**ИИТ БГУИР**

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

**Лабораторная работа №7**

Работа с базой данных в подключенном стиле

Вариант 3

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

**Гончаров Максим Евгеньевич Желакович Ирина Миролюбовна**

Минск 2025

*Цели лабораторной работы*:

Освоить возможности работы с базой данных на основе классов ADO.Net в подключенном стиле.

*Выполнение работы:*

Индивидуальное задание: 3. Здравоохранение.

Для заданной в индивидуальном задании предметной области спроектировать таблицу БД, в которой есть 4-5 полей. Таблица должна содержать допускающие присваивание значение поля NULL.

CREATE TABLE Patients (

Id INT PRIMARY KEY IDENTITY(1,1),

LastName NVARCHAR(100) NOT NULL,

FirstName NVARCHAR(100) NOT NULL,

Diagnosis NVARCHAR(255) NULL,

AdmissionDate DATE NULL,

DoctorInCharge NVARCHAR(100) NULL);

INSERT INTO Patients (LastName, FirstName, Diagnosis, AdmissionDate, DoctorInCharge) VALUES

(N'Иванов', N'Алексей', N'Гипертоническая болезнь', '2024-01-15', N'Петрова О.В.'),

(N'Смирнова', N'Мария', N'Сахарный диабет 2 типа', '2024-02-20', N'Козлов С.И.'),

(N'Петров', N'Дмитрий', N'Острый бронхит', '2024-03-10', N'Сидорова Е.А.')…

Программа должна иметь возможность отображать все записи таблицы, отображать записи, удовлетворяющие заданному критерию поиска, выполнять вставку и удаление записей. Для корректной работы установлен пакет NuGet Microsoft.Data.SqlClient 6.1.1.

<Window x:Class="LB7.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"

xmlns:local="clr-namespace:LB7"

mc:Ignorable="d"

Title="Учет пациентов" Height="600" Width="900">

<DockPanel LastChildFill="True" Background="Bisque">

<StackPanel Background="LightGreen" Width="150">

<Button x:Name="buttonView" Height="50" Margin="10" Click="buttonView\_Click">Просмотр</Button>

<Button x:Name="buttonInsert" Height="50" Margin="10" Click="buttonInsert\_Click">Вставить</Button>

<Button x:Name="buttonFind" Height="50" Margin="10" Click="buttonFind\_Click">Найти</Button>

<Button x:Name="buttonChange" Height="50" Margin="10" Click="buttonChange\_Click">Обновить</Button>

<Button x:Name="buttonDelete" Height="50" Margin="10" Click="buttonDelete\_Click">Удалить</Button>

</StackPanel>

<StackPanel x:Name="stackpanelPatient" DockPanel.Dock="Top" Background="LightSkyBlue">

<StackPanel Orientation="Horizontal">

<Label Width="200" Margin="10">Фамилия</Label>

<TextBox x:Name="textBoxLastName" Width="200" Margin="10" Text="{Binding Path=LastName}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label Width="200" Margin="10">Имя</Label>

<TextBox x:Name="textBoxFirstName" Width="200" Margin="10" Text="{Binding Path=FirstName}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label Width="200" Margin="10">Диагноз</Label>

<TextBox x:Name="textBoxDiagnosis" Width="200" Margin="10" Text="{Binding Path=Diagnosis}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label Width="200" Margin="10">Дата поступления</Label>

<TextBox x:Name="textBoxAdmissionDate" Width="200" Margin="10" Text="{Binding Path=AdmissionDate}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label Width="200" Margin="10">Лечащий врач</Label>

<TextBox x:Name="textBoxDoctor" Width="200" Margin="10" Text="{Binding Path=DoctorInCharge}"/>

</StackPanel>

<!-- Панель поиска -->

<StackPanel Orientation="Horizontal" Margin="10" Background="LightYellow">

<Label Width="100">Поиск по:</Label>

<ComboBox x:Name="comboSearchBy" Width="120" Margin="5" SelectedIndex="0">

<ComboBoxItem Content="Фамилия"/>

<ComboBoxItem Content="Имя"/>

<ComboBoxItem Content="Диагноз"/>

<ComboBoxItem Content="Врач"/>

<ComboBoxItem Content="ID"/>

</ComboBox>

<TextBox x:Name="textBoxSearch" Width="200" Margin="5" />

<Button x:Name="buttonSearch" Content="Найти" Margin="5" Click="buttonFind\_Click"/>

</StackPanel>

</StackPanel>

<ListBox x:Name="listBox" ItemsSource="{Binding}" Margin="10" Background="Beige"

SelectionChanged="listBox\_SelectionChanged"/>

</DockPanel>

</Window>

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

public static IEnumerable<Patient> getAllPatients()

{

using (var connection = GetConnection())

{

connection.Open();

var SQLstr = "SELECT \* FROM Patients";

SqlCommand getAllCommand = new SqlCommand(SQLstr, connection);

var reader = getAllCommand.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

var id = reader.GetInt32(0);

var lastName = reader.GetString(1);

var firstName = reader.GetString(2);

string diagnosis = null;

if (!reader.IsDBNull(3))

diagnosis = reader.GetString(3);

DateTime? admissionDate = null;

if (!reader.IsDBNull(4))

admissionDate = reader.GetDateTime(4);

string doctorInCharge = null;

if (!reader.IsDBNull(5))

doctorInCharge = reader.GetString(5);

yield return new Patient()

{

Id = id,

LastName = lastName,

FirstName = firstName,

Diagnosis = diagnosis,

AdmissionDate = admissionDate,

DoctorInCharge = doctorInCharge

};

}

}

}

}

// Универсальный метод поиска по любому параметру

public static List<Patient> Find(string searchTerm = null, string searchBy = "LastName")

{

var foundPatients = new List<Patient>();

using (var connection = GetConnection())

{

connection.Open();

string commandString;

SqlCommand findCommand;

if (string.IsNullOrEmpty(searchTerm))

{

// Если поисковый термин пустой, возвращаем всех пациентов

commandString = "SELECT \* FROM Patients";

findCommand = new SqlCommand(commandString, connection);

}

else

{

// Поиск по конкретному полю

commandString = $"SELECT \* FROM Patients WHERE {GetSearchField(searchBy)} LIKE @searchTerm";

findCommand = new SqlCommand(commandString, connection);

findCommand.Parameters.AddWithValue("@searchTerm", "%" + searchTerm + "%");

}

var reader = findCommand.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

var id = reader.GetInt32(0);

var lastName = reader.GetString(1);

var firstName = reader.GetString(2);

string diagnosis = null;

if (!reader.IsDBNull(3))

diagnosis = reader.GetString(3);

DateTime? admissionDate = null;

if (!reader.IsDBNull(4))

admissionDate = reader.GetDateTime(4);

string doctorInCharge = null;

if (!reader.IsDBNull(5))

doctorInCharge = reader.GetString(5);

foundPatients.Add(new Patient()

{

Id = id,

LastName = lastName,

FirstName = firstName,

Diagnosis = diagnosis,

AdmissionDate = admissionDate,

DoctorInCharge = doctorInCharge

});

}

}

}

return foundPatients;

}

// Вспомогательный метод для определения поля поиска

private static string GetSearchField(string searchBy)

{

switch (searchBy.ToLower())

{

case "lastname": return "LastName";

case "firstname": return "FirstName";

case "diagnosis": return "Diagnosis";

case "doctor": return "DoctorInCharge";

case "id": return "Id";

default: return "LastName";

}

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

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.

# SqlDataAdapter и DataTable:

# Создан новый проект LB7\_DA с подключениями через SqlDataAdapter и работой с DataTable:

public class Patient

{

static string connectionString;

static SqlConnection connection;

static SqlDataAdapter adapter;

static DataTable patientTable = new DataTable();

public int ID { 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; }

public static void NewConnection()

{

connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

connection = new SqlConnection(connectionString);

connection.Open();

}

public static DataTable ViewAll()

{

NewConnection();

string sql = "SELECT \* FROM Patients";

adapter = new SqlDataAdapter(sql, connection);

patientTable.Clear();

adapter.Fill(patientTable);

connection.Close();

return patientTable;

}

public static void Update()

{

NewConnection();

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(adapter);

adapter.Update(patientTable);

connection.Close();

}

public string Find()

{

NewConnection();

DataTable searchTable = new DataTable();

string result = "";

if (!string.IsNullOrEmpty(LastName))

{

SqlCommand command = new SqlCommand("SELECT \* FROM Patients WHERE LastName LIKE @lastName", connection);

command.Parameters.AddWithValue("@lastName", "%" + LastName + "%");

adapter = new SqlDataAdapter(command);

adapter.Fill(searchTable);

}

else if (!string.IsNullOrEmpty(FirstName))

{

SqlCommand command = new SqlCommand("SELECT \* FROM Patients WHERE FirstName LIKE @firstName", connection);

command.Parameters.AddWithValue("@firstName", "%" + FirstName + "%");

adapter = new SqlDataAdapter(command);

adapter.Fill(searchTable);

}

else if (!string.IsNullOrEmpty(Diagnosis))

{

SqlCommand command = new SqlCommand("SELECT \* FROM Patients WHERE Diagnosis LIKE @diagnosis", connection);

command.Parameters.AddWithValue("@diagnosis", "%" + Diagnosis + "%");

adapter = new SqlDataAdapter(command);

adapter.Fill(searchTable);

}

else if (!string.IsNullOrEmpty(DoctorInCharge))

{

SqlCommand command = new SqlCommand("SELECT \* FROM Patients WHERE DoctorInCharge LIKE @doctor", connection);

command.Parameters.AddWithValue("@doctor", "%" + DoctorInCharge + "%");

adapter = new SqlDataAdapter(command);

adapter.Fill(searchTable);

}

else if (!string.IsNullOrEmpty(AdmissionDate))

{

SqlCommand command = new SqlCommand("SELECT \* FROM Patients WHERE AdmissionDate = @date", connection);

command.Parameters.AddWithValue("@date", AdmissionDate);

adapter = new SqlDataAdapter(command);

adapter.Fill(searchTable);

}

foreach (DataRow row in searchTable.Rows)

{

var cells = row.ItemArray;

foreach (object cell in cells)

result += $"\t{cell}";

result += "\n";

}

connection.Close();

return string.IsNullOrEmpty(result) ? "Пациенты не найдены" : result;

}

public override string ToString()

{

return $"{ID}: {LastName} {FirstName} - {Diagnosis ?? "Диагноз не указан"}";

}

}

<Window x:Class="LB7\_DA.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"

xmlns:local="clr-namespace:LB7\_DA"

mc:Ignorable="d"

Title="Учет пациентов - DataAdapter" Height="550" Width="900">

<Grid Background="LightSkyBlue">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="150"></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" Background="LightSteelBlue">

<Button Height="50" Margin="10" Click="ButtonView\_Click">Просмотр</Button>

<Button Height="50" Margin="10" Click="ButtonUpdate\_Click">Обновить</Button>

<Button Height="50" Margin="10" Click="ButtonFind\_Click">Найти</Button>

<Button Height="50" Margin="10" Click="ButtonAdd\_Click">Добавить</Button>

<Button Height="50" Margin="10" Click="ButtonDelete\_Click">Удалить</Button>

</StackPanel>

<DataGrid AutoGenerateColumns="False" x:Name="patientGrid" Grid.Column="1" Margin="15"

Background="Beige" SelectionMode="Single" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Id}" Header="ID" Width="50"/>

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

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

<DataGridTextColumn Binding="{Binding Diagnosis}" Header="Диагноз" Width="150"/>

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

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

</DataGrid.Columns>

</DataGrid>

</Grid>

</Window>

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Созданы вспомогательные окна для поиска и редактирования:

<Window x:Class="LB7\_DA.SearchWindow"

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"

xmlns:local="clr-namespace:LB7\_DA"

mc:Ignorable="d"

Title="Поиск пациентов" Height="400" Width="450" FontSize="16">

<Grid x:Name="grid" Background="LightSalmon">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Label Content="Фамилия" Margin="10"/>

<Label Content="Имя" Margin="10" Grid.Row="1"/>

<Label Content="Диагноз" Margin="10" Grid.Row="2"/>

<Label Content="Дата поступления" Margin="10" Grid.Row="3"/>

<Label Content="Лечащий врач" Margin="10" Grid.Row="4"/>

<TextBox x:Name="textBoxLastName" Margin="10" Grid.Row="0" Grid.Column="1"

Text="{Binding Path=LastName}"/>

<TextBox x:Name="textBoxFirstName" Margin="10" Grid.Row="1" Grid.Column="1"

Text="{Binding Path=FirstName}"/>

<TextBox x:Name="textBoxDiagnosis" Margin="10" Grid.Row="2" Grid.Column="1"

Text="{Binding Path=Diagnosis}"/>

<TextBox x:Name="textBoxAdmissionDate" Margin="10" Grid.Row="3" Grid.Column="1"

Text="{Binding Path=AdmissionDate}"/>

<TextBox x:Name="textBoxDoctor" Margin="10" Grid.Row="4" Grid.Column="1"

Text="{Binding Path=DoctorInCharge}"/>

<Button x:Name="buttonCancel" Content="Отмена" Margin="10" Grid.Row="5" Click="buttonCancel\_Click" />

<Button x:Name="buttonOk" Content="Найти" Margin="10" Grid.Row="5" Grid.Column="1" Click="buttonOk\_Click"/>

</Grid>

</Window>

namespace LB7\_DA

{

public partial class SearchWindow : Window

{

public SearchWindow(Patient patient)

{

InitializeComponent();

grid.DataContext = patient;

}

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

{

this.DialogResult = true;

}

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

{

this.DialogResult = false;

}

}

}

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, число, программное обеспечение, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

<Window x:Class="LB7\_DA.EditWindow"

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"

xmlns:local="clr-namespace:LB7\_DA"

mc:Ignorable="d"

Title="Добавить пациента" Height="300" Width="500" FontSize="16">

<Grid x:Name="grid" Background="LightGreen">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="150"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

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

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

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

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

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

<TextBox x:Name="textBoxLastName" Margin="5" Grid.Row="0" Grid.Column="1"

Text="{Binding LastName}"/>

<TextBox x:Name="textBoxFirstName" Margin="5" Grid.Row="1" Grid.Column="1"

Text="{Binding FirstName}"/>

<TextBox x:Name="textBoxDiagnosis" Margin="5" Grid.Row="2" Grid.Column="1"

Text="{Binding Diagnosis}"/>

<TextBox x:Name="textBoxAdmissionDate" Margin="5" Grid.Row="3" Grid.Column="1"

Text="{Binding AdmissionDate}" ToolTip="Формат: ГГГГ-ММ-ДД"/>

<TextBox x:Name="textBoxDoctor" Margin="5" Grid.Row="4" Grid.Column="1"

Text="{Binding DoctorInCharge}"/>

<StackPanel Orientation="Horizontal" Grid.Row="5" Grid.ColumnSpan="2"

HorizontalAlignment="Center" Margin="10">

<Button x:Name="buttonOk" Content="OK" Width="80" Margin="10" Click="buttonOk\_Click"/>

<Button x:Name="buttonCancel" Content="Отмена" Width="80" Margin="10" Click="buttonCancel\_Click"/>

</StackPanel>

</Grid>

</Window>

namespace LB7\_DA

{

public partial class EditWindow : Window

{

public EditWindow(Patient patient, bool isEditMode = false)

{

InitializeComponent();

grid.DataContext = patient;

if (isEditMode)

{

Title = "Редактировать пациента";

}

}

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

{

if (string.IsNullOrEmpty(textBoxLastName.Text) || string.IsNullOrEmpty(textBoxFirstName.Text))

{

MessageBox.Show("Фамилия и имя обязательны для заполнения!");

return;

}

this.DialogResult = true;

}

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

{

this.DialogResult = false;

}

}

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

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:*

Я получил навыки работы с базой данных на основе классов ADO.Net, в том числе с использованием подключения через SqlDataAdapter и работой с DataTable.

.