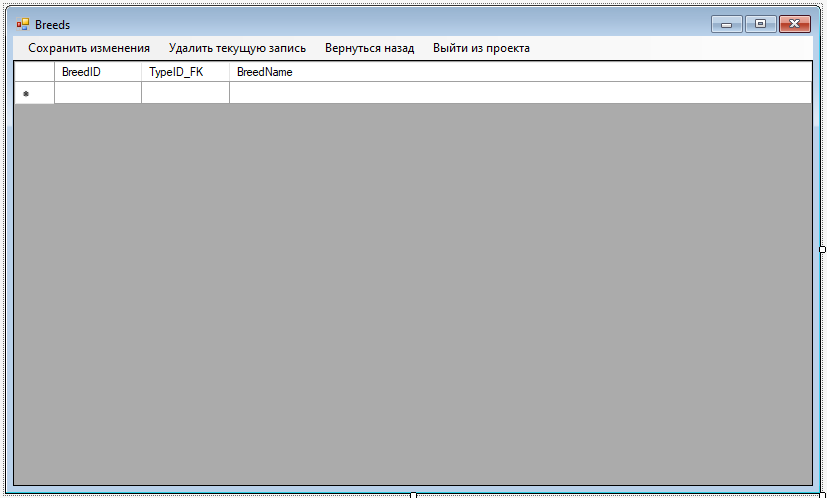


Рис 4.6. Breeds (изображение).

namespace LW8\_DataBases

{

public partial class Breeds : Form

{

public Breeds()

{

InitializeComponent();

}

private void Breeds\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "veterinary\_ClinicDataSet.Breeds". При необходимости она может быть перемещена или удалена.

this.breedsTableAdapter.Fill(this.veterinary\_ClinicDataSet.Breeds);

}

private void saveChangesToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

// записываем данные из DataGridView в DataSet

dataGridView1.DataSource = veterinary\_ClinicDataSet.Illness;

// обновляем БД

breedsTableAdapter.Update(veterinary\_ClinicDataSet);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ОШИБКА");

}

}

private void deleteCurrentNoteToolStripMenuItem\_Click(object sender, EventArgs e)

{

// в буферную таблицу записываем текущую таблицу из DataGrid

CurrencyManager curMan = (CurrencyManager)

dataGridView1.BindingContext[dataGridView1.DataSource];

// если таблица непустая, то

if (curMan.Count > 0)

{

curMan.RemoveAt(curMan.Position); // удаляем текущую позицию

// по логике, сохранение в БД нам здесь не нужно при удалении

// ведь у нас есть кнопка "сохранить изменения"

//illnessTableAdapter.Update(veterinary\_ClinicDataSet);

}

}

private void exitFromProjectToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void backToolStripMenuItem\_Click(object sender, EventArgs e)

{

// закрываем текущую форму

this.Close();

// выводим на 1 план главную форму

Program.mainForm.Activate();

}

}

}

Формы Owners, Pets, PetsIllness и Visits – пустые (пока что). В их коде то же самое, что и у формы AboutProgram. А выглядят они как на рисунке 4.7

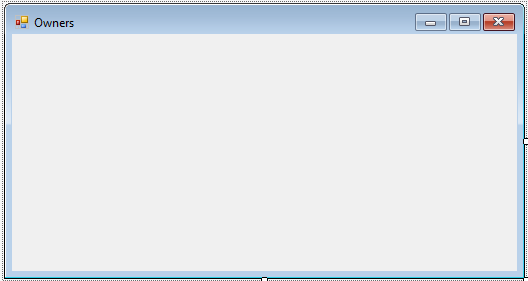


Рис 4.7. Пустые формы(изображение).

Вывод: в результате выполнения данной лабораторной работы мне удалось дать начало своему собственному Windows-приложению, которое позволяет работать с созданной мною ранее базой данных.