**ВВЕДЕНИЕ**

В данной пояснительной записке рассматривается описание программного модуля «Сеть детских садов» на основе объектно-ориентированного подхода.

При объектно-ориентированном подходе программа представляет собой описание объектов, их свойств (или атрибутов), совокупностей (или классов), отношений между ними, способов их взаимодействия и операций над объектами (или методов).

Несомненным преимуществом данного подхода является концептуальная близость к предметной области произвольной структуры и назначения. Механизм наследования атрибутов и методов позволяет строить производные понятия на основе базовых и таким образом создавать модель сколь угодно сложной предметной области с заданными свойствами.

Еще одним теоретически интересным и практически важным свойством объектно-ориентированного подхода является поддержка механизма обработки событий, которые изменяют атрибуты объектов и моделируют их взаимодействие в предметной области.

В отличие от предыдущих подходов к программированию, объектно-ориентированный подход требует глубокого понимания основных принципов, или, иначе, концепций, на которых он базируется. К числу основополагающих понятий ООП обычно относят абстракцию данных, наследование, инкапсуляцию и полиморфизм.

Объектно-ориентированное программирование в настоящее время является абсолютным лидером в области прикладного программирования.

В качестве основного инструмента разработки применяется Microsoft Visual Studio 2019. Язык программирования C#.

**ЦЕЛЬ И ЗАДАЧИ ВЫПОЛНЕНИЯ РАБОТЫ**

Целью данного курсового проекта является учебный план на основе объектно-ориентированного подхода, а также получение навыков в реализации этого подхода, проектировании и реализации схемы данных.

Вы работаете в МФЦ. Вашей задачей является отслеживание, редактирование и добавление детских садов относящихся к вашей организации.

Ваша деятельность организована следующим образом: Вы должны собирать информацию о саде и редактировать по мере изменений (Название сада, город, адрес, количество детей, плата за месяц, директор и его данные).

Задачи выполнения работы:

* исследование предметной области с целью выявления основных сущностей и их атрибутов;
* проектирование иерархии классов и интерфейсов на основе выделенных сущностей;
* использование наследования и агрегации для структуризации классов;
* применение принципа инкапсуляции к классам;
* использование интерфейсов, описывающих операции в данной предметной области, для обеспечения множественного наследования;
* тестирование созданных классов в клиентском приложении, автоматизирующем деятельность предметной области;
* документация разработанных классов при помощи XML комментариев.
* сохранение данных в SQL -файл.
* считывание данных из SQL -файла.

**1 ВНЕШНЕЕ ОПИСАНИЕ**

**1.1 Анализ предметной области**

Анализ предметной области – важнейший этап разработки программного обеспечения.

Анализ работы детского сада позволяет определить ключевые понятия, выяснить задачи, требующие решения в рамках проекта, проанализировать функциональные требования будущего продукта (набор функций, предоставляемых программой) и нефункциональные (надёжность, эффективность, изучаемость, модифицируемость и т.п.).

Компонентами данной предметной области являются Детские сады и директора. Данное приложение должно позволять пользователю создавать записи, удалять, редактировать и сохранять данные в приложении.

Нефункциональные требования к программному средству:

* надежность: программа должна быть автономной;
* эффективность: программа должна иметь минимальные требования к аппаратному обеспечению. Для использования данного программного средства не должна требоваться установка на компьютер. Не должно требоваться дополнительных периферийных средств (принтер, сканер, дополнительные дисководы и т.д.);
* изучаемость: программа должна быть интуитивно понятна, иметь удобный пользовательский интерфейс;
* модифицируемость: программа должна быть легко модифицируемой вследствие небольшого исходного размера и объектно-ориентированного подхода.

**1.2. Функциональная спецификация**

Минимальные технические характеристики компьютера, на котором гарантируется стабильная работа программы:

* компьютер/процессор: компьютер с процессором класса Pentium II 450 МГц;
* память: 64 МБ ОЗУ;
* монитор: монитор VGA с разрешением 800x600 точек или более высоким, поддерживающий 256 цветов;
* операционная система: операционная система Windows XP с установленным Microsoft .NET Framework 4.5.2;
* наличие свободного дискового пространства на жёстком диске.

Внешние функции:

* внесение данных и редактирование данных об клиентах, номерах и поселении;
* сохранение данных в SQL -файл;
* считывание данных из SQL -файла;
* проверка вводимых данных и вывод сообщений об ошибках;
* программа должна выполнять все функции за предельно малое количество времени;

Критические случаи:

* отсутствие свободного места на жёстком диске. Любая операция с файловой системой должна быть отменена;
* некорректный SQL -файл. Программа не должна аварийно завершаться;
* завершение работы Windows. В этом случае программа должна завершать свою работу, не применяя выбранные пользователем настройки.

**2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Техническое задание определяет требования к будущему программному модулю со стороны основных процессов.

Система предназначена для решения перечисленных ниже задач:

* изменение списка Директоров (добавление, удаление, редактирование);
* изменение списка садиков (добавление, удаление, редактирование);
* изменение списка общей информации о садиках (добавление, удаление, редактирование);
* сохранение данных в SQL -файл;
* считывание данных из SQL -файла;

**3 ПРОГРАММИРОВАНИЕ**

**3.1 Описание классов**

В ООП главным элементом является класс, включающий множество объектов с одинаковыми свойствами, операциями и отношениями. Класс имеет внутреннее (реализацию) и внешнее представление – интерфейс. В данной программе реализованы классы SadNet, Director, DetSad, а также интерфейс WindowsForm.

Класс SadNet описывает общую информацию о саде. Содержит такие свойства, как: плата за месяц, дату открытия, название садика, имя директора.

public partial class SadNet

{

public int SadID { get; set; }

//Идентификатор

public int SadDirectorID { get; set; }

//Связь с Id директора

public decimal PlataForMonth { get; set; }

//Плата в месяц

public System.DateTime DateOpen { get; set; }

//Дата открытия

public int IdSad { get; set; }

//Связь с id садика

public virtual DetSad DetSad { get; set; }

// вирт. связь с таблицей сада

public virtual Director Director { get; set; }

//вирт. связь с таблицей директоров

}

Класс DetSad cодержит расширенную информацию о садике и такие свойства и данные, как его название, город, адрес и количество детей в нём.

public DetSad()

{

this.SadNets = new HashSet<SadNet>();

//Создание списка из элементов

}

public int SadID { get; set; }

//Идентификатор

public string Name { get; set; }

//Название сада

public string City { get; set; }

//Город

public string Adress { get; set; }

//Адрес

public string People { get; set; }

//Количесво детей

Класс Director описывает директоров. Содержит такие свойства, как фамилия, имя и отчество директора, возраст а также его стаж и образование.

public Director()

{

this.SadNets = new HashSet<SadNet>();

// Создание списка из элементов

}

public int DirectorID { get; set; }

//Идентификатор

public string DirName { get; set; }

//ФИО

public int DirAge { get; set; }

//Возраст

public string DirStage { get; set; }

//Стаж

public string DirEducation { get; set; }

//Образование

**3.2 Схема данных**

На рисунках 1-3 представлены таблицы файла SQL в режиме конструктора.

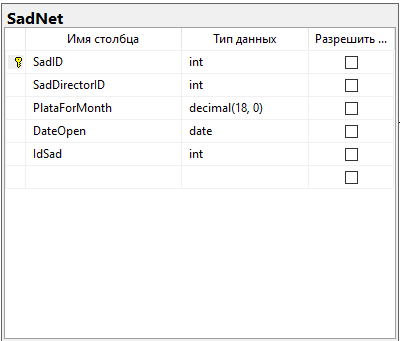


Рисунок 1- Таблица SadNet



Рисунок 2- Таблица DetSad

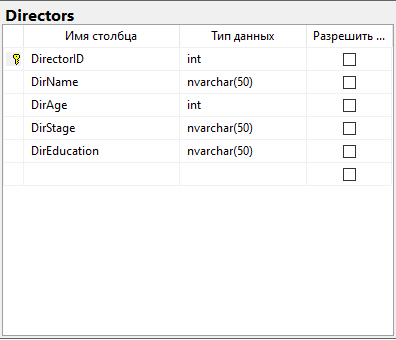


Рисунок 3- Таблица Director

На рисунке 4 представлена схема данных SQL

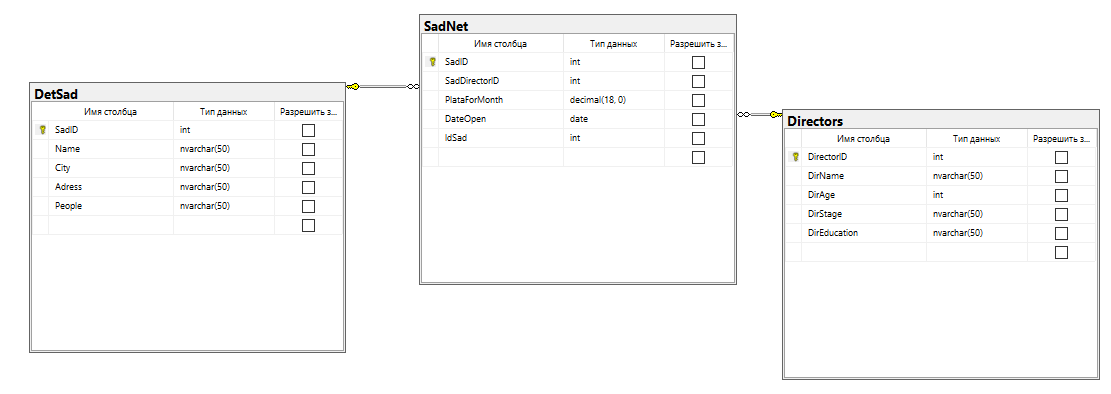


Рисунок 4-Схема данных

Словарь данных представлен в таблицах 1-3.

Таблица 1 – Словарь данных таблицы SadNet

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE | Null |
| 1 | SadID | INT | No |
| 2 | SadDirectorID | INT | No |
| 3 | PlataForMonth | Decimal | No |
| 4 | DateOpen | Date | No |
| 5 | idSad | INT | No |

Таблица 2 – Словарь данных таблицы DetSad

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE | Null |
| 1 | SadID | INT | No |
| 2 | Name | Nvachar(50) | No |
| 3 | City | Nvachar(50) | No |
| 4 | Adress | Nvachar(50) | No |
| 5 | People | Nvachar(50) | No |

Таблица 3 – Словарь данных таблицы Director

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE | REQUIRED |
| 1 | DirectorID | INT | No |
| 2 | DirName | Nvchar(50) | No |
| 3 | DirAge | INT | No |
| 4 | DirEducation | Nvchar(50) | No |
| 5 | DirStage | Nvchar(50) | No |

**4 КЛИЕНТСКОЕ ПРИЛОЖЕНИЕ**

**4.1 Структура**

В данном ПП для продуктивной работы реализован удобный пользовательский интерфейс, похожий на большинство Windows – приложений. Данное программное средство можно разбить на модули, взаимодействующие друг с другом. На рисунке 5 представлена структура приложения:

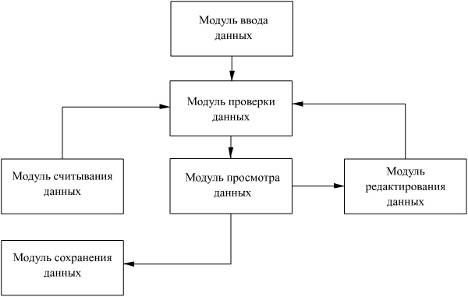


Рисунок 5 - Структура приложения

Модуль просмотра данных предоставляет возможность наглядного отображения данных.

Модуль проверки данных реализует методы:

* корректность вводимых пользователем данных;
* корректность считываемых из файла данных.

Модуль ввода данных, реализует простой графический интерфейс для ввода необходимых значений.

Модуль сохранения данных, формирует все записи в один SQL - файл.

Модуль считывания предоставляет возможность открытия SQL -файла и заполнения списков в программе.

Модуль редактирования записей предоставляет простой интерфейс для изменения ранее внесенных данных, а также добавления и удаления.

**4.2 Проектирование графического интерфейса**

На рисунке 6-7 представлена форма приложения и панель, содержащая таблицы данных:





Рисунок 6 - Формы MainWindow. Навиация по каждой странице и редактирование

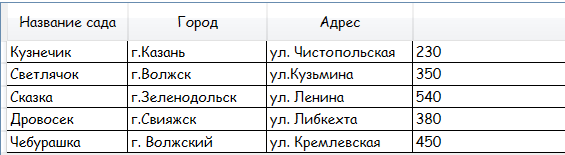


Рисунок 7 - Форма MainWindow. Таблица: SadNet

Для редактирования записей о детских садах используются отдельные формы приложения и панели для редактирования таблиц в MainWindow. Для редактирования таблицы «Детских садов» используется форма AddOrEditDetSad под названием «Добавление/Редактирование Детского сада», для таблицы «Директоров», форма AddOrEditDirectors с названием «Добавление/Редактирование директора, а для таблицы «Сети садов», форма AddOrEditSadNet с названием «Добавление/Редактирование в сети. Эти формы изображены на рисунках 8-10.

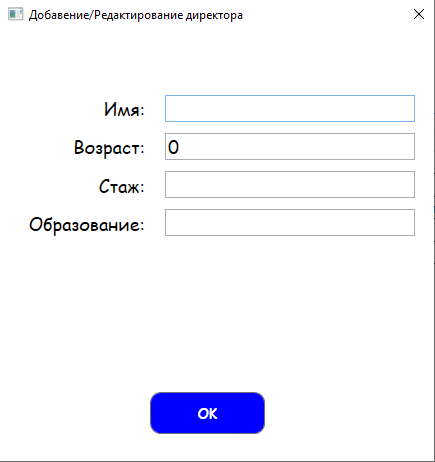


Рисунок 8 - Форма AddOrEditDetSad

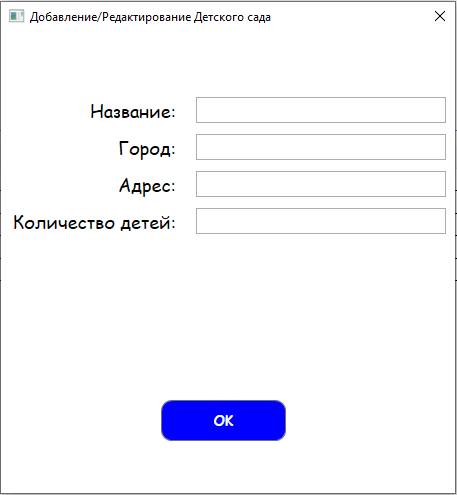


Рисунок 9 - Форма AddOrEditDirectors

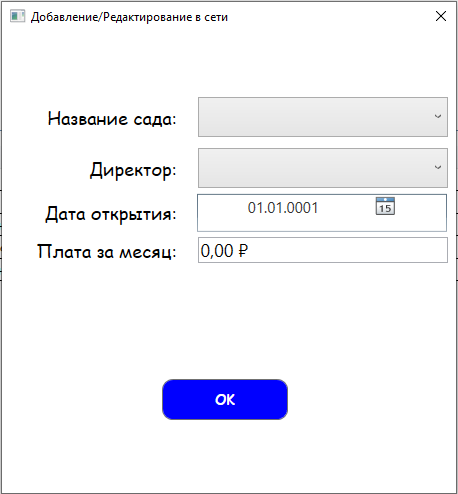


Рисунок 10 - Форма AddOrEditSadNet

Удаление студентов и предметов, осуществляется с помощью специальной кнопки, расположенной на главной форме приложения у каждой таблицы. Нажатием кнопки «Удалить» - удаляем из списка. Аналогично для других списков. Кнопки отображены на рисунке 11(Рисунок мусорки):



Рисунок 11 – Панель инструментов

**5 ТЕСТИРОВАНИЕ**

Для тестирования программы производились различные манипуляции с данными.

Проводились следующие тесты:

Тест 1 – Пустое значение в обязательном поле ввода данных, изображен на рисунке 12:

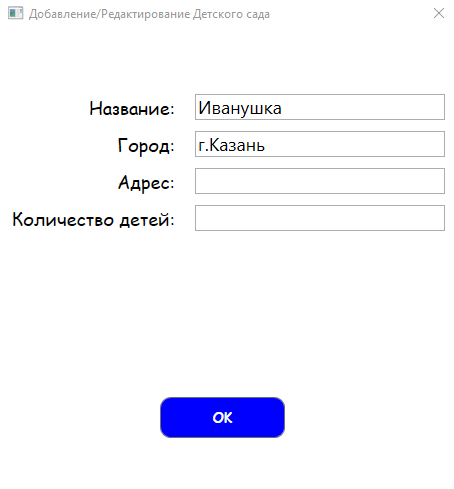


Рисунок 12 - Ввод пустого значения в поле «Название предмета»

Сообщение, которое будет показано пользователю после ввода пустого значения, представлено на рисунке 13.

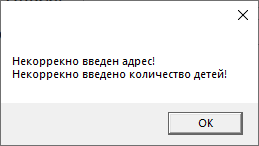


Рисунок 13 - Ошибка при вводе пустого значения

Тест 2 – Попытка ввода текста в числовое поле, изображен на рисунке 14.

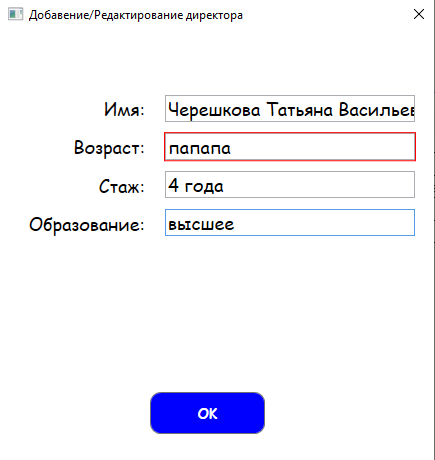


Рисунок 14 - Ввод текста в поле «Возраст»

Сообщение, которое будет показано пользователю после ввода текста вместо числового значения, представлено на рисунке 15.

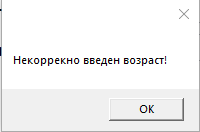


Рисунок 15 - Ввод некорректных данных

Рисунок 17 - Ввод значения которого нет в базе, привязанный к полю клиент

А также были проведены другие операции, призванные выявить ошибки в программе.

Тестирование выявило мелкие недочеты, которые впоследствии были устранены.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Основы алгоритмизации и программирования : учеб. пособие / В.Д.  Колдаев; под ред. проф. Л.Г. Гагариной. — М. : ИД «ФОРУМ» : ИНФРА-М, 2019. — 414 с. — (Среднее профессиональное образование). - Режим доступа: http://znanium.com/catalog/product/980416

2. Программирование графики на С++. Теория и примеры : учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2019. — 517 с. + Доп. материалы [Электронный ресурс; Режим доступа http://www.znanium.com].

3. Программирование на С++ с погружением: практические задания и примеры кода - М.:НИЦ ИНФРА-М, 2019. - 80 с.: 60x90 1/16 - Режим доступа: http://znanium.com/catalog/product/563294

4. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — М. : ИД «ФОРУМ» : ИНФРА-М, 2019. — 512 с. — (Среднее профессиональное образование). - Режим доступа: http://znanium.com/catalog/product/918098

5. Программирование. Процедурное программирование: Учебное пособие / Кучунова Е.В., Олейников Б.В., Чередниченко О.М. - Краснояр.:СФУ, 2019. - 92 с.: ISBN 978-5-7638-3555-7 - Режим доступа: http://znanium.com/catalog/product/978627

6. Введение в программирование на языке Visual C# : учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2018. — 447 с. — (Высшее образование: Бакалавриат). - Режим доступа: http://znanium.com/catalog/product/967691

7. Объектно-ориентированное программирование с примерами на С# : учеб. пособие / П.Б. Хорев. — М. : ФОРУМ : ИНФРА-М, 2019. — 200 с. — (Высшее образование: Бакалавриат). - Режим доступа: http://znanium.com/catalog/product/1018034

**ПРИЛОЖЕНИЕ А**

**Листинг программного модуля**

**Форма приложения MainWindow**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using DetSadNet.Pages;

using DetSadNet.Entities;

namespace DetSadNet

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Content = new SadNetPage();

}

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

{

Close();

}

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

{

MainFrame.Content = new DetSadPage();

}

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

{

MainFrame.Content = new DirectorPage();

}

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

{

MainFrame.Content = new SadNetPage();

}

}

}

Форма приложения DetSad

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using DetSadNet.Entities;

using DetSadNet.Pages;

using DetSadNet.Windows;

namespace DetSadNet.Pages

{

/// <summary>

/// Логика взаимодействия для DetSad.xaml

/// </summary>

public partial class DetSadPage : Page

{

public DetSadPage()

{

InitializeComponent();

}

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

{

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

DataGridSadNet.ItemsSource = SadNetEntities.GetContext().DetSads.ToList();

}

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

{

AddOrEditDetSads addOrEditDetSads = new AddOrEditDetSads(null);

addOrEditDetSads.Show();

}

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

{

if (DataGridSadNet.SelectedItem is DetSad detSad)

{

AddOrEditDetSads addOrEditDetSads = new AddOrEditDetSads(detSad);

addOrEditDetSads.Show();

}

}

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

{

Page\_IsVisibleChanged(null, default(DependencyPropertyChangedEventArgs));

}

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

{

if (DataGridSadNet.SelectedItem is DetSad detSad)

{

try

{

MessageBoxResult result = MessageBox.Show("Удалить данную запись?", "Warning", MessageBoxButton.YesNo, MessageBoxImage.Information);

if (result == MessageBoxResult.Yes)

{

SadNetEntities.GetContext().DetSads.Remove(detSad);

SadNetEntities.GetContext().SaveChanges();

Page\_IsVisibleChanged(null, default(DependencyPropertyChangedEventArgs));

MessageBox.Show("Запись удалена.");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

}

Форма приложения DirectorPage

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using DetSadNet.Entities;

using DetSadNet.Pages;

using DetSadNet.Windows;

namespace DetSadNet.Pages

{

/// <summary>

/// Логика взаимодействия для DirectorPage.xaml

/// </summary>

public partial class DirectorPage : Page

{

public DirectorPage()

{

InitializeComponent();

}

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

{

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

DataGridDirector.ItemsSource = SadNetEntities.GetContext().Directors.ToList();

}

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

{

AddOrrEditDirectors addOrrEditDirectors = new AddOrrEditDirectors(null);

addOrrEditDirectors.Show();

}

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

{

if (DataGridDirector.SelectedItem is Director director)

{

AddOrrEditDirectors addOrrEditDirectors = new AddOrrEditDirectors(director);

addOrrEditDirectors.Show();

}

}

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

{

Page\_IsVisibleChanged(null, default(DependencyPropertyChangedEventArgs));

}

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

{

if(DataGridDirector.SelectedItem is Director director)

{

try

{

MessageBoxResult result = MessageBox.Show("Удалить данную запись?", "Warning", MessageBoxButton.YesNo, MessageBoxImage.Information);

if (result == MessageBoxResult.Yes)

{

SadNetEntities.GetContext().Directors.Remove(director);

SadNetEntities.GetContext().SaveChanges();

Page\_IsVisibleChanged(null, default(DependencyPropertyChangedEventArgs));

MessageBox.Show("Запись удалена.");

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

}

Форма приложения SadNetPage

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using DetSadNet.Entities;

using DetSadNet.Pages;

using DetSadNet.Windows;

namespace DetSadNet.Pages

{

/// <summary>

/// Логика взаимодействия для SadNetPage.xaml

/// </summary>

public partial class SadNetPage : Page

{

public SadNetPage()

{

InitializeComponent();

}

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

{

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

DataGridSadNet.ItemsSource = SadNetEntities.GetContext().SadNets.ToList();

}

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

{

AddOrEditSadNet addOrEditSadNet = new AddOrEditSadNet(null);

addOrEditSadNet.Show();

}

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

{

if (DataGridSadNet.SelectedItem is SadNet netSad)

{

AddOrEditSadNet addOrEditSadNet = new AddOrEditSadNet(netSad);

addOrEditSadNet.Show();

}

}

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

{

Page\_IsVisibleChanged(null, default(DependencyPropertyChangedEventArgs));

}

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

{

if (DataGridSadNet.SelectedItem is SadNet netSad)

{

try

{

MessageBoxResult result = MessageBox.Show("Удалить данную запись?", "Warning", MessageBoxButton.YesNo, MessageBoxImage.Information);

if (result == MessageBoxResult.Yes)

{

SadNetEntities.GetContext().SadNets.Remove(netSad);

SadNetEntities.GetContext().SaveChanges();

Page\_IsVisibleChanged(null, default(DependencyPropertyChangedEventArgs));

MessageBox.Show("Запись удалена.");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

}

Форма приложения AddOrEditDetSads

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using DetSadNet.Entities;

namespace DetSadNet.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrEditDetSads.xaml

/// </summary>

public partial class AddOrEditDetSads : Window

{

private DetSad \_currentSad = new DetSad();

public AddOrEditDetSads(DetSad detSad)

{

InitializeComponent();

if (detSad != null)

\_currentSad = detSad;

DataContext = \_currentSad;

}

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

{

try

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

}

else

{

if (\_currentSad.SadID == 0)

SadNetEntities.GetContext().DetSads.Add(\_currentSad);

SadNetEntities.GetContext().SaveChanges();

Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private StringBuilder CheckField()

{

StringBuilder str = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentSad.Name))

str.AppendLine("Некоррекно введено название!");

if (string.IsNullOrWhiteSpace(\_currentSad.City))

str.AppendLine("Некоррекно введен город!");

if (string.IsNullOrWhiteSpace(\_currentSad.Adress))

str.AppendLine("Некоррекно введен адрес!");

if (string.IsNullOrWhiteSpace(\_currentSad.People))

str.AppendLine("Некоррекно введено количество детей!");

return str;

}

}

}

Форма приложения AddOrEditDirectors

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using DetSadNet.Entities;

namespace DetSadNet.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrrEditDirectors.xaml

/// </summary>

public partial class AddOrrEditDirectors : Window

{

private Director \_currentDirector = new Director();

public AddOrrEditDirectors(Director director)

{

InitializeComponent();

if (director != null)

\_currentDirector = director;

DataContext = \_currentDirector;

}

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

{

try

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

}

if (\_currentDirector.DirectorID == 0)

SadNetEntities.GetContext().Directors.Add(\_currentDirector);

SadNetEntities.GetContext().SaveChanges();

Close();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private StringBuilder CheckField()

{

StringBuilder str = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentDirector.DirName))

str.AppendLine("Некоррекно введено имя!");

if (\_currentDirector.DirAge == 0)

str.AppendLine("Некоррекно введен возраст!");

if (string.IsNullOrWhiteSpace(\_currentDirector.DirStage))

str.AppendLine("Некоррекно введен стаж!");

if (string.IsNullOrWhiteSpace(\_currentDirector.DirEducation))

str.AppendLine("Некоррекно введено образование!");

return str;

}

}

}

Форма приложения AddOrEditSadNet

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using DetSadNet.Entities;

namespace DetSadNet.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrEditSadNet.xaml

/// </summary>

public partial class AddOrEditSadNet : Window

{

private SadNet \_currentSadNet = new SadNet();

public AddOrEditSadNet(SadNet sadNet)

{

InitializeComponent();

if (sadNet != null)

\_currentSadNet = sadNet;

DataContext = \_currentSadNet;

ComboBoxName.ItemsSource = SadNetEntities.GetContext().DetSads.ToList();

ComboBoxDirector.ItemsSource = SadNetEntities.GetContext().Directors.ToList();

}

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

{

try

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

}

else

{

if (\_currentSadNet.SadID == 0)

SadNetEntities.GetContext().SadNets.Add(\_currentSadNet);

SadNetEntities.GetContext().SaveChanges();

Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private StringBuilder CheckField()

{

StringBuilder str = new StringBuilder();

if (ComboBoxName.SelectedItem == null)

str.AppendLine("Некоррекно введено название!");

if (ComboBoxDirector.SelectedItem == null)

str.AppendLine("Некоррекно введен город!");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentSadNet.DateOpen)))

str.AppendLine("Некоррекно введена дата открытия!");

if (\_currentSadNet.PlataForMonth == 0)

str.AppendLine("Некоррекно введенa плата!");

return str;

}

}

}

**ПРИЛОЖЕНИЕ Б**

**Диск с исполняемым кодом программного модуля**