**ИИТ БГУИР**

**Факультет повышения квалификации и переподготовки**   
  
**Дисциплина: Средства визуального программирования приложений**

**Лабораторная работа №8**

Знакомство с технологией ADO.NET Entity Framework

Вариант 3

**Выполнил слушатель гр. 40322-1** Проверила

**Гончаров Максим Евгеньевич Желакович Ирина Миролюбовна**

Минск 2025

*Цели лабораторной работы*:

Изучить возможности использования автономной модели данных на основе классов Entity Framework.

*Выполнение работы:*

Индивидуальное задание: 3. Здравоохранение.

Для заданной в индивидуальном задании предметной области, используя подход «Code First» создать базу данных, содержащую одну таблицу.

Реализовать для спроектированной БД возможность просмотра и редактирования данных с соблюдением следующих условий:

- при выводе списка всех записей в DataGrid свойство «Id» (ключевое поле) не должно отображаться;

- в элементе DataGrid заголовки столбцов должны быть на русском языке;

- текст в первой колонке выровнять по правому краю;

- строки списка должны быть пронумерованы;

- редактирование и добавление данных должно осуществляться в отдельном окне.

Создана таблица: Patients со следующими полями:

* + PatientId (int, первичный ключ)
  + LastName (string, обязательное)
  + FirstName (string, обязательное)
  + Diagnosis (string, опциональное)
  + AdmissionDate (string, опциональное)
  + DoctorInCharge (string, опциональное)

Начальное заполнение таблицы реализовано в файле DataBaseInitializer.cs:

namespace LB8.Models

{

public class DataBaseInitializer : DropCreateDatabaseIfModelChanges<EntityContext>

{

protected override void Seed(EntityContext context)

{

// Небольшая задержка для избежания конфликтов

Thread.Sleep(100);

context.Patients.AddRange(new Patient[]

{

new Patient { LastName = "Иванов", FirstName = "Алексей", Diagnosis = "Гипертоническая болезнь", AdmissionDate = "2024-01-15", DoctorInCharge = "Петрова О.В." },

new Patient { LastName = "Смирнова", FirstName = "Мария", Diagnosis = "Сахарный диабет 2 типа", AdmissionDate = "2024-02-20", DoctorInCharge = "Козлов С.И." },

new Patient { LastName = "Петров", FirstName = "Дмитрий", Diagnosis = "Острый бронхит", AdmissionDate = "2024-03-10", DoctorInCharge = "Сидорова Е.А." }

});

}

}

}

EntityContext.cs:

namespace LB8.Models

{

public class EntityContext : DbContext

{

public EntityContext() : base("DefaultConnection")

{

Database.SetInitializer(new DataBaseInitializer());

}

public DbSet<Patient> Patients { get; set; }

}

}

Класс, описывающий сущность предметной области Patient.cs:

namespace LB8.Models

{

public class Patient

{

// [Key]

public int PatientId { get; set; }

public string LastName { get; set; }

public string FirstName { get; set; }

public string Diagnosis { get; set; }

public string AdmissionDate { get; set; }

public string DoctorInCharge { get; set; }

}

Основная логика приложения описана в заоконном коде главного окна MainWindow.xaml.cs:

namespace LB8

{

public partial class MainWindow : Window

{

EntityContext context;

public MainWindow()

{

InitializeComponent();

context = new EntityContext();

LoadData();

}

private void LoadData()

{

context.Patients.Load();

dGrid.ItemsSource = context.Patients.Local;

}

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

{

var patient = new Patient();

EditWindow ew = new EditWindow(patient);

if (ew.ShowDialog() == true)

{

context.Patients.Add(patient);

context.SaveChanges();

LoadData();

}

}

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

{

if (dGrid.SelectedItem is Patient patient)

{

EditWindow ew = new EditWindow(patient);

if (ew.ShowDialog() == true)

{

context.SaveChanges();

LoadData();

}

else

{

// Откатываем изменения

context.Entry(patient).Reload();

LoadData();

}

}

}

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

{

if (dGrid.SelectedItem is Patient patient)

{

string message = $"Вы уверены, что хотите удалить пациента:\n{patient.LastName} {patient.FirstName}?";

MessageBoxResult result = MessageBox.Show(

message,

"Подтверждение удаления",

MessageBoxButton.YesNo,

MessageBoxImage.Warning,

MessageBoxResult.No);

if (result == MessageBoxResult.Yes)

{

context.Patients.Remove(patient);

context.SaveChanges();

LoadData();

MessageBox.Show("Пациент удален!", "Удаление", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

else

{

MessageBox.Show("Выберите пациента для удаления!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

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

{

LoadData();

}

private void dGrid\_LoadingRow(object sender, System.Windows.Controls.DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

protected override void OnClosed(EventArgs e)

{

context?.Dispose(); // Освобождаем контекст при закрытии окна

base.OnClosed(e);

}

}

}

При создании разметки главного окна фамилия пациента выровнена по правому краю (согласно заданию). DataGrig добавлен флаг IsReadOnly="True", для невозможности случайного редактирования. Кнопка «Удалить» вызывает MessageBox для подтверждения удаления или отмены случайного нажатия (реализовано в заоконном коде):

<Window x:Class="LB8.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Title="Учет пациентов" Height="400" Width="950">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Margin="10" Width="140">

<Button x:Name="btnAdd" Click="btnAdd\_Click" Content="Добавить" Margin="5"/>

<Button x:Name="btnDelete" Click="btnDelete\_Click" Content="Удалить" Margin="5"/>

<Button x:Name="btnEdit" Click="btnEdit\_Click" Content="Редактировать" Margin="5"/>

<Button x:Name="btnRefresh" Click="btnRefresh\_Click" Content="Обновить" Margin="5"/>

</StackPanel>

<DataGrid x:Name="dGrid" Grid.Column="1" Margin="10" AutoGenerateColumns="False"

LoadingRow="dGrid\_LoadingRow" IsReadOnly="True" RowHeaderWidth="50">

<DataGrid.Columns>

<DataGridTextColumn Header="Фамилия" Binding="{Binding LastName}" Width="120">

<DataGridTextColumn.ElementStyle>

<Style TargetType="TextBlock">

<Setter Property="TextAlignment" Value="Right"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Имя" Binding="{Binding FirstName}" Width="120"/>

<DataGridTextColumn Header="Диагноз" Binding="{Binding Diagnosis}" Width="200"/>

<DataGridTextColumn Header="Дата поступления" Binding="{Binding AdmissionDate}" Width="120"/>

<DataGridTextColumn Header="Лечащий врач" Binding="{Binding DoctorInCharge}" Width="150"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Window>

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Содержимое, созданное искусственным интеллектом, может быть неверным.

Добавление и редактирование записей реализовано с помощью дополнительного окна редактирования EditWindow.xaml:

<Window x:Class="LB8.EditWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Редактирование" Height="280" Width="280">

<Grid Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Label Content="Фамилия:" Grid.Row="0" Margin="5"/>

<Label Content="Имя:" Grid.Row="1" Margin="5"/>

<Label Content="Диагноз:" Grid.Row="2" Margin="5"/>

<Label Content="Дата:" Grid.Row="3" Margin="5"/>

<Label Content="Врач:" Grid.Row="4" Margin="5"/>

<TextBox Grid.Row="0" Grid.Column="1" Margin="5" Text="{Binding LastName}"/>

<TextBox Grid.Row="1" Grid.Column="1" Margin="5" Text="{Binding FirstName}"/>

<TextBox Grid.Row="2" Grid.Column="1" Margin="5" Text="{Binding Diagnosis}"/>

<TextBox Grid.Row="3" Grid.Column="1" Margin="5" Text="{Binding AdmissionDate}"/>

<TextBox Grid.Row="4" Grid.Column="1" Margin="5" Text="{Binding DoctorInCharge}"/>

<StackPanel Grid.Row="6" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="btnOK" Content="OK" Width="80" Margin="5" Click="btnOK\_Click"/>

<Button x:Name="btnCancel" Content="Отмена" Width="80" Margin="5" Click="btnCancel\_Click"/>

</StackPanel>

</Grid>

</Window>

EditWindow.xaml.cs:

namespace LB8

{

public partial class EditWindow : Window

{

public EditWindow(Patient patient)

{

InitializeComponent();

DataContext = patient;

}

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

{

DialogResult = true;

}

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

{

DialogResult = false;

}

}

}

Добавление записи:

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Редактирование записи:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Код проектов всех лабораторных работ, задания и отчеты находятся в публичном доступе на сайте GitHub: <https://github.com/Maxevgen555/SVPP_LB>. Все проекты находятся в одном решении и выполнены в VS2022. Файл решения SVPP\_LB.sln находится в папке \ЛБ1\LB1\_Calc\.

*Вывод:*

В ходе выполнения лабораторной работы были успешно освоены основы работы с Entity Framework в режиме Code First. Реализовано полнофункциональное приложение для учета пациентов с соблюдением всех требований задания.