Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**УЧЕБНАЯ ПРАКТИКА**

**«Составление и отладка программ с двумерными массивами»**

Вариант: 22

по дисциплине

«Разработка программных модулей для компьютерных сетей»

**Выполнил(а):**

Синюкова Ирина

студент(ка) группы

ИСП-20А

09.02.07 Информационные системы и программирование

очной формы обучения

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_ \_\_\_\_\_\_\_\_\_\_/

**Оглавление**

Постановка задачи...................................................................................................3

[Разработка БД](#Бд) 3-6

[Разработка главной страницы](#Главнаястраница) 6-8

Реализация функций………………………...…...….….……………………..8-11

[Вывод данных на печать 11-12](" \l "выводданных)

[Приложения](#приложения)......................................................................................................12-19

**1.** **Постановка задачи**

Разработать БД по своей предметной области и на её основе создать проект WPF.

Задача: Необходимо написать программу, которая будет хранить информацию питомцах приюта для собак. (например, такую как: кличка, возраст, вид животного, где был найден, вес, список прививок, есть ли опекун или нет, уникальный номер, примерная порода).

**2. Ра****зработка структуры БД в среде SQL SERVER. Создание проекта WPF в Visual Studio (подключение БД). Спроектировать интерфейс приложения.**

**2.1 Разработка структуры БД в среде SQL SERVER.**

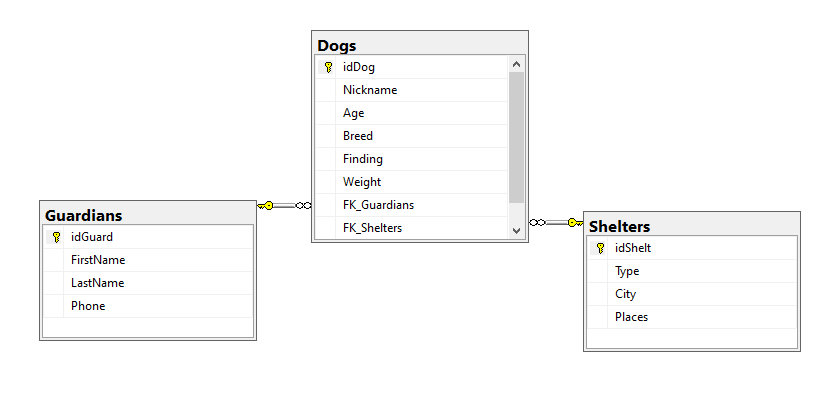
****

рис.1 «БД в SQL SERVER »

**2.2 Создание проекта WPF в Visual Studio (подключение БД).**

Подключаем базу с помощью ADO.NET EDM.

Создаём класс Frame для загрузки данных:

internal class ClassFrame

{

public static Frame frmObj { get; set; } //объект типа Frame

}

На главном окне используем сам класс:

public MainWindow()

{

InitializeComponent();

ClassFrame.frmObj = frmMain;

frmMain.Navigate(new Pages.PageDogs());

}

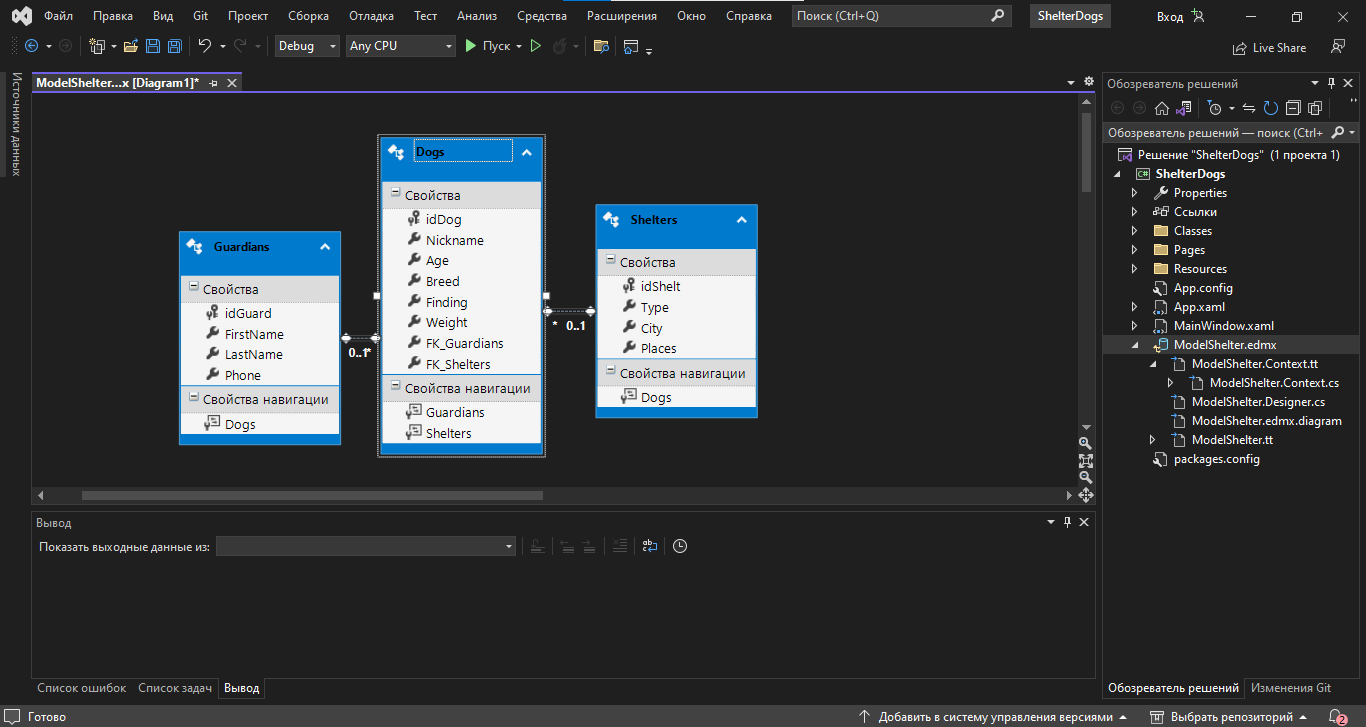
****

рис.2 «БД в Visual Studio »

**2.3 Спроектировать интерфейс приложения (примерный)**

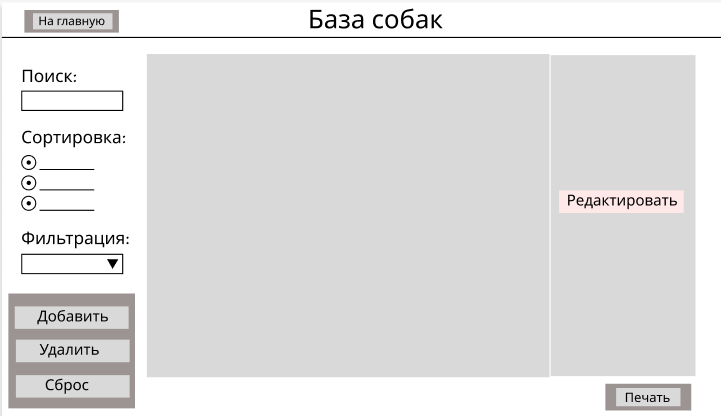


рис.3 «Интерфейс(БД собак)»

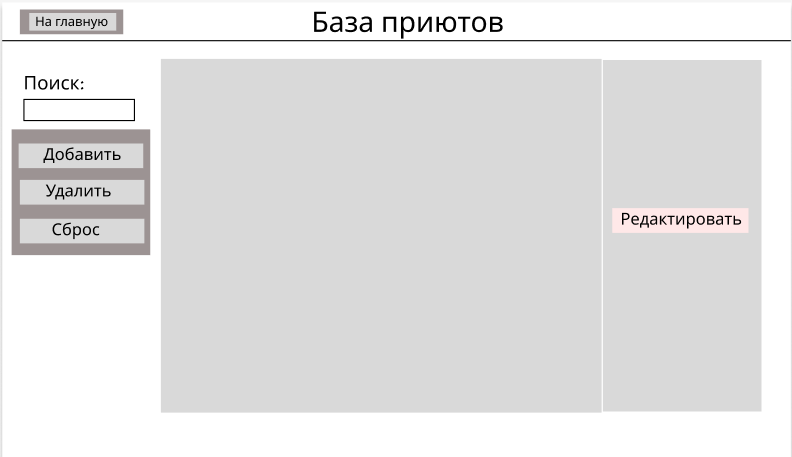


рис.4 «Интерфейс(БД приютов)»

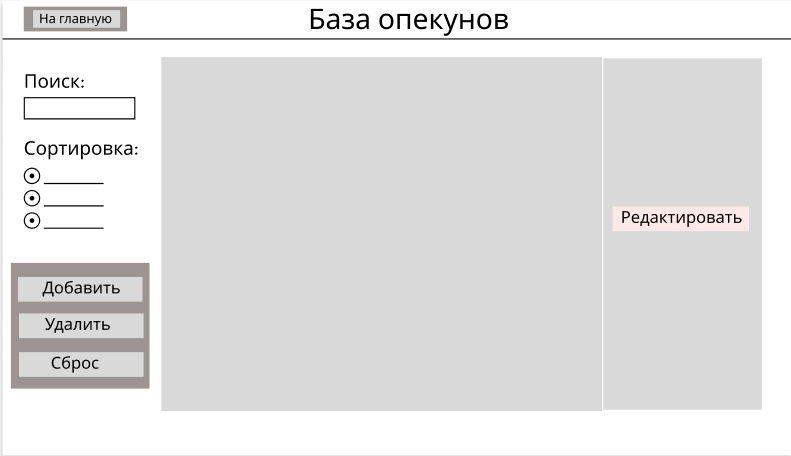


рис.5 «Интерфейс(БД опекунов)»



рис.6 «Интерфейс(Окно добавления/редактирования)»

**3. Разработка главной страницы. Реализация функций: просмотр, добавление, редактирование, удаление данных.**

**3.1 Главная страница Dogs**

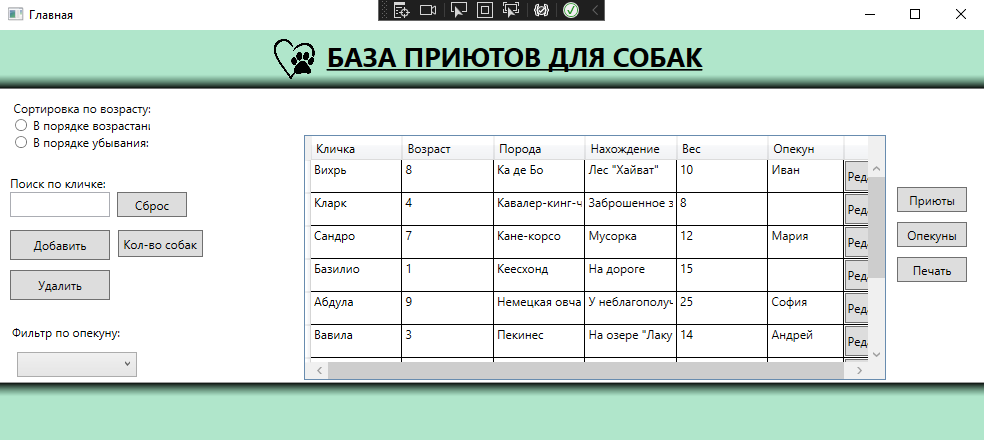
****

рис.7 «Программа»

**3.2 Просмотр**

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

ShelterEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.ToList();

}

}

**3.3 Добавление**

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditDogs(null));

}

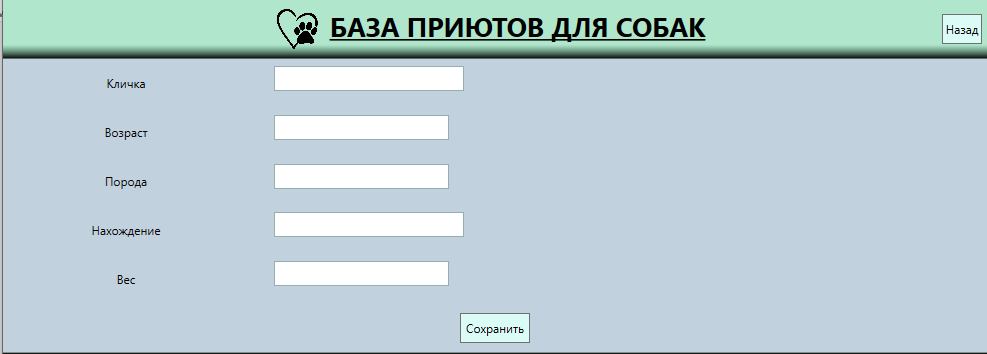
****

рис.8 «Окно добавления»

**3.4 Редактирование**

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditDogs((sender as Button).DataContext as Dogs));

}

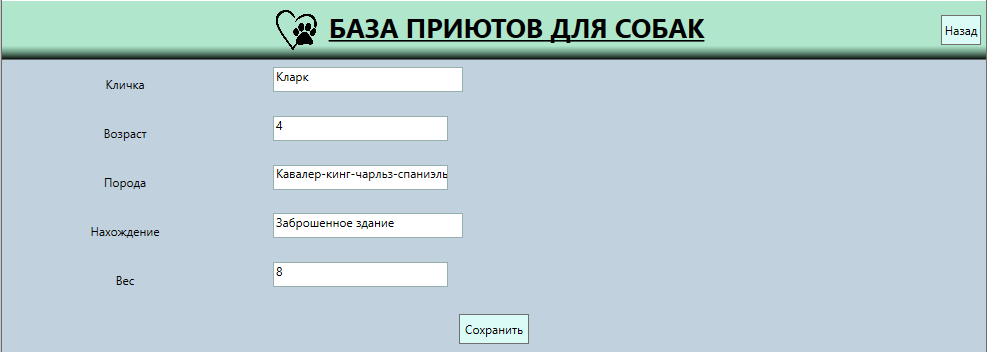
****

рис.9 «Окно редактирования»

**3.5 Удаление**

private void BtnDelete\_Click(object sender, RoutedEventArgs e)

{

var DogForRemoving = DGridDog.SelectedItems.Cast<Dogs>().ToList();

if (MessageBox.Show($"Вы действительно хотите удалить {DogForRemoving.Count()} данную собаку?",

"Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ShelterEntities.GetContext().Dogs.RemoveRange(DogForRemoving);

ShelterEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

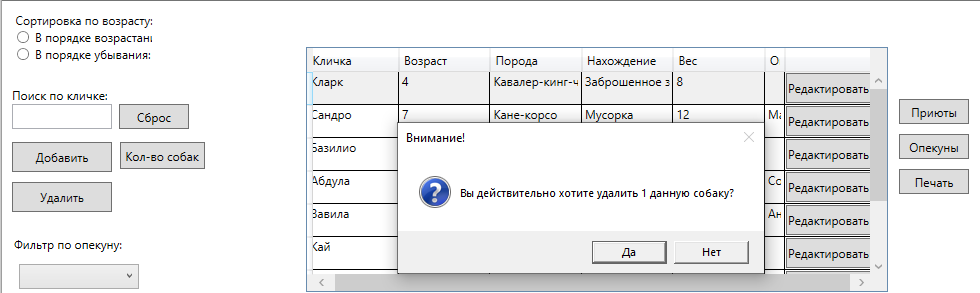
****

рис.10 «Удаление: сообщение пользователю»

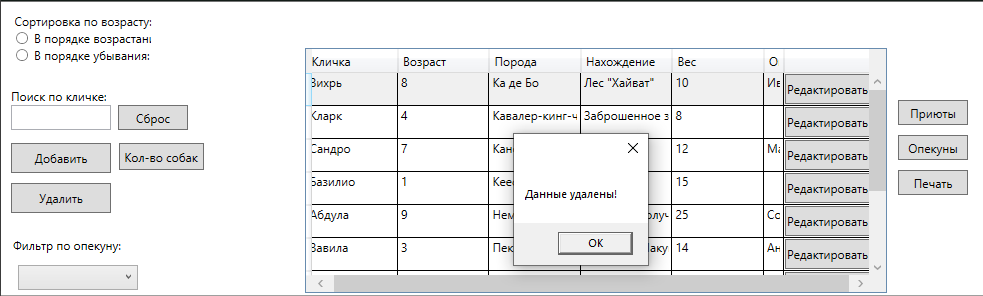
****

рис.11 «Удаление: действие произведено»

**4. Реализация функция: сортировка, поиск, фильтрация, подсчёт значений.**

**4.1 Сортировка (по возрасту: по возрастанию, по убыванию)**

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.OrderBy(x => x.Age).ToList();

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.OrderByDescending(x => x.Age).ToList();

}

****

рис.12 «Сортировка (по возрастанию)»

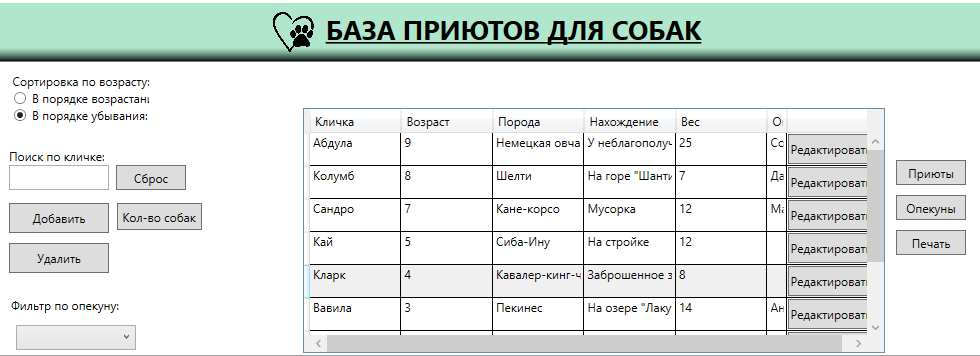
****

рис.13 «Сортировка (по убыванию)»

**4.2 Поиск (по кличке)**

private void TxbSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.Where(x => x.Nickname.ToLower().Contains(TxbSearch.Text.ToLower())).ToList();

}

****

рис.14 «Поиск по кличке»

**4.3 Фильтрация**

private void CmbFiltrProv\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int id = CmbFiltrProv.SelectedIndex + 1;

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.Where(x => x.FK\_Guardians == id).ToList();

}

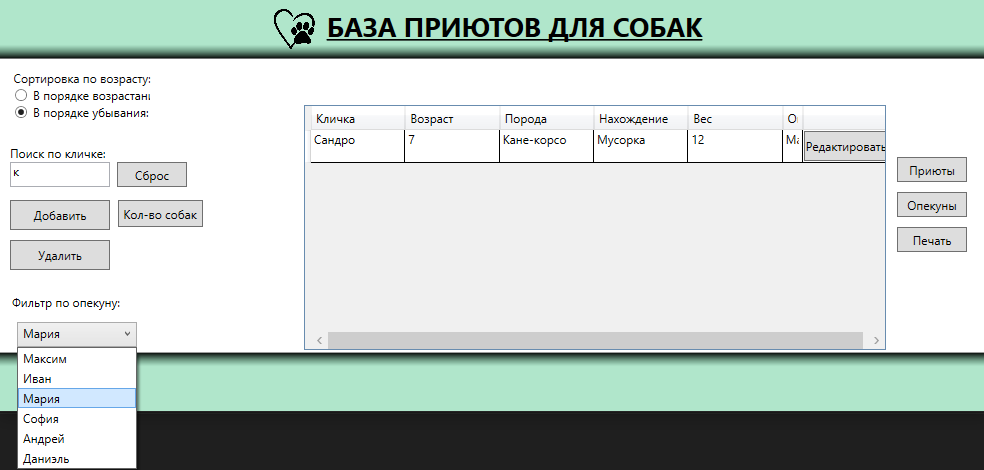
****

рис.15 «Фильтр по опекуну»

**4.4 Кол-во записей**

private void BtnCount\_Click(object sender, RoutedEventArgs e)

{

int num = DGridDog.Items.Count;

MessageBox.Show("Всего собак:" + num);

}

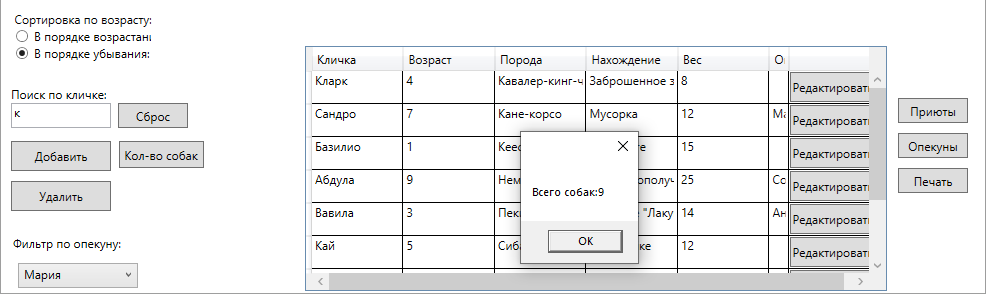


рис.16 «Подсчёт данных: кол-во собак»

**5. Вывод данных на печать (Excel).**

private void BtnExcel\_Click(object sender, RoutedEventArgs e)

{

var app = new Excel.Application();

Excel.Workbook wb = app.Workbooks.Add();

Excel.Worksheet worksheet = app.Worksheets.Item[1];

int indexRows = 5;

worksheet.Cells[2, 2] = "Собаки, зарегестрированные в приютах:";

worksheet.Cells[4, 1] = DateTime.Now.ToString();

worksheet.Cells[1][indexRows] = "Номер";

worksheet.Cells[2][indexRows] = "Кличка";

worksheet.Cells[3][indexRows] = "Возраст";

worksheet.Cells[4][indexRows] = "Порода";

worksheet.Cells[5][indexRows] = "Вес";

var printItems = DGridDog.Items;

foreach (Dogs item in printItems)

{

worksheet.Cells[1][indexRows + 1] = indexRows;

worksheet.Cells[2][indexRows + 1] = item.Nickname;

worksheet.Cells[3][indexRows + 1] = item.Age;

worksheet.Cells[4][indexRows + 1] = item.Breed;

worksheet.Cells[5][indexRows + 1] = item.Weight;

indexRows++;

}

Excel.Range range = worksheet.Range[worksheet.Cells[2][indexRows + 1],

worksheet.Cells[6][indexRows + 1]];

range.ColumnWidth = 10;

range.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

worksheet.Cells[indexRows + 2, 3] = "Подпись:";

worksheet.Cells[indexRows + 2, 5] = "Синюкова И.А.";

app.Visible = true;

}

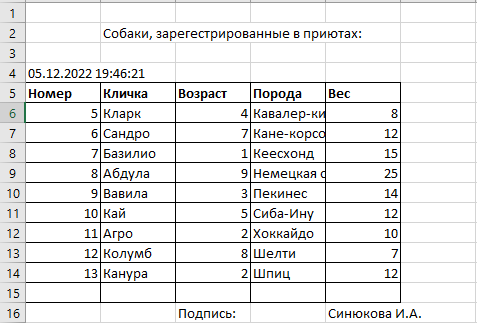
****

рис.17 «Вывод данных (Excel)»

**6. Приложения (логика)**

# Главное окно (собаки)

public partial class PageDogs : Page

{

private static ShelterEntities \_context = new ShelterEntities();

public PageDogs()

{

InitializeComponent();

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.ToList();

CmbFiltrProv.ItemsSource = ShelterEntities.GetContext().Guardians.ToList();

CmbFiltrProv.SelectedValuePath = "idGuard";

CmbFiltrProv.DisplayMemberPath = "FirstName";

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

ShelterEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.ToList();

}

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditDogs((sender as Button).DataContext as Dogs));

}

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditDogs(null));

}

private void BtnDelete\_Click(object sender, RoutedEventArgs e)

{

var DogForRemoving = DGridDog.SelectedItems.Cast<Dogs>().ToList();

if (MessageBox.Show($"Вы действительно хотите удалить {DogForRemoving.Count()} данную собаку?",

"Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ShelterEntities.GetContext().Dogs.RemoveRange(DogForRemoving);

ShelterEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void TxbSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.Where(x => x.Nickname.ToLower().Contains(TxbSearch.Text.ToLower())).ToList();

}

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.OrderBy(x => x.Age).ToList();

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.OrderByDescending(x => x.Age).ToList();

}

private void BtnResetAll\_Click(object sender, RoutedEventArgs e)

{

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.ToList();

}

private void CmbFiltrProv\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int id = CmbFiltrProv.SelectedIndex + 1;

DGridDog.ItemsSource = ShelterEntities.GetContext().Dogs.Where(x => x.FK\_Guardians == id).ToList();

}

private void BtnShelter\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageShelters((sender as Button).DataContext as Shelters));

}

private void BtnGuardian\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageGuardians((sender as Button).DataContext as Guardians));

}

private void BtnExcel\_Click(object sender, RoutedEventArgs e)

{

var app = new Excel.Application();

Excel.Workbook wb = app.Workbooks.Add();

Excel.Worksheet worksheet = app.Worksheets.Item[1];

int indexRows = 5;

worksheet.Cells[2, 2] = "Собаки, зарегестрированные в приютах:";

worksheet.Cells[4, 1] = DateTime.Now.ToString();

worksheet.Cells[1][indexRows] = "Номер";

worksheet.Cells[2][indexRows] = "Кличка";

worksheet.Cells[3][indexRows] = "Возраст";

worksheet.Cells[4][indexRows] = "Порода";

worksheet.Cells[5][indexRows] = "Вес";

Excel.Range Bold = worksheet.Range[worksheet.Cells[1]

[indexRows], worksheet.Cells[5][indexRows]];

Bold.Font.Bold = worksheet.Cells[1][indexRows].Font.Bold = true;

Excel.Range Borders = worksheet.Range[worksheet.Cells[1][5], worksheet.Cells[5][15]];

Borders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle = Excel.XlLineStyle.xlContinuous;

Borders.ColumnWidth = 10;

var printItems = DGridDog.Items;

foreach (Dogs item in printItems)

{

worksheet.Cells[1][indexRows + 1] = indexRows;

worksheet.Cells[2][indexRows + 1] = item.Nickname;

worksheet.Cells[3][indexRows + 1] = item.Age;

worksheet.Cells[4][indexRows + 1] = item.Breed;

worksheet.Cells[5][indexRows + 1] = item.Weight;

indexRows++;

}

Excel.Range range = worksheet.Range[worksheet.Cells[2][indexRows + 1],

worksheet.Cells[6][indexRows + 1]];

range.ColumnWidth = 10;

range.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

worksheet.Cells[indexRows + 2, 3] = "Подпись:";

worksheet.Cells[indexRows + 2, 5] = "Синюкова И.А.";

app.Visible = true;

}

private void BtnCount\_Click(object sender, RoutedEventArgs e)

{

int num = DGridDog.Items.Count;

MessageBox.Show("Всего собак:" + num);

}

}

# Окно опекунов

public partial class PageGuardians : Page

{

public Guardians \_currentGuardians = new Guardians();

private static ShelterEntities \_context = new ShelterEntities();

public PageGuardians(Guardians selectedGuardians)

{

InitializeComponent();

if (selectedGuardians != null)

\_currentGuardians = selectedGuardians;

DataContext = \_currentGuardians;

DGridGuardians.ItemsSource = ShelterEntities.GetContext().Guardians.ToList();

}

private void Page\_IsVisibleChanged1(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

ShelterEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridGuardians.ItemsSource = ShelterEntities.GetContext().Guardians.ToList();

}

}

private void BtnEditGuard\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditGuardians((sender as Button).DataContext as Guardians));

}

private void btnAddGuardians\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditGuardians(null));

}

private void btnDeleteGuardians\_Click(object sender, RoutedEventArgs e)

{

var GuardiansForRemoving = DGridGuardians.SelectedItems.Cast<Guardians>().ToList();

if (MessageBox.Show($"Вы действительно хотите удалить {GuardiansForRemoving.Count()} данного опекуна?",

"Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ShelterEntities.GetContext().Guardians.RemoveRange(GuardiansForRemoving);

ShelterEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

DGridGuardians.ItemsSource = ShelterEntities.GetContext().Guardians.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void TxbSearchLastName\_TextChanged(object sender, TextChangedEventArgs e)

{

DGridGuardians.ItemsSource = ShelterEntities.GetContext().Guardians.Where(x => x.LastName.ToLower().Contains(TxbSearchLastName.Text.ToLower())).ToList();

}

}

# Окно приютов

public partial class PageShelters : Page

{

public Shelters \_currentShelters = new Shelters();

private static ShelterEntities \_context = new ShelterEntities();

public PageShelters(Shelters selectedShelters)

{

InitializeComponent();

if (selectedShelters != null)

\_currentShelters = selectedShelters;

DataContext = \_currentShelters;

DGridShelter.ItemsSource = ShelterEntities.GetContext().Shelters.ToList();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditShelter((sender as Button).DataContext as Shelters));

}

private void Page\_IsVisibleChanged1(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

ShelterEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridShelter.ItemsSource = ShelterEntities.GetContext().Shelters.ToList();

}

}

private void btnAddShelters\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditShelter(null));

}

private void btnDeleteShelter\_Click(object sender, RoutedEventArgs e)

{

var ShelterForRemoving = DGridShelter.SelectedItems.Cast<Shelters>().ToList();

if (MessageBox.Show($"Вы действительно хотите удалить {ShelterForRemoving.Count()} данный приют?",

"Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ShelterEntities.GetContext().Shelters.RemoveRange(ShelterForRemoving);

ShelterEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

DGridShelter.ItemsSource = ShelterEntities.GetContext().Shelters.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void TxbSearchCity\_TextChanged(object sender, TextChangedEventArgs e)

{

DGridShelter.ItemsSource = ShelterEntities.GetContext().Shelters.Where(x => x.City.ToLower().Contains(TxbSearchCity.Text.ToLower())).ToList();

}

}

# Окно добавления/редактирования

public partial class PageEditDogs : Page

{

private Dogs \_currentDogs = new Dogs();

public PageEditDogs(Dogs selectedDogs)

{

InitializeComponent();

if (selectedDogs != null)

\_currentDogs = selectedDogs;

DataContext= \_currentDogs;

}

private void BtnSave\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentDogs.Nickname))

error.AppendLine("Укажите кличку");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if(\_currentDogs.idDog == 0)

ShelterEntities.GetContext().Dogs.Add( \_currentDogs );

try

{

ShelterEntities.GetContext().SaveChanges();

MessageBox.Show("Новая собака добавлена");

}

catch(Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}