Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.07 «Информационные системы и программирование»

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Выполнил:

обучающийся учебной группы № 619

\_РомановНиколай\_4ИСИП-619

*(подпись) (И.О. Фамилия)*

Проверили:

Руководители практики от колледжа:

*(должность) (И.О. Фамилия)*

*(должность) (И.О. Фамилия)*

*(должность) (И.О. Фамилия)*

*(должность) (И.О. Фамилия)*

*(оценка) (подпись)*

**Москва**

**20­­­­22**

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №1-12**

**Тема:** **РАЗРАБОТКА СТРАНИЦЫ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ.**

**СОЗДАНИЕ СЛОВАРЯ СТИЛЕЙ**

Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: Т.Г. Аксёнова

Дата: 05.09.2022

**Цель работы:** приобрести практические навыки проектирования интерфейса приложения WPF, навигации по страницам WPF, подключения базы данных с помощью технологии Entity Framework ADO.NET, реализации страницы авторизации пользователей, создания и настройки словаря стилей.

Рисунок 1. Скриншот проекта таблицы User

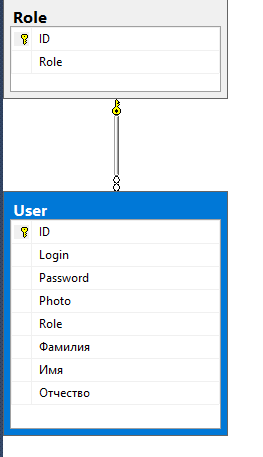


Рисунок 2. Скриншот содержимого таблицы User

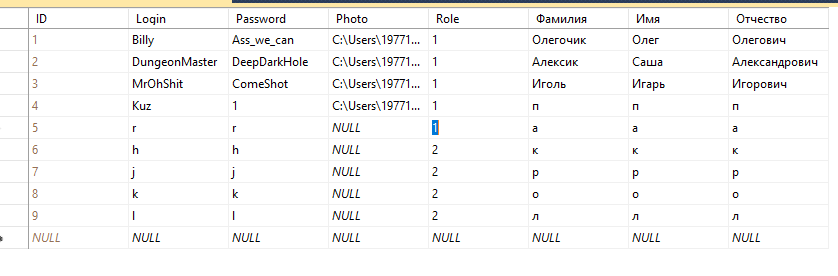


Рисунок 3. Демонстрация работы авторизации

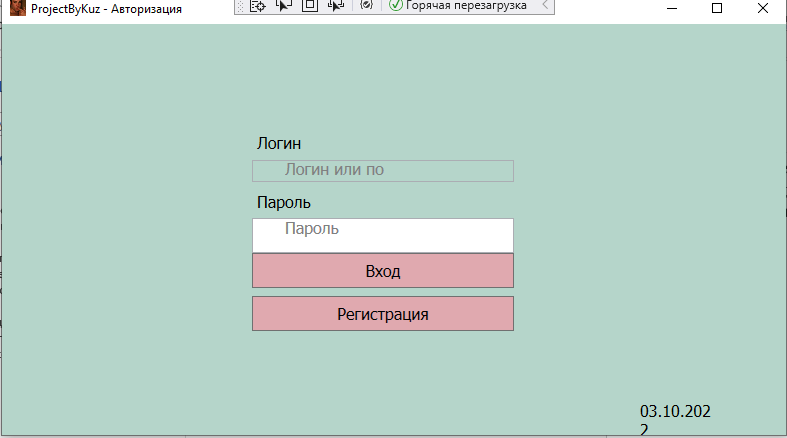


Рисунок 4. Демонстрация работы авторизации

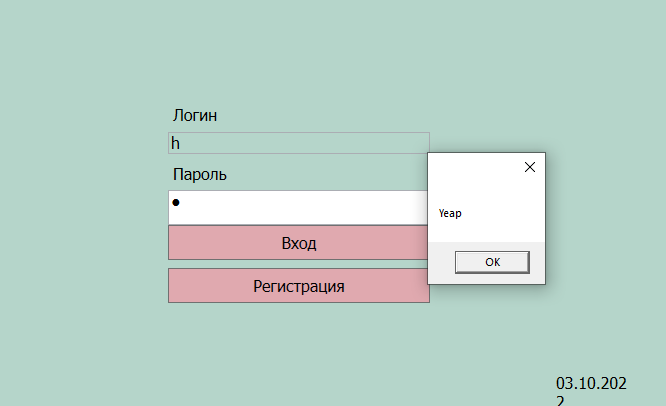


Рисунок 5. Демонстрация работы авторизации

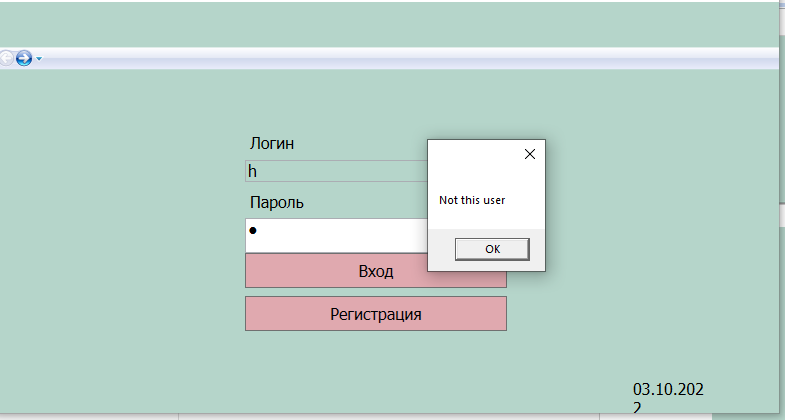
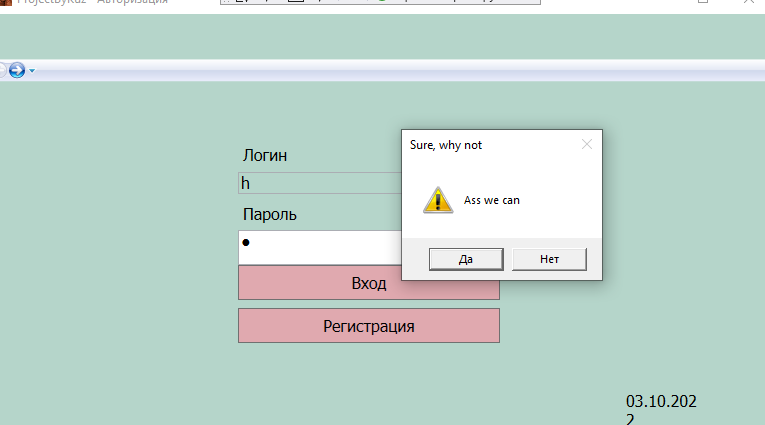


Рисунок 6. Подтверждение выхода из программы



Листинг 1. Файл MainWindow.xaml

<Window

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:\_619\_Kuznetsov\_Danila"

xmlns:Properties="clr-namespace:\_619\_Kuznetsov\_Danila.Properties" x:Class="\_619\_Kuznetsov\_Danila.MainWindow"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" Closing="Window\_Closing">

<Grid Background="#b5d5ca" Margin="0,0,-8,-6">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0\*"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="45\*"/>

<RowDefinition Height="325\*"/>

<RowDefinition Height="45\*"/>

</Grid.RowDefinitions>

<Frame x:Name="MainFrame" Content="" HorizontalAlignment="Left" Height="328" Grid.Row="1" VerticalAlignment="Top" Width="792" Grid.ColumnSpan="2" Source="Pages/AuthPage.xaml" Navigated="MainFrame\_OnNavigated"/>

<TextBlock Name="DateTime2" HorizontalAlignment="Left" Margin="638,5,0,0" Grid.Row="2" TextWrapping="Wrap" Text="TextBlock" VerticalAlignment="Top" Height="41" Width="77" Grid.Column="1"/>

<Button x:Name="ButtonBack" Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Top" Width="93" Height="35" Click="Button\_Click" Grid.ColumnSpan="2" Margin="10,5,0,0"/>

</Grid>

</Window>

Листинг 2. Файл MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Forms;

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 MessageBox = System.Windows.Forms.MessageBox;

namespace \_619\_Kuznetsov\_Danila

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

DateTime2.Text = DateTime.Now.ToString();

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

DialogResult result = System.Windows.Forms.MessageBox.Show("Ass we can", "Sure, why not", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == System.Windows.Forms.DialogResult.Yes)

{

e.Cancel = false;

}

else

{

e.Cancel = true;

}

}

private void MainFrame\_OnNavigated(object sender, NavigationEventArgs e)

{

if (!(e.Content is Page page)) return;

this.Title = $"ProjectByKuz - {page.Title}";

if (page is Pages.AuthPage)

ButtonBack.Visibility = Visibility.Hidden;

else

ButtonBack.Visibility = Visibility.Visible;

}

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

{

if (MainFrame.CanGoBack) MainFrame.GoBack();

}

}

}

Листинг 3. Файл AuthPage.xaml

<Page x:Class="\_619\_Kuznetsov\_Danila.Pages.AuthPage"

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

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

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

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

xmlns:local="clr-namespace:\_619\_Kuznetsov\_Danila.Pages"

mc:Ignorable="d"

d:DesignHeight="328" d:DesignWidth="792"

Title="Авторизация">

<Grid>

<Label Content="Логин" HorizontalAlignment="Left" Margin="250,59,0,0" VerticalAlignment="Top" RenderTransformOrigin="2,-3.615" Width="96"/>

<Label Content="Пароль" HorizontalAlignment="Left" Margin="250,118,0,0" VerticalAlignment="Top" RenderTransformOrigin="1.421,-4.385" Width="111"/>

<TextBox x:Name="TextBoxLogin" Margin="250,91,0,0" Background="Transparent" Text="" VerticalAlignment="Top" Width="262" HorizontalAlignment="Left" TextChanged="TextBoxLogin\_Changed" Height="22"/>

<PasswordBox x:Name="PasswordBox" HorizontalAlignment="Left" Margin="250,149,0,0" VerticalAlignment="Top" Width="262" PasswordChanged="PasswordBox\_Changed"/>

<Button Click="ButtonEnter\_OnClick" Content="Вход" HorizontalAlignment="Left" Margin="250,184,0,0" VerticalAlignment="Top" Width="262" ToolTip="enter"/>

<Button Content="Регистрация" HorizontalAlignment="Left" Margin="250,227,0,0" VerticalAlignment="Top" Width="262" ToolTip="Regit" Click="Button\_Click"/>

<TextBlock Text="Логин или почта" Foreground="Gray" Name="txtHintLogin" Margin="283,90,0,0" VerticalAlignment="Top" Width="99" HorizontalAlignment="Left" Height="22"/>

<TextBlock Text="Пароль" Foreground="Gray" Name="txtHintParol" Margin="283,149,0,0" VerticalAlignment="Top" Width="99" HorizontalAlignment="Left" Height="22"/>

</Grid>

</Page>

Листинг 4. Файл AuthPage.xaml.cs

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;

namespace \_619\_Kuznetsov\_Danila.Pages

{

/// <summary>

/// Логика взаимодействия для AuthPage.xaml

/// </summary>

public partial class AuthPage : Page

{

public AuthPage()

{

InitializeComponent();

}

private void TextBoxLogin\_Changed(object sender, TextChangedEventArgs e)

{

txtHintLogin.Visibility = Visibility.Visible;

if (TextBoxLogin.Text.Length > 0)

{

txtHintLogin.Visibility = Visibility.Hidden;

}

}

private void ButtonEnter\_OnClick(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(TextBoxLogin.Text) || string.IsNullOrEmpty(PasswordBox.Password))

{

MessageBox.Show("Enter your password!");

return;

}

using (var db = new Entities3())

{

var user = db.User

.AsNoTracking()

.FirstOrDefault(u => u.Login == TextBoxLogin.Text && u.Password == PasswordBox.Password);

if (user == null)

{

MessageBox.Show("Not this user");

return;

}

MessageBox.Show("Yeap");

switch (user.Role)

{

case 1:

NavigationService?.Navigate(new AdminPage());

break;

case 2:

NavigationService?.Navigate(new UserPage());

break;

}

}

}

private void PasswordBox\_Changed(object sender, RoutedEventArgs e)

{

txtHintParol.Visibility = Visibility.Visible;

if (PasswordBox.Password.Length > 0)

{

txtHintParol.Visibility = Visibility.Hidden;

}

}

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

{

NavigationService?.Navigate(new RegPage());

}

}

}

Листинг 5. Файл Dictionary.xaml

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

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

<Style TargetType="Button">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="12"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Background" Value="#FFE0A9AF"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="PasswordBox">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="ComboBox">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="DataGrid">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

</ResourceDictionary>

Листинг 6. Файл App.xaml

<Application x:Class="\_619\_Kuznetsov\_Danila.App"

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

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

xmlns:local="clr-namespace:\_619\_Kuznetsov\_Danila"

StartupUri="MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Dictionary1.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №4-6**

**Тема: Разработка страницы регистрации пользователей. Экспорт данных из базы данных в таблицу DataGrid**

Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: Т.Г. Аксёнова

Дата: 12.09.2022

**Цель работы:** приобрести практические навыки проектирования и реализации страницы регистрации пользователей в приложении WPF, научиться экспортировать данные из базы данных в таблицу DataGrid с помощью технологии ADO.NET Entity Framework.

Рисунок 1. Страница регистрации

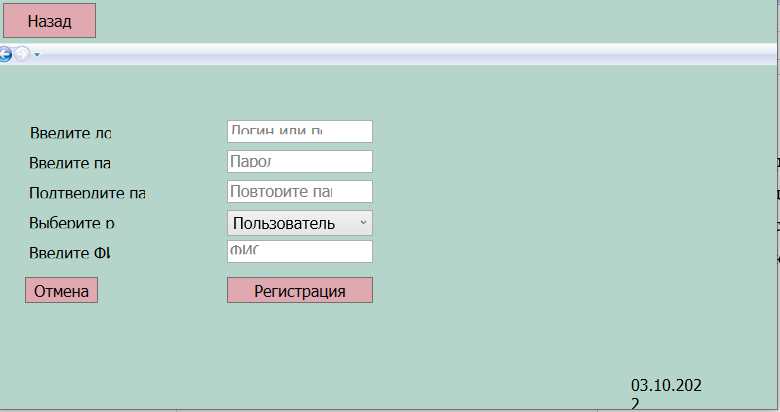


Рисунок 2. Демонстрация окна, если не заполнить поля

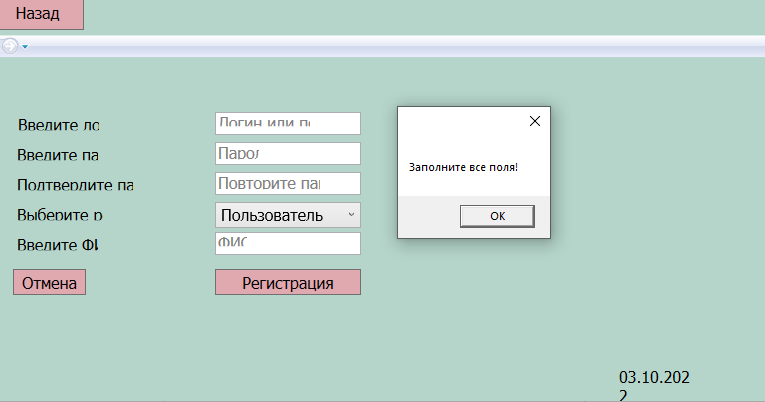


Рисунок 3. Проверка на существующего пользователя

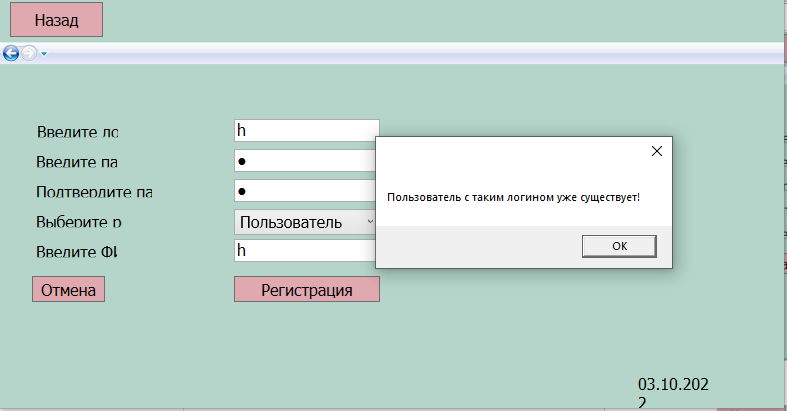


Рисунок 4. Проверка пароля

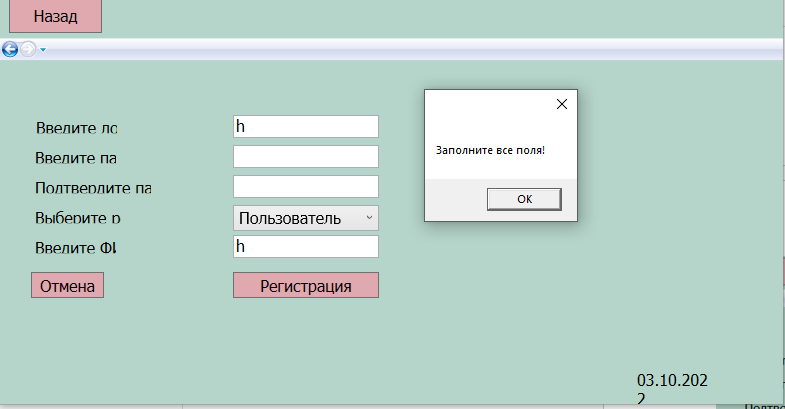


Рисунок 5. Проверка на совпадение паролей

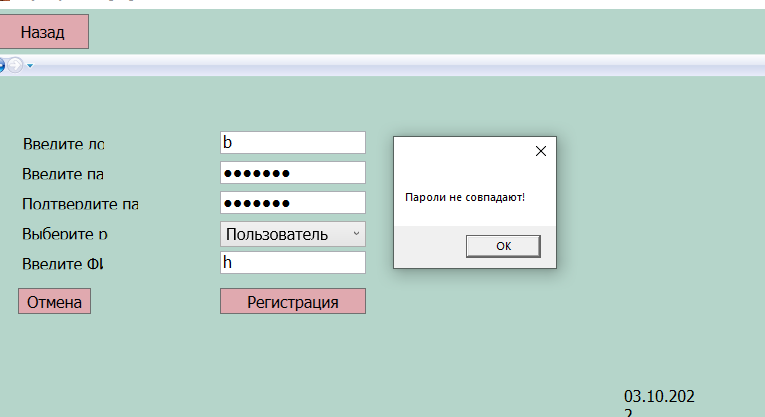


Рисунок 6. Успешная регистрация пользователя

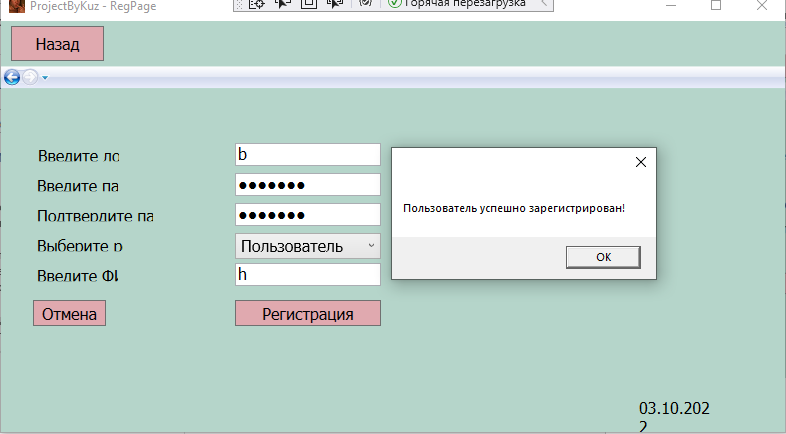


Рисунок 7. Вход под только что созданным пользователем

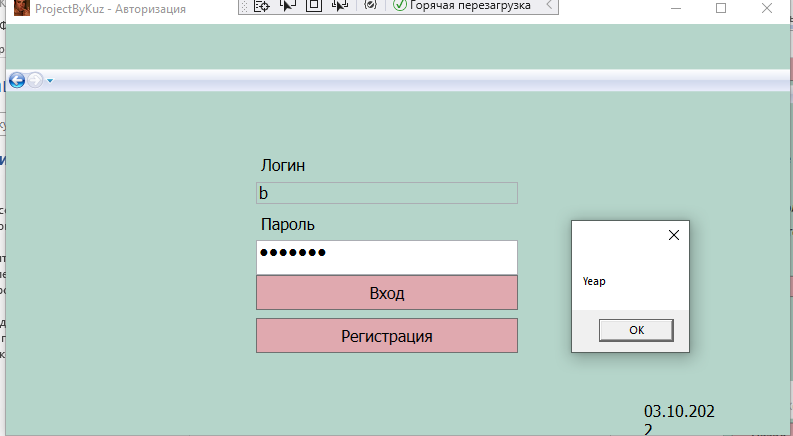


Рисунок 8. Регистрация админа

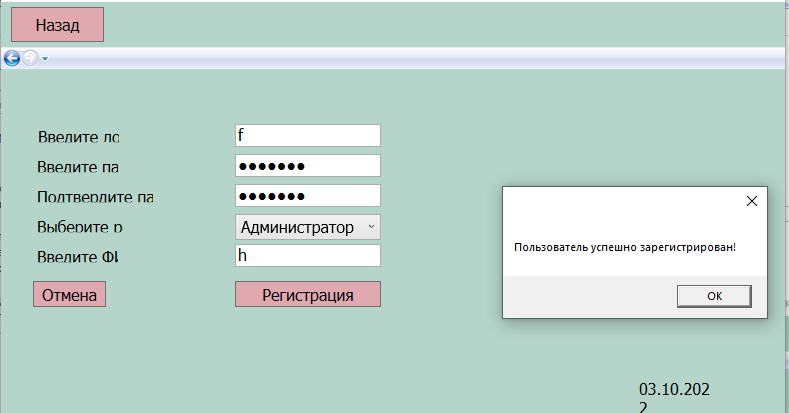
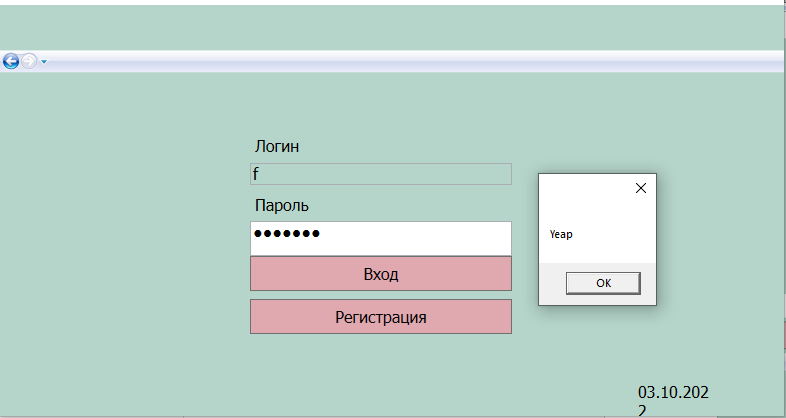


Рисунок 9. Вход под только что созданным админом

Рисунок 10. Страница админа

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

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

<Style TargetType="Button">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="12"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Background" Value="#FFE0A9AF"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="PasswordBox">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="ComboBox">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="DataGrid">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

</ResourceDictionary>

Листинг 1. Файл Dictionary.xaml

<Page x:Class="\_619\_Kuznetsov\_Danila.Pages.AdminPage"

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

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

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

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

mc:Ignorable="d"

d:DesignHeight="329" d:DesignWidth="792"

Title="Страница админа">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="377\*"/>

<RowDefinition Height="73\*"/>

</Grid.RowDefinitions>

<DataGrid Grid.Row="0" Name="DataGridUser" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Логин" Binding="{Binding Login}" Width="60"/>

<DataGridTextColumn Header="Пароль" Binding="{Binding Password}" Width="70"/>

<DataGridTextColumn Header="Роль" Binding="{Binding Role}" Width="80"/>

<DataGridTextColumn Header="Ф.И.О." Binding="{Binding Фамилия}" Width="100"/>

<DataGridTextColumn Header="Фото" Binding="{Binding Photo}" Width="100"/>

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="ButtonEdit" Content="Редактировать" ToolTip="Редактировать выбранного пользователя" Click="ButtonEdit\_Click"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Grid.Row="1" x:Name="AddButton" Content="Добавить" Margin="10,9,0,9" Height="Auto" HorizontalAlignment="Left" Width="100" Click="AddButton\_Click" ToolTip="Добавить нового пользователя"/>

<Button Grid.Row="1" x:Name="DelButton" Content="Удалить" Margin="0,9,10,9" Height="Auto" HorizontalAlignment="Right" Width="100" ToolTip="Удалить выбранного пользователя" Click="DelButton\_Click"/>

</Grid>

</Page>

Листинг 2. Файл AdminPage.xaml

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;

namespace \_619\_Kuznetsov\_Danila.Pages

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public AdminPage()

{

InitializeComponent();

DataGridUser.ItemsSource = Entities3.GetContext().User.ToList();

}

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

{

if (Visibility == System.Windows.Visibility.Visible)

{

Entities3.GetContext().ChangeTracker.Entries().ToList().ForEach(x => x.Reload());

DataGridUser.ItemsSource = Entities3.GetContext().User.ToList();

}

}

private void AddButton\_Click(object sender, System.Windows.RoutedEventArgs e)

{

NavigationService?.Navigate(new AddUserPage(null));

}

private void ButtonEdit\_Click(object sender, System.Windows.RoutedEventArgs e)

{

NavigationService?.Navigate(new AddUserPage(((sender as Button).DataContext as User)));

}

private void DelButton\_Click(object sender, System.Windows.RoutedEventArgs e)

{

var usersForRemoving = DataGridUser.SelectedItems.Cast<User>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить записи в количестве {usersForRemoving.Count()} элементов?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

Entities3.GetContext().User.RemoveRange(usersForRemoving);

Entities3.GetContext().SaveChanges();

MessageBox.Show("Данные успешно удалены!");

DataGridUser.ItemsSource = Entities3.GetContext().User.ToList();

}

catch (Exception ex)

{

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

}

}

}

}

}

Листинг 3. Файл AdminPage.xaml.cs

<Page x:Class="\_619\_Kuznetsov\_Danila.Pages.RegPage"

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

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

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

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

xmlns:local="clr-namespace:\_619\_Kuznetsov\_Danila.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="RegPage">

<Grid Background="#b5d5ca">

<Label Content="Введите логин" HorizontalAlignment="Left" VerticalAlignment="Top" Height="25" Width="91" Margin="32,53,0,0"/>

<Label Content="Введите пароль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="25" Width="91" Margin="31,83,0,0"/>

<Label Content="Подтвердите пароль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="25" Width="126" Margin="31,113,0,0"/>

<Label Content="Выберите роль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="25" Width="96" Margin="31,143,0,0"/>

<Label Content="Введите ФИО" HorizontalAlignment="Left" VerticalAlignment="Top" Height="25" Width="91" Margin="31,173,0,0"/>

<Button x:Name="CancelButton" Content="Отмена" HorizontalAlignment="Left" Margin="32,212,0,0" VerticalAlignment="Top" Width="73" Height="26" Click="CancelButton\_Click">

<Button.ToolTip>

<ToolTip Content="Кнопка для отмена"/>

</Button.ToolTip>

</Button>

<Button x:Name="RegButton" Content="Регистрация" HorizontalAlignment="Left" Margin="234,212,0,0" VerticalAlignment="Top" Width="146" Height="26" Click="RegButton\_Click">

<Button.ToolTip>

<ToolTip Content="Кнопка для регистрации"/>

</Button.ToolTip>

</Button>

<TextBox x:Name="TextBoxLogin" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="234,55,0,0" TextChanged="TextBoxLogin\_TextChanged">

<TextBox.ToolTip>

<ToolTip Content="Введите логин или почту"/>

</TextBox.ToolTip>

</TextBox>

<TextBox x:Name="TextBoxFIO" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="234,175,0,0" TextChanged="TextBoxFIO\_TextChanged">

<TextBox.ToolTip>

<ToolTip Content="Введите ФИО"/>

</TextBox.ToolTip>

</TextBox>

<PasswordBox x:Name="PasswordBox" HorizontalAlignment="Left" Margin="234,85,0,0" VerticalAlignment="Top" Width="146" Height="23" PasswordChanged="PasswordBox\_PasswordChanged">

<PasswordBox.ToolTip>

<ToolTip Content="Введите пароль"/>

</PasswordBox.ToolTip>

</PasswordBox>

<PasswordBox x:Name="PasswordBox\_Copy" HorizontalAlignment="Left" Margin="234,115,0,0" VerticalAlignment="Top" Width="146" Height="23" PasswordChanged="PasswordBox\_Copy\_PasswordChanged">

<PasswordBox.ToolTip>

<ToolTip Content="Введите пароль ещё раз"/>

</PasswordBox.ToolTip>

</PasswordBox>

<ComboBox x:Name="ComboBoxRole" SelectedIndex="0" HorizontalAlignment="Left" Margin="234,145,0,0" VerticalAlignment="Top" Width="146">

<ComboBox.ToolTip>

<ToolTip Content="Выберите роль"/>

</ComboBox.ToolTip>

<TextBlock Text="Пользователь"/>

<TextBlock Text="Администратор"/>

</ComboBox>

<TextBlock Text="Логин или почта" Foreground="Gray" x:Name="txtHintLogin" VerticalAlignment="Top" Width="92" HorizontalAlignment="Left" Height="13" Margin="237,56,0,0"/>

<TextBlock Foreground="Gray" x:Name="txtHintPassword" VerticalAlignment="Top" Width="41" HorizontalAlignment="Left" Height="16" Margin="237,86,0,0"><Run Language="ru-ru" Text="Пароль"/></TextBlock>

<TextBlock Foreground="Gray" x:Name="txtHintPassword\_Copy" VerticalAlignment="Top" Width="102" HorizontalAlignment="Left" Height="16" Margin="237,116,0,0"><Run Language="ru-ru" Text="Повторите пароль"/></TextBlock>

<TextBlock Foreground="Gray" x:Name="txtHintFIO" VerticalAlignment="Top" Width="29" HorizontalAlignment="Left" Height="13" Margin="237,176,0,0"><Run Language="ru-ru" Text="ФИО"/></TextBlock>

</Grid>

</Page>

Листинг 4. Файл RegPage.xaml

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

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;

namespace \_619\_Kuznetsov\_Danila.Pages

{

/// <summary>

/// Логика взаимодействия для RegPage.xaml

/// </summary>

public partial class RegPage : Page

{

public string RoleNumber { get; private set; }

public RegPage()

{

InitializeComponent();

}

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

{

txtHintLogin.Visibility = Visibility.Visible;

if (TextBoxLogin.Text.Length > 0)

{

txtHintLogin.Visibility = Visibility.Hidden;

}

}

private void PasswordBox\_PasswordChanged(object sender, RoutedEventArgs e)

{

txtHintPassword.Visibility = Visibility.Visible;

if (TextBoxLogin.Text.Length > 0)

{

txtHintPassword.Visibility = Visibility.Hidden;

}

}

private void PasswordBox\_Copy\_PasswordChanged(object sender, RoutedEventArgs e)

{

txtHintPassword\_Copy.Visibility = Visibility.Visible;

if (TextBoxLogin.Text.Length > 0)

{

txtHintPassword\_Copy.Visibility = Visibility.Hidden;

}

}

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

{

txtHintFIO.Visibility = Visibility.Visible;

if (TextBoxLogin.Text.Length > 0)

{

txtHintFIO.Visibility = Visibility.Hidden;

}

}

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

{

NavigationService?.Navigate(new AuthPage());

}

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

{

if (string.IsNullOrEmpty(TextBoxLogin.Text) || string.IsNullOrEmpty(PasswordBox.Password) || string.IsNullOrEmpty(PasswordBox\_Copy.Password) || string.IsNullOrEmpty(TextBoxFIO.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

using (var db = new Entities3())

{

var user = db.User.AsNoTracking().FirstOrDefault(u => u.Login == TextBoxLogin.Text);

if (user != null)

{

MessageBox.Show("Пользователь с таким логином уже существует!");

return;

}

if (!Regex.Match(PasswordBox.Password, "^(?=.\*?[a-z])(?=.\*?[0-9]).{6,}$").Success)

{

MessageBox.Show("Пароль должен состоять минимум из 6 символов, а также иметь только английские буквы и хотя бы одну цифру!");

return;

}

if (PasswordBox.Password != PasswordBox\_Copy.Password)

{

MessageBox.Show("Пароли не совпадают!");

return;

}

if (ComboBoxRole.SelectedIndex == 0)

RoleNumber = "2";

else RoleNumber = "1";

User userObject = new User

{

Фамилия = TextBoxFIO.Text,

Login = TextBoxLogin.Text,

Password = PasswordBox.Password,

Role = Convert.ToInt32(RoleNumber)

};

db.User.Add(userObject);

db.SaveChanges();

MessageBox.Show("Пользователь успешно зарегистрирован!");

}

}

}

}

Листинг 5. Файл RegPage.xaml.cs

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №7-9**

**Тема: Манипулирование данными из базы данных в таблице DataGrid**

Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: Т.Г. Аксёнова

Дата: 19.09.2022

**Цель работы:** приобрести практические навыки добавления, удаления и редактирования данных в приложении WPF с помощью таблицы DataGrid.

Рисунок 1. Страница добавления пользователя

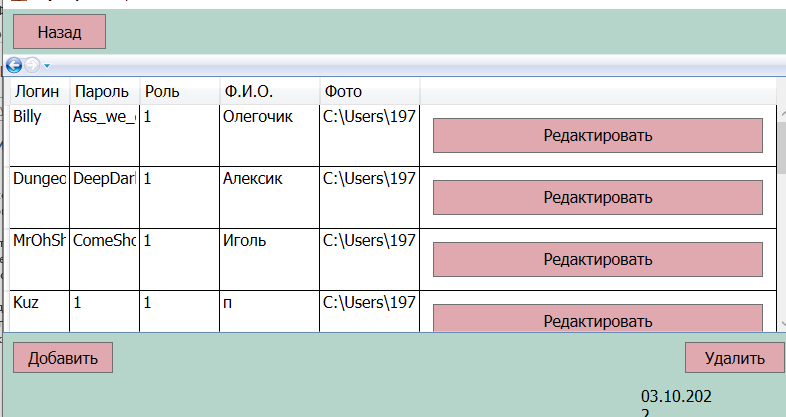


Рисунок 2. Проверка работы добавления нового пользователя

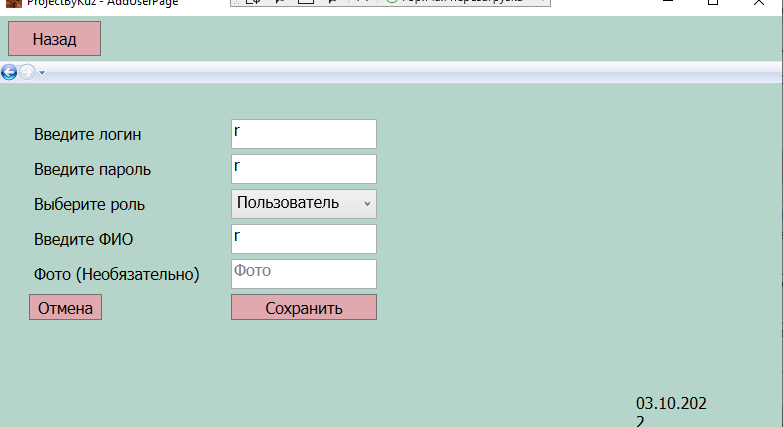


Рисунок 3. Проверка заполнения полей

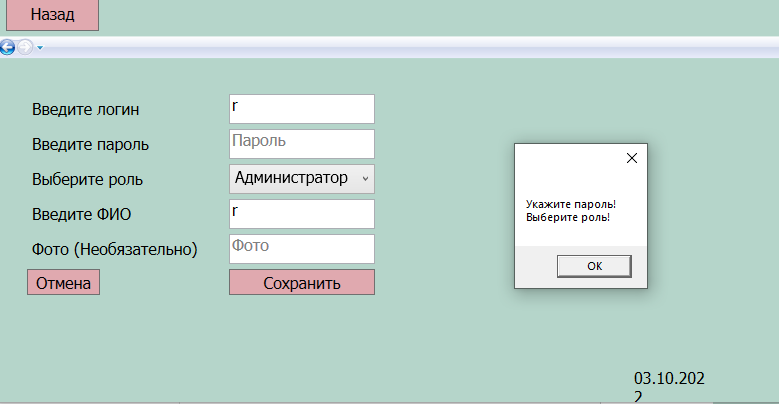


Рисунок 4. Страница редактирования пользователя

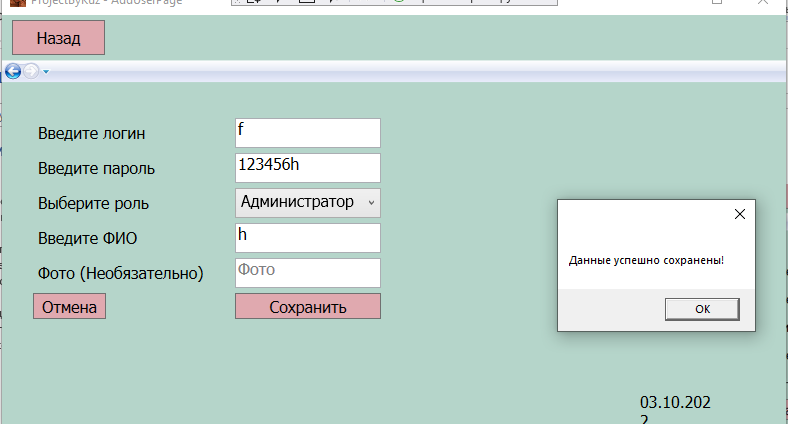


Рисунок 5. Проверка работы редактирования пользователя

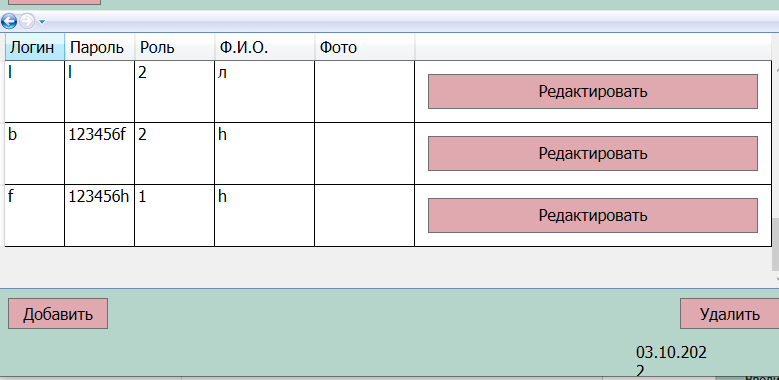


Рисунок 6. Удаление пользователя

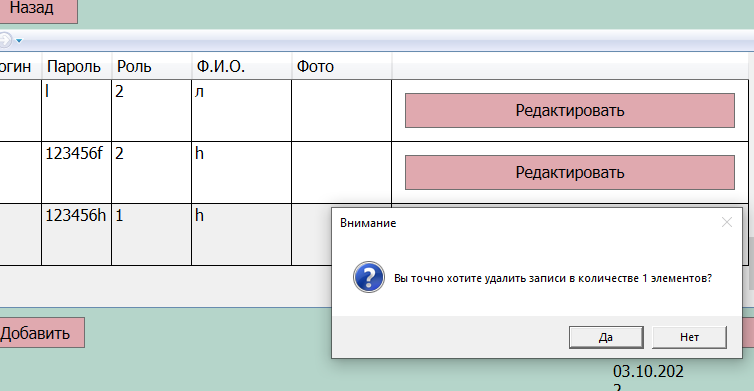


Рисунок 7. Проверка удаления пользователя

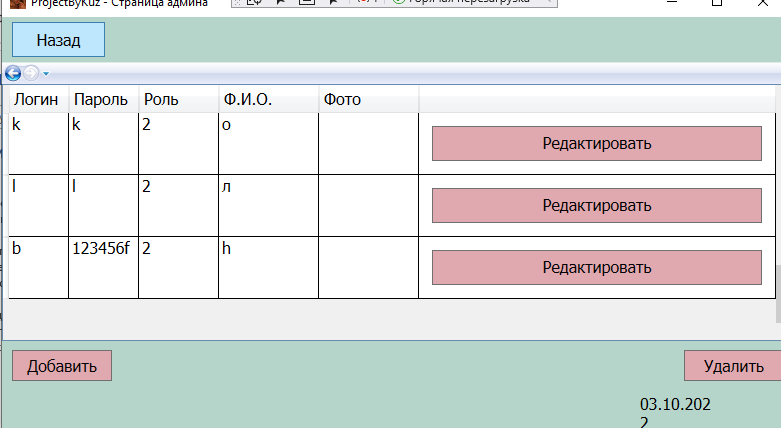


Рисунок 8. Проверка удаления пользователя

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

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

<Style TargetType="Button">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="12"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Background" Value="#FFE0A9AF"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="PasswordBox">

<Setter Property="Height" Value="35"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="ComboBox">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="DataGrid">

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

</ResourceDictionary>

Листинг 1. Файл Dictionary.xaml

<Page x:Class="\_619\_Kuznetsov\_Danila.Pages.AddUserPage"

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

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

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

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

xmlns:local="clr-namespace:\_619\_Kuznetsov\_Danila.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddUserPage">

<Grid Background="#b5d5ca">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="210"/>

<ColumnDefinition Width="210"/>

</Grid.ColumnDefinitions>

<Label Content="Введите логин" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,36,0,0"/>

<Label Content="Введите пароль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,71,0,0"/>

<Label Content="Выберите роль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,106,0,0"/>

<Label Content="Введите ФИО" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,141,0,0"/>

<Button x:Name="CancelButton" Content="Отмена" HorizontalAlignment="Left" Margin="31,211,0,0" VerticalAlignment="Top" Width="73" Height="26" Click="CancelButton\_Click">

<Button.ToolTip>

<ToolTip Content="Кнопка для отмены"/>

</Button.ToolTip>

</Button>

<Button x:Name="SaveButton" Content="Сохранить" HorizontalAlignment="Left" Margin="23,211,0,0" VerticalAlignment="Top" Width="146" Height="26" Click="SaveButton\_Click" Grid.Column="1">

<Button.ToolTip>

<ToolTip Content="Кнопка для сохранения данных"/>

</Button.ToolTip>

</Button>

<TextBox x:Name="TextBoxLogin" Text="{Binding Login}" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="23,36,0,0" TextChanged="TextBoxLogin\_TextChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите логин или почту"/>

</TextBox.ToolTip>

</TextBox>

<TextBox x:Name="PasswordBox" Text="{Binding Password}" HorizontalAlignment="Left" Margin="23,71,0,0" VerticalAlignment="Top" Width="146" Height="30" TextChanged="PasswordBox\_PasswordChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите пароль"/>

</TextBox.ToolTip>

</TextBox>

<ComboBox x:Name="ComboBoxRole" SelectedItem="{Binding Role.Role}" HorizontalAlignment="Left" Margin="23,106,0,0" VerticalAlignment="Top" Width="146" Grid.Column="1" Height="30">

<ComboBox.ToolTip>

<ToolTip Content="Выберите роль"/>

</ComboBox.ToolTip>

<TextBlock Text="Пользователь"/>

<TextBlock Text="Администратор"/>

</ComboBox>

<TextBlock Text="Логин или почта" Foreground="Gray" x:Name="txtHintLogin" VerticalAlignment="Top" Width="127" HorizontalAlignment="Left" Height="18" Margin="26,37,0,0" Grid.Column="1"/>

<TextBlock Foreground="Gray" x:Name="txtHintPassword" VerticalAlignment="Top" Width="55" HorizontalAlignment="Left" Height="20" Margin="26,72,0,0" Grid.Column="1"><Run Language="ru-ru" Text="Пароль"/></TextBlock>

<Label Content="Фото (Необязательно)" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="179" Margin="31,176,0,0"/>

<TextBox x:Name="TextBoxPhoto" Text="{Binding Photo}" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="23,176,0,0" TextChanged="TextBoxPhoto\_TextChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите путь к фотографии"/>

</TextBox.ToolTip>

</TextBox>

<TextBlock Foreground="Gray" x:Name="txtHintPhoto" VerticalAlignment="Top" Width="42" HorizontalAlignment="Left" Height="17" Margin="26,177,0,0" Grid.Column="1"><Run Language="ru-ru" Text="Фото"/></TextBlock>

<TextBox x:Name="TextBoxFIO" Text="{Binding Фамилия}" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="23,141,0,0" TextChanged="TextBoxFIO\_TextChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите ФИО"/>

</TextBox.ToolTip>

</TextBox>

<TextBlock Foreground="Gray" x:Name="txtHintFIO" VerticalAlignment="Top" Width="36" HorizontalAlignment="Left" Height="18" Margin="26,143,0,0" Grid.Column="1"><Run Language="ru-ru" Text="ФИО"/></TextBlock>

</Grid>

</Page>

Листинг 2. Файл AddUserPage.xaml

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;

namespace \_619\_Kuznetsov\_Danila.Pages

{

/// <summary>

/// Логика взаимодействия для AuthPage.xaml

/// </summary>

public partial class AuthPage : Page

{

public AuthPage()

{

InitializeComponent();

}

private void TextBoxLogin\_Changed(object sender, TextChangedEventArgs e)

{

txtHintLogin.Visibility = Visibility.Visible;

if (TextBoxLogin.Text.Length > 0)

{

txtHintLogin.Visibility = Visibility.Hidden;

}

}

private void ButtonEnter\_OnClick(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(TextBoxLogin.Text) || string.IsNullOrEmpty(PasswordBox.Password))

{

MessageBox.Show("Enter your password!");

return;

}

using (var db = new Entities3())

{

var user = db.User

.AsNoTracking()

.FirstOrDefault(u => u.Login == TextBoxLogin.Text && u.Password == PasswordBox.Password);

if (user == null)

{

MessageBox.Show("Not this user");

return;

}

MessageBox.Show("Yeap");

switch (user.Role)

{

case 1:

NavigationService?.Navigate(new AdminPage());

break;

case 2:

NavigationService?.Navigate(new UserPage());

break;

}

}

}

private void PasswordBox\_Changed(object sender, RoutedEventArgs e)

{

txtHintParol.Visibility = Visibility.Visible;

if (PasswordBox.Password.Length > 0)

{

txtHintParol.Visibility = Visibility.Hidden;

}

}

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

{

NavigationService?.Navigate(new RegPage());

}

}

}

Листинг 3. Файл AddUserPage.xaml.cs

<Page x:Class="\_619\_Kuznetsov\_Danila.Pages.AddUserPage"

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

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

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

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

xmlns:local="clr-namespace:\_619\_Kuznetsov\_Danila.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddUserPage">

<Grid Background="#b5d5ca">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="210"/>

<ColumnDefinition Width="210"/>

</Grid.ColumnDefinitions>

<Label Content="Введите логин" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,36,0,0"/>

<Label Content="Введите пароль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,71,0,0"/>

<Label Content="Выберите роль" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,106,0,0"/>

<Label Content="Введите ФИО" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="170" Margin="31,141,0,0"/>

<Button x:Name="CancelButton" Content="Отмена" HorizontalAlignment="Left" Margin="31,211,0,0" VerticalAlignment="Top" Width="73" Height="26" Click="CancelButton\_Click">

<Button.ToolTip>

<ToolTip Content="Кнопка для отмены"/>

</Button.ToolTip>

</Button>

<Button x:Name="SaveButton" Content="Сохранить" HorizontalAlignment="Left" Margin="23,211,0,0" VerticalAlignment="Top" Width="146" Height="26" Click="SaveButton\_Click" Grid.Column="1">

<Button.ToolTip>

<ToolTip Content="Кнопка для сохранения данных"/>

</Button.ToolTip>

</Button>

<TextBox x:Name="TextBoxLogin" Text="{Binding Login}" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="23,36,0,0" TextChanged="TextBoxLogin\_TextChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите логин или почту"/>

</TextBox.ToolTip>

</TextBox>

<TextBox x:Name="PasswordBox" Text="{Binding Password}" HorizontalAlignment="Left" Margin="23,71,0,0" VerticalAlignment="Top" Width="146" Height="30" TextChanged="PasswordBox\_PasswordChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите пароль"/>

</TextBox.ToolTip>

</TextBox>

<ComboBox x:Name="ComboBoxRole" SelectedItem="{Binding Role.Role}" HorizontalAlignment="Left" Margin="23,106,0,0" VerticalAlignment="Top" Width="146" Grid.Column="1" Height="30">

<ComboBox.ToolTip>

<ToolTip Content="Выберите роль"/>

</ComboBox.ToolTip>

<TextBlock Text="Пользователь"/>

<TextBlock Text="Администратор"/>

</ComboBox>

<TextBlock Text="Логин или почта" Foreground="Gray" x:Name="txtHintLogin" VerticalAlignment="Top" Width="127" HorizontalAlignment="Left" Height="18" Margin="26,37,0,0" Grid.Column="1"/>

<TextBlock Foreground="Gray" x:Name="txtHintPassword" VerticalAlignment="Top" Width="55" HorizontalAlignment="Left" Height="20" Margin="26,72,0,0" Grid.Column="1"><Run Language="ru-ru" Text="Пароль"/></TextBlock>

<Label Content="Фото (Необязательно)" HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="179" Margin="31,176,0,0"/>

<TextBox x:Name="TextBoxPhoto" Text="{Binding Photo}" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="23,176,0,0" TextChanged="TextBoxPhoto\_TextChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите путь к фотографии"/>

</TextBox.ToolTip>

</TextBox>

<TextBlock Foreground="Gray" x:Name="txtHintPhoto" VerticalAlignment="Top" Width="42" HorizontalAlignment="Left" Height="17" Margin="26,177,0,0" Grid.Column="1"><Run Language="ru-ru" Text="Фото"/></TextBlock>

<TextBox x:Name="TextBoxFIO" Text="{Binding Фамилия}" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="146" Margin="23,141,0,0" TextChanged="TextBoxFIO\_TextChanged" Grid.Column="1">

<TextBox.ToolTip>

<ToolTip Content="Введите ФИО"/>

</TextBox.ToolTip>

</TextBox>

<TextBlock Foreground="Gray" x:Name="txtHintFIO" VerticalAlignment="Top" Width="36" HorizontalAlignment="Left" Height="18" Margin="26,143,0,0" Grid.Column="1"><Run Language="ru-ru" Text="ФИО"/></TextBlock>

</Grid>

</Page>

Листинг 4. Файл AdminPage.xaml

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;

namespace \_619\_Kuznetsov\_Danila.Pages

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public AdminPage()

{

InitializeComponent();

DataGridUser.ItemsSource = Entities3.GetContext().User.ToList();

}

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

{

if (Visibility == System.Windows.Visibility.Visible)

{

Entities3.GetContext().ChangeTracker.Entries().ToList().ForEach(x => x.Reload());

DataGridUser.ItemsSource = Entities3.GetContext().User.ToList();

}

}

private void AddButton\_Click(object sender, System.Windows.RoutedEventArgs e)

{

NavigationService?.Navigate(new AddUserPage(null));

}

private void ButtonEdit\_Click(object sender, System.Windows.RoutedEventArgs e)

{

NavigationService?.Navigate(new AddUserPage(((sender as Button).DataContext as User)));

}

private void DelButton\_Click(object sender, System.Windows.RoutedEventArgs e)

{

var usersForRemoving = DataGridUser.SelectedItems.Cast<User>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить записи в количестве {usersForRemoving.Count()} элементов?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

Entities3.GetContext().User.RemoveRange(usersForRemoving);

Entities3.GetContext().SaveChanges();

MessageBox.Show("Данные успешно удалены!");

DataGridUser.ItemsSource = Entities3.GetContext().User.ToList();

}

catch (Exception ex)

{

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

}

}

}

}

}

Листинг 5. Файл AdminPage.xaml.cs

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №10-12**

**Тема: Экспорт данных из базы данных в список ListView. Поиск и фильтрация данных**

Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: Т.Г. Аксёнова

Дата: 26.09.2022

**Цель работы:** приобрести практические навыки экспорта данных из базы данных в список ListView, а также поиска и фильтрации данных.

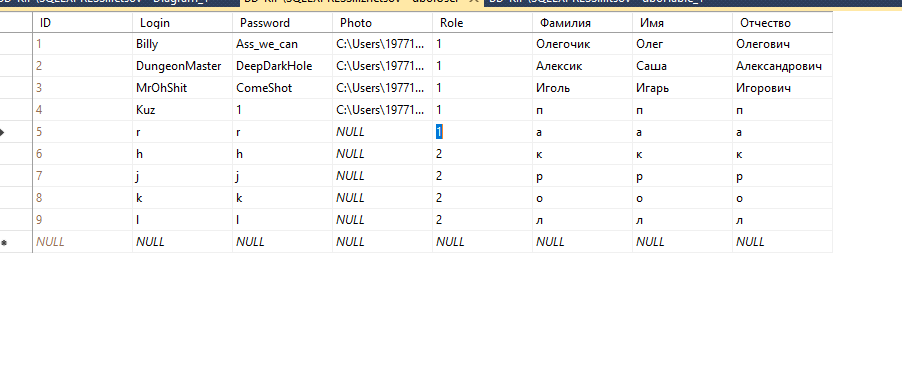


Рисунок 1. Данные в БД

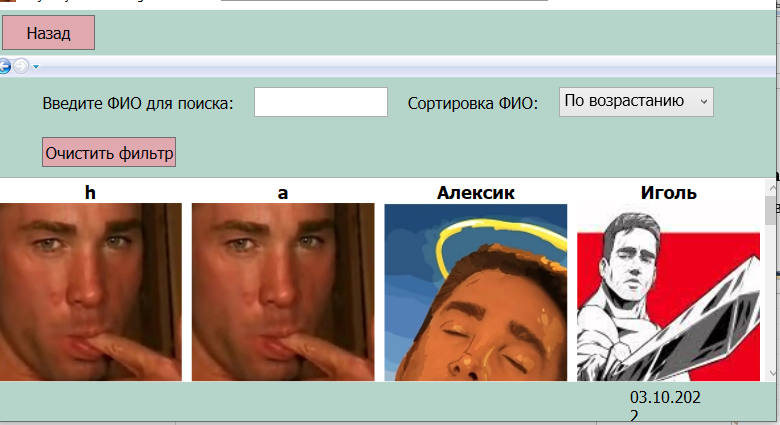


Рисунок 2. Страница пользователя

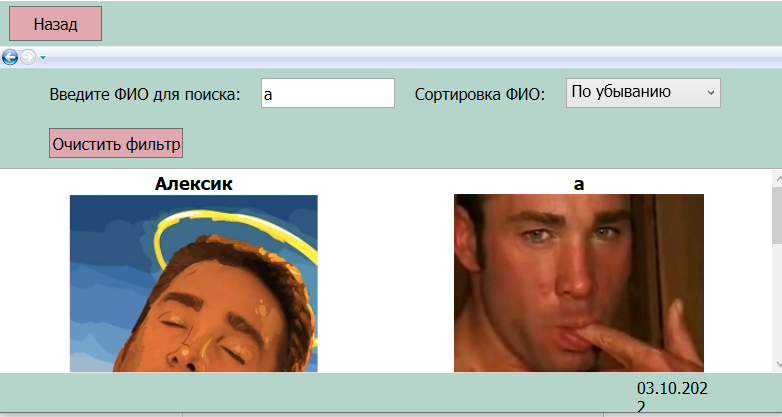
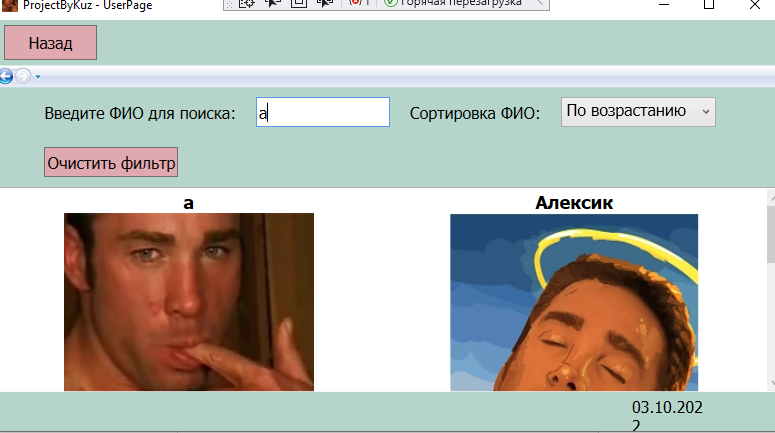


Рисунок 3. Страница пользователя с фотографиями пользователей

Рисунок 4. Проверка работы сортировки по пользователям

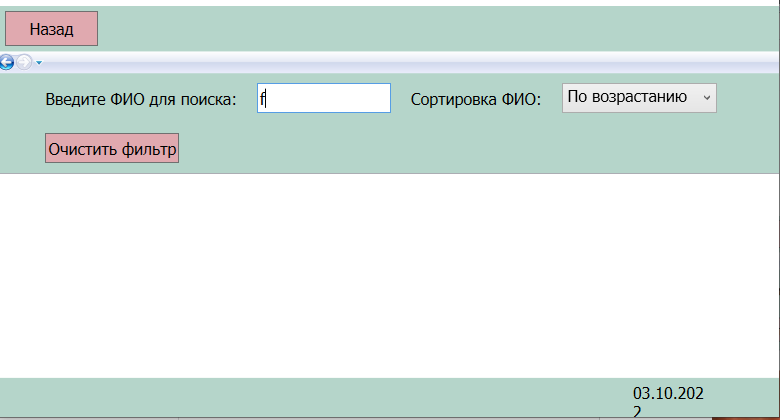


Рисунок 5. Проверка работы сортировки по ФИО

<Page x:Class="\_619\_Kuznetsov\_Danila.Pages.UserPage"

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

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

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

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

xmlns:local="clr-namespace:\_619\_Kuznetsov\_Danila.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="UserPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel HorizontalAlignment="Center">

<TextBlock Text="Введите ФИО для поиска:" VerticalAlignment="Center" Margin="0 10 20 10" />

<TextBox x:Name="TextBoxФамилия" VerticalContentAlignment="Center" Margin="0 10 20 10" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="134" Grid.Row="0" TextChanged="TextBoxФамилия\_TextChanged">

<TextBox.ToolTip>

<ToolTip Content="Введите ФИО"/>

</TextBox.ToolTip>

</TextBox>

<TextBlock Text="Сортировка ФИО:" VerticalAlignment="Center" Margin="0 10 20 10" />

<ComboBox x:Name="ComboBoxTypeSort" Margin="0 10 20 10" Text="Только пользователи" HorizontalAlignment="Left" Height="30" VerticalAlignment="Top" Width="155" Grid.Row="0" SelectionChanged="ComboBoxTypeSort\_SelectionChanged">

<ComboBox.ToolTip>

<ToolTip Content="Выберите тип сортировки"/>

</ComboBox.ToolTip>

<TextBlock Text="По возрастанию"/>

<TextBlock Text="По убыванию"/>

</ComboBox>

<Button x:Name="ClearButton" Margin="0 10 20 10" Content="Очистить фильтр" Height="30" Width="134" Grid.Row="0" Click="ClearButton\_Click"/>

</WrapPanel>

<ListView Name="ListUser" Grid.Row="1" ScrollViewer.HorizontalScrollBarVisibility="Disabled" HorizontalContentAlignment="Center">

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<UniformGrid HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"></RowDefinition>

<RowDefinition Height="250"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

</Grid.RowDefinitions>

<Image Width="250" Height="250" Grid.Row="1" Stretch="UniformToFill" HorizontalAlignment="Center" VerticalAlignment="Top">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>/Photo/2.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock VerticalAlignment="Center" TextAlignment="Center" Width="450" Grid.Row="0" TextWrapping="Wrap" Text="{Binding Фамилия}" HorizontalAlignment="Center" FontSize="18" FontWeight="Bold"/>

<TextBlock VerticalAlignment="Center" TextAlignment="Center" Width="450" Grid.Row="2" HorizontalAlignment="Center" FontSize="18" Text="{Binding Role}"/>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

Листинг 1. Файд UserPage.xaml

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;

namespace \_619\_Kuznetsov\_Danila.Pages

{

/// <summary>

/// Логика взаимодействия для UserPage.xaml

/// </summary>

public partial class UserPage : Page

{

public UserPage()

{

InitializeComponent();

var currentUsers = Entities3.GetContext().User.ToList();

ListUser.ItemsSource = currentUsers;

ComboBoxTypeSort.SelectedIndex = 0;

UpdateUsers();

}

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

{

UpdateUsers();

}

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

{

UpdateUsers();

}

private void CheckBoxRole\_Unchecked(object sender, RoutedEventArgs e)

{

UpdateUsers();

}

private void UpdateUsers()

{

var currentUsers = Entities3.GetContext().User.ToList();

currentUsers = currentUsers.Where(x => x.Фамилия.ToLower().Contains(TextBoxФамилия.Text.ToLower())).ToList();

if (ComboBoxTypeSort.SelectedIndex == 0)

ListUser.ItemsSource = currentUsers.OrderBy(x => x.Фамилия).ToList();

else ListUser.ItemsSource = currentUsers.OrderByDescending(x => x.Фамилия).ToList();

}

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

{

ComboBoxTypeSort.SelectedIndex = 0;

TextBoxФамилия.Text = null;

UpdateUsers();

}

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

{

UpdateUsers();

}

}

}

Листинг 2. Файл UserPage.xaml.cs

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №13-18**

**Тема:** **РАЗРАБОТКА СТРАНИЦЫ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ.**

**СОЗДАНИЕ СЛОВАРЯ СТИЛЕЙ**

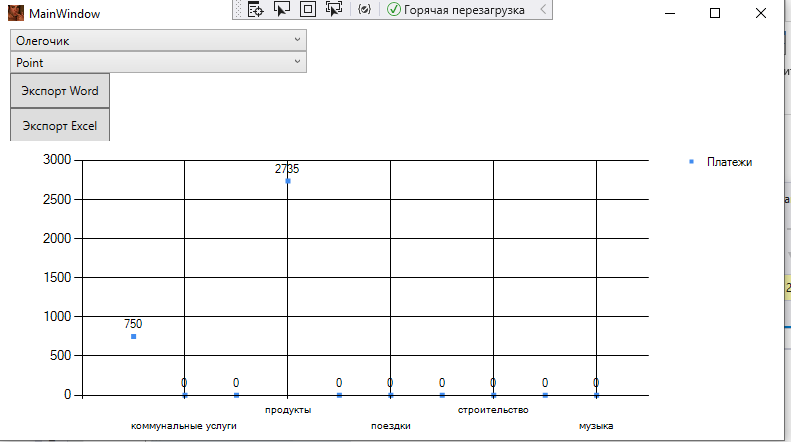
Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: Т.Г. Аксёнова

Дата: 05.09.2022

Рисунок 1. Страница диаграммы



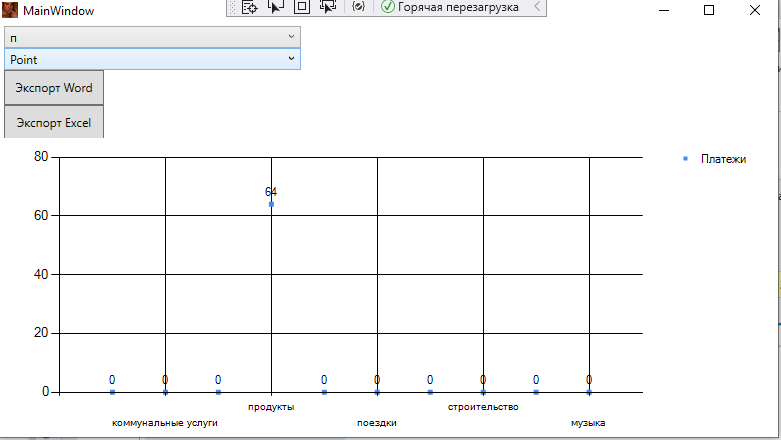
Рисуонок 2. Страница диаграммы с другим пользователем

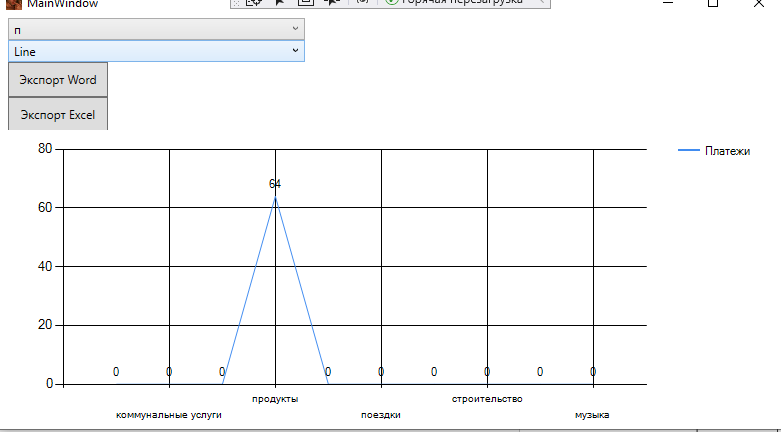
Рисунок 3. Страница с другим типом диаграммы 

Рисунок 4. Экспорт Excel

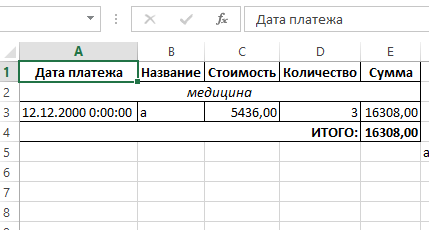


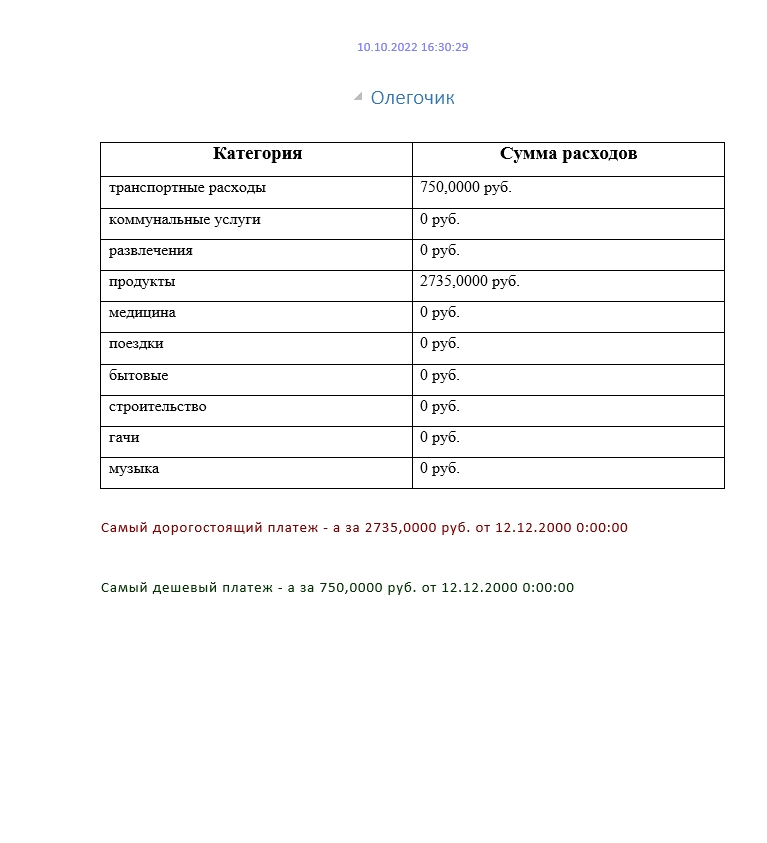
Рисунок 5. Экспорт Word

Рисунок 6. MainWindow

<Window x:Class="WpfApp1.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:WpfApp1"

mc:Ignorable="d"

xmlns:charts="clr-namespace:System.Windows.Forms.DataVisualization.Charting;assembly=System.Windows.Forms.DataVisualization"

Title="MainWindow" Height="450" Width="800" Closing="Window\_Closing">

<Grid Grid.Row="0">

<Grid.RowDefinitions>

<RowDefinition Height="70\*"/>

<RowDefinition Height="0\*"/>

<RowDefinition Height="349\*"/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Left" Height="113" Margin="10,0,0,0" VerticalAlignment="Top" Width="772" Grid.RowSpan="3">

<ComboBox x:Name="CmbUser" DisplayMemberPath="Фамилия" Margin="0,0,475,0" SelectionChanged="UpdateChart"/>

<ComboBox x:Name="CmbDiagram" SelectionChanged="UpdateChart" Margin="0,0,475,0"/>

<Button Content="Экспорт Word" HorizontalAlignment="Left" VerticalAlignment="Top" Width="100" Height="35" Click="Button\_Click"/>

<Button Content="Экспорт Excel" HorizontalAlignment="Left" VerticalAlignment="Top" Width="100" Height="35" Click="Button\_Click\_1"/>

</StackPanel>

<WindowsFormsHost HorizontalAlignment="Left" Height="306" Grid.Row="2" VerticalAlignment="Top" Width="792" Margin="0,43,0,0">

<charts:Chart x:Name="ChartPayments">

<charts:Chart.Legends>

<charts:Legend>

</charts:Legend>

</charts:Chart.Legends>

</charts:Chart>

</WindowsFormsHost>

</Grid>

</Window>

Рисунок 7. MainWindow.cs

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 System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private Entities1 \_context = new Entities1();

public MainWindow()

{

InitializeComponent();

ChartPayments.ChartAreas.Add(new ChartArea("Main"));

var currentSeries = new Series("Платежи")

{

IsValueShownAsLabel = true

};

ChartPayments.Series.Add(currentSeries);

CmbUser.ItemsSource = \_context.User.ToList();

CmbDiagram.ItemsSource = Enum.GetValues(typeof(SeriesChartType));

CmbUser.SelectedIndex = 0;

CmbDiagram.SelectedIndex = 0;

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

DialogResult result = System.Windows.Forms.MessageBox.Show("Ass we can", "Sure, why not", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == System.Windows.Forms.DialogResult.Yes)

{

e.Cancel = false;

}

else

{

e.Cancel = true;

}

}

private void UpdateChart(object sender, SelectionChangedEventArgs e)

{

if (CmbUser.SelectedItem is User currentUser && CmbDiagram.SelectedItem is SeriesChartType currentType)

{

Series currentSeries = ChartPayments.Series.FirstOrDefault();

currentSeries.ChartType = currentType;

currentSeries.Points.Clear();

var categoriesList = \_context.Category.ToList();

foreach (var category in categoriesList)

{

currentSeries.Points.AddXY(category.Name,

\_context.Payment.ToList().Where(u => u.User == currentUser

&& u.Category == category).Sum(u => u.Price \* u.Num));

}

}

}

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

{

int a = 0;

var allUsers = \_context.User.ToList();

var allCategories = \_context.Category.ToList();

var application = new Word.Application();

Word.Document document = application.Documents.Add();

foreach (var user in allUsers)

{

a++;

Word.Paragraph userParagraph = document.Paragraphs.Add();

Word.Range userRange = userParagraph.Range;

userRange.Text = user.Фамилия;

userParagraph.set\_Style("Заголовок 1");

userRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

userRange.InsertParagraphAfter();

document.Paragraphs.Add();

Word.Paragraph tableParagraph = document.Paragraphs.Add();

Word.Range tableRange = tableParagraph.Range;

Word.Table paymentsTable = document.Tables.Add(tableRange, allCategories.Count() + 1, 2);

paymentsTable.Borders.InsideLineStyle = paymentsTable.Borders.OutsideLineStyle =

Word.WdLineStyle.wdLineStyleSingle;

paymentsTable.Range.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

Word.Range cellRange;

cellRange = paymentsTable.Cell(1, 1).Range;

cellRange.Text = "Категория";

cellRange = paymentsTable.Cell(1, 2).Range;

cellRange.Text = "Сумма расходов";

paymentsTable.Rows[1].Range.Font.Name = "Times New Roman";

paymentsTable.Rows[1].Range.Font.Size = 14;

paymentsTable.Rows[1].Range.Bold = 1;

paymentsTable.Rows[1].Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

for (int i = 0; i < allCategories.Count(); i++)

{

var currentCategory = allCategories[i];

cellRange = paymentsTable.Cell(i + 2, 1).Range;

cellRange.Text = currentCategory.Name;

cellRange.Font.Name = "Times New Roman";

cellRange.Font.Size = 12;

cellRange = paymentsTable.Cell(i + 2, 2).Range;

cellRange.Text = user.Payment.ToList().

Where(u => u.Category == currentCategory).Sum(u => u.Num \* u.Price).ToString() + " руб.";

cellRange.Font.Name = "Times New Roman";

cellRange.Font.Size = 12;

} //завершение цикла по строкам таблицы

document.Paragraphs.Add(); //пустая строка

Payment maxPayment = user.Payment.OrderByDescending(u => u.Price \* u.Num).FirstOrDefault();

if (maxPayment != null)

{

Word.Paragraph maxPaymentParagraph = document.Paragraphs.Add();

Word.Range maxPaymentRange = maxPaymentParagraph.Range;

maxPaymentRange.Text = $"Самый дорогостоящий платеж - {maxPayment.Name} за {(maxPayment.Price \* maxPayment.Num).ToString()} " + $"руб. от {maxPayment.Date.ToString()}";

maxPaymentParagraph.set\_Style("Подзаголовок");

maxPaymentRange.Font.Color = Word.WdColor.wdColorDarkRed;

maxPaymentRange.InsertParagraphAfter();

}

document.Paragraphs.Add(); //пустая строка

Payment minPayment = user.Payment.OrderBy(u => u.Price \* u.Num).FirstOrDefault();

if (maxPayment != null)

{

Word.Paragraph minPaymentParagraph = document.Paragraphs.Add();

Word.Range minPaymentRange = minPaymentParagraph.Range;

minPaymentRange.Text = $"Самый дешевый платеж - {minPayment.Name} за {(minPayment.Price \* minPayment.Num).ToString()} " + $"руб. от {minPayment.Date.ToString()}";

minPaymentParagraph.set\_Style("Подзаголовок");

minPaymentRange.Font.Color = Word.WdColor.wdColorDarkGreen;

minPaymentRange.InsertParagraphAfter();

}

if (user != allUsers.LastOrDefault()) document.Words.Last.InsertBreak(Word.WdBreakType.wdPageBreak);

application.Visible = true;

document.SaveAs2(@"C:\Users\197719\source\repos\WpfApp1\Новая"); document.SaveAs2(@"C:\Users\197719\source\repos\WpfApp1\Новая", Word.WdExportFormat.wdExportFormatPDF);

foreach (Microsoft.Office.Interop.Word.Section section in document.Sections)

{

Microsoft.Office.Interop.Word.Range headerRange =

section.Headers[Microsoft.Office.Interop.Word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;

headerRange.Fields.Add(headerRange, Microsoft.Office.Interop.Word.WdFieldType.wdFieldPage);

headerRange.ParagraphFormat.Alignment =

Microsoft.Office.Interop.Word.WdParagraphAlignment.wdAlignParagraphCenter;

headerRange.Font.ColorIndex = Microsoft.Office.Interop.Word.WdColorIndex.wdBlue;

headerRange.Font.Size = 10;

headerRange.Text = DateTime.Now.ToString();

}

foreach (Microsoft.Office.Interop.Word.Section wordSection in document.Sections)

{

Microsoft.Office.Interop.Word.Range footerRange =

wordSection.Footers[Microsoft.Office.Interop.Word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;

footerRange.Font.ColorIndex = Microsoft.Office.Interop.Word.WdColorIndex.wdDarkRed;

footerRange.Font.Size = 10;

footerRange.ParagraphFormat.Alignment =

Microsoft.Office.Interop.Word.WdParagraphAlignment.wdAlignParagraphCenter;

footerRange.Text = Convert.ToString(user.ID);

}

} //завершение цикла по пользователям

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

var allUsers = \_context.User.ToList().OrderBy(u => u.Фамилия).ToList();

var application = new Excel.Application();

application.SheetsInNewWorkbook = allUsers.Count();

Excel.Workbook workbook = application.Workbooks.Add(Type.Missing);

for (int i = 0; i < allUsers.Count(); i++)

{

int startRowIndex = 1;

Excel.Worksheet worksheet = application.Worksheets.Item[i + 1];

worksheet.Name = allUsers[i].Фамилия;

worksheet.Cells[1][startRowIndex] = "Дата платежа";

worksheet.Cells[2][startRowIndex] = "Название";

worksheet.Cells[3][startRowIndex] = "Стоимость";

worksheet.Cells[4][startRowIndex] = "Количество";

worksheet.Cells[5][startRowIndex] = "Сумма";

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

columlHeaderRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

columlHeaderRange.Font.Bold = true;

startRowIndex++;

var userCategories = allUsers[i].Payment.OrderBy(u => u.Date).GroupBy(u => u.Category).OrderBy(u => u.Key.Name);

foreach (var groupCategory in userCategories)

{

Excel.Range headerRange = worksheet.Range[worksheet.Cells[1][startRowIndex], worksheet.Cells[5][startRowIndex]];

headerRange.Merge();

headerRange.Value = groupCategory.Key.Name;

headerRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

headerRange.Font.Italic = true;

startRowIndex++;

foreach (var payment in groupCategory)

{

worksheet.Cells[1][startRowIndex] = payment.Date.ToString();

worksheet.Cells[2][startRowIndex] = payment.Name;

worksheet.Cells[3][startRowIndex] = payment.Price;

(worksheet.Cells[3][startRowIndex] as Excel.Range).NumberFormat = "0.00";

worksheet.Cells[4][startRowIndex] = payment.Num;

worksheet.Cells[5][startRowIndex].Formula = $"=C{startRowIndex}\*D{startRowIndex}";

(worksheet.Cells[5][startRowIndex] as Excel.Range).NumberFormat = "0.00";

startRowIndex++;

Excel.Range sumRange = worksheet.Range[worksheet.Cells[1][startRowIndex], worksheet.Cells[4][startRowIndex]];

sumRange.Merge();

sumRange.Value = "ИТОГО:";

sumRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

worksheet.Cells[5][startRowIndex].Formula = $"=SUM(E{startRowIndex - groupCategory.Count()}:" +

$"E{startRowIndex - 1})";

sumRange.Font.Bold = worksheet.Cells[5][startRowIndex].Font.Bold = true;

startRowIndex++;

Excel.Range rangeBorders = worksheet.Range[worksheet.Cells[1][1], worksheet.Cells[5][startRowIndex - 1]]; rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle = rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle = rangeBorders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle = rangeBorders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle = Excel.XlLineStyle.xlContinuous;

worksheet.Columns.AutoFit();

} //завершение цикла по платежам

application.Visible = true;

worksheet.Cells[6][startRowIndex] = "a";

}

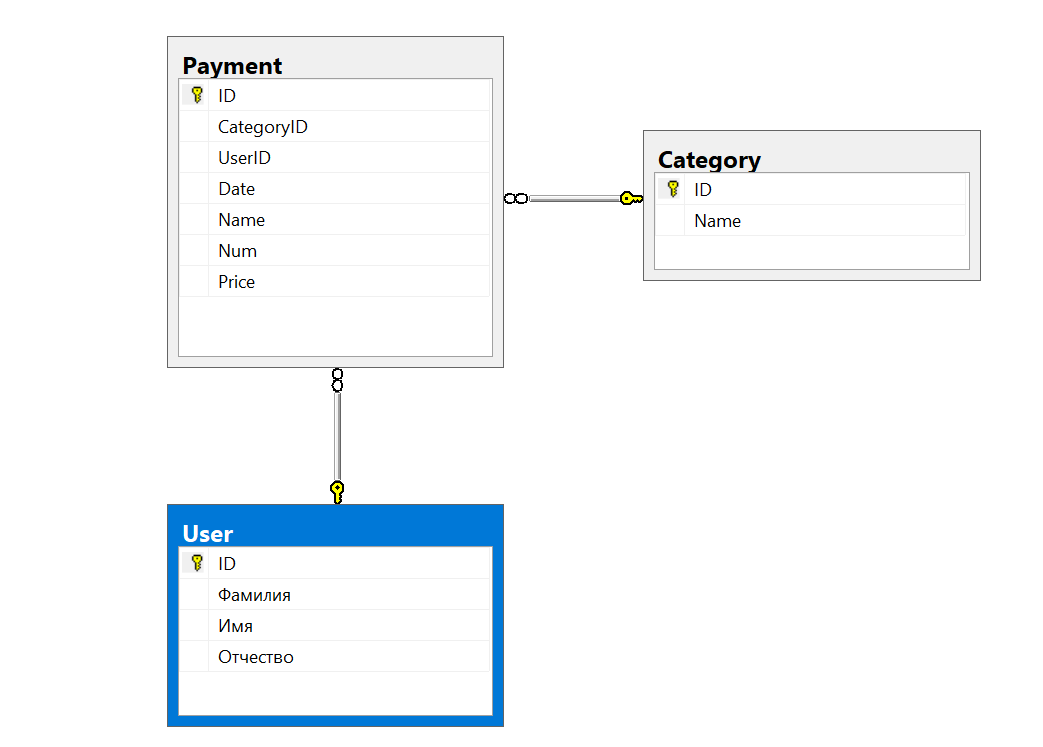
}

}

}

}

Рисунок 8. Диаграмма база данных



Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

# ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №19-20

**Тема: Разработка Desktop-приложения**

Студент: Кузнецов Д.Д.

Группа: ИСИП-619

Преподаватель: Е.Л. Альшакова

Дата: 03.10.2022

**Цель работы:** приобрести практические навыки по разработке Desktop-приложения

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Вывод всех агентов

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Поиск агентов по имени

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Поиск агентов по номеру телефона

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Сортировка агентов по типу организации

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Сортировка агентов по названию

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6. Сортировка агентов по продажам за год

Изображение выглядит как текст

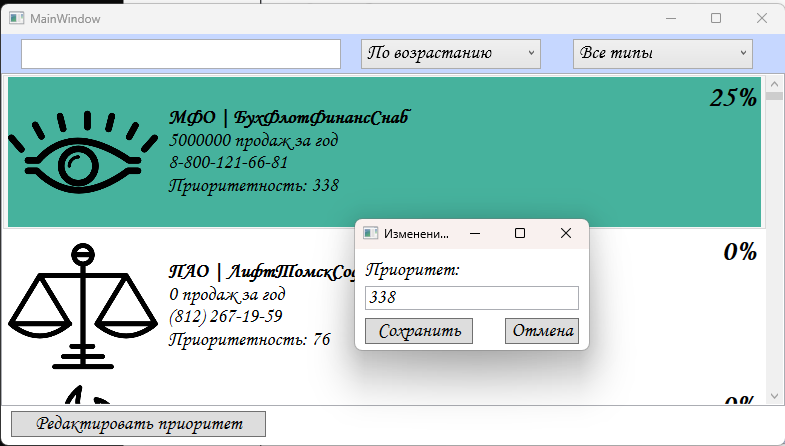
Автоматически созданное описание

Рисунок 7. Сортировка агентов по приоритетности

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Сортировка агентов по скидке

  
Рисунок 9. Редактирование приоритета

<Window x:Class="Gorlanov\_619.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:Gorlanov\_619"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40\*"/>

<RowDefinition Height="340\*"/>

<RowDefinition Height="40\*"/>

</Grid.RowDefinitions>

<DockPanel Grid.Row="0" HorizontalAlignment="Stretch" Background="#C6D7FF">

<TextBox x:Name="TextBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Margin="20, 0, 0, 0" TextWrapping="Wrap" FontSize="20" Height="30" Width="320" VerticalContentAlignment="Center" TextChanged="TextBox\_TextChanged"/>

<ComboBox x:Name="ComboBoxTypeSort" HorizontalAlignment="Stretch" VerticalAlignment="Center" Margin="20, 0, 0, 0" Text="Сортировка" FontSize="20" Height="30" Width="180" VerticalContentAlignment="Center" SelectedIndex="0" SelectionChanged="ComboBox\_SelectionChanged">

<TextBlock Text="По возрастанию"/>

<TextBlock Text="По убыванию"/>

</ComboBox>

<ComboBox x:Name="ComboBoxTypeFilter" HorizontalAlignment="Stretch" VerticalAlignment="Center" Margin="20, 0, 20, 0" Text="Фильтрация" FontSize="20" Height="30" Width="180" VerticalContentAlignment="Center" SelectedIndex="0" SelectionChanged="ComboBox\_SelectionChanged">

<TextBlock Text="Все типы"/>

<TextBlock Text="Тип организации"/>

<TextBlock Text="Название агента"/>

<TextBlock Text="Продажи за год"/>

<TextBlock Text="Приоритетность"/>

<TextBlock Text="Скидка"/>

</ComboBox>

</DockPanel>

<ListBox x:Name="ListAgent" Grid.Row="1" SelectionChanged="ListAgent\_SelectionChanged" >

<ListBox.ItemTemplate>

<DataTemplate>

<DockPanel x:Name="Dock">

<Image Height="150" Width="150">

<Image.Source>

<Binding Path="PhotoLogo">

<Binding.TargetNullValue>

<ImageSource>/agents/default.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<StackPanel Margin="10, 0, 0, 0" VerticalAlignment="Center">

<TextBlock FontSize="20" FontWeight="Bold">

<TextBlock.Text>

<MultiBinding StringFormat="{}{0} | {1}">

<Binding Path="AgentType.Name"/>

<Binding Path="Name"/>

</MultiBinding>

</TextBlock.Text>

</TextBlock>

<TextBlock FontSize="20" >

<TextBlock.Text>

<Binding StringFormat="{}{0} продаж за год" Path="Quantity">

<Binding.TargetNullValue>

0 продаж за год

</Binding.TargetNullValue>

</Binding>

</TextBlock.Text>

</TextBlock>

<TextBlock FontSize="20" Text="{Binding Path=Phone}" />

<TextBlock FontSize="20" Text="{Binding StringFormat=Приоритетность: {0}, Path=Priority}" />

</StackPanel>

<TextBlock FontSize="30" FontWeight="Bold" Text="{Binding StringFormat={}{0}%, Path=Discount}" HorizontalAlignment="Right" Margin="5"/>

<DockPanel.Style>

<Style TargetType="DockPanel">

<Style.Triggers>

<DataTrigger Binding="{Binding Discount}" Value="25">

<Setter Property="Background" Value="#46B29D"/>

</DataTrigger>

</Style.Triggers>

</Style>

</DockPanel.Style>

</DockPanel>

</DataTemplate>

</ListBox.ItemTemplate>

<ListBox.ItemContainerStyle>

<Style TargetType="ListBoxItem">

<Setter Property="HorizontalContentAlignment" Value="Stretch"/>

</Style>

</ListBox.ItemContainerStyle>

</ListBox>

<Button x:Name="EditPriorityButton" Content="Редактировать приоритет" HorizontalAlignment="Left" Margin="10,5,0,0" Grid.Row="2" VerticalAlignment="Top" Width="255" FontSize="20" Visibility="Hidden" Click="EditPriorityButton\_Click"/>

</Grid>

</Window>

Листинг 1. Файл MainWindow.xaml

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace Gorlanov\_619

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private List<Agent> agents = new List<Agent>();

public MainWindow()

{

InitializeComponent();

UpdateAgents();

}

private void UpdateAgents()

{

agents = Entities.GetContext().Agent.ToList();

var result = from agent in Entities.GetContext().Agent

join productSale in Entities.GetContext().ProductSale on agent.ID equals productSale.AgentID

join product in Entities.GetContext().Product on productSale.ProductID equals product.ID

group agent by new { agent.ID, AgentTypeName = agent.AgentType.Name, agent.Name, agent.Email, agent.Phone, agent.PhotoLogo, agent.Address, agent.Priority, agent.Director, agent.INN, agent.KPP, productSale.Quantity, productSale.DateRealization } into g

select new

{

ID = g.Key.ID,

AgentTypeName = g.Key.AgentTypeName,

Name = g.Key.Name,

Email = g.Key.Email,

Phone = g.Key.Phone,

PhotoLogo = g.Key.PhotoLogo,

Address = g.Key.Address,

Priority = g.Key.Priority,

Director = g.Key.Director,

INN = g.Key.INN,

KPP = g.Key.KPP,

Quantity = g.Key.Quantity,

DataRealization = g.Key.DateRealization

};

var r = result.ToList();

foreach (var item in r)

{

foreach (var item2 in agents)

{

item2.Quantity = r.Where(a => a.ID == item2.ID && a.DataRealization.Year == 2019).Sum(q => q.Quantity);

item2.QuantityAll = r.Where(a => a.ID == item2.ID).Sum(q => q.Quantity);

}

}

foreach (var agent in agents)

{

if (agent.QuantityAll >= 0 && agent.QuantityAll < 10000)

{

agent.Discount = 0;

}

else if (agent.QuantityAll >= 10000 && agent.QuantityAll < 50000)

{

agent.Discount = 5;

}

else if (agent.QuantityAll >= 50000 && agent.QuantityAll < 150000)

{

agent.Discount = 10;

}

else if (agent.QuantityAll >= 150000 && agent.QuantityAll < 500000)

{

agent.Discount = 20;

}

else if (agent.QuantityAll >= 500000)

{

agent.Discount = 25;

}

}

ListAgent.ItemsSource = agents;

}

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

{

FilterUpdateAgents();

}

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

{

FilterUpdateAgents();

}

private void FilterUpdateAgents()

{

//загружаем всех пользователей в список

var currentAgents = agents;

//осуществляем поиск по Ф.И.О. без учета регистра букв

currentAgents = currentAgents.Where(x => x.Name.ToLower().Contains(TextBox.Text.ToLower()) || x.Email.ToLower().Contains(TextBox.Text.ToLower()) || x.Phone.ToLower().Contains(TextBox.Text.ToLower())).ToList();

if (ComboBoxTypeSort != null && ComboBoxTypeFilter != null)

{

if (ComboBoxTypeSort.SelectedIndex == 1)

{

if (ComboBoxTypeFilter.SelectedIndex == 1)

{

currentAgents = currentAgents.OrderBy(x => x.AgentType.Name).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 2)

{

currentAgents = currentAgents.OrderBy(x => x.Name).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 3)

{

currentAgents = currentAgents.OrderBy(x => x.Quantity).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 4)

{

currentAgents = currentAgents.OrderBy(x => x.Priority).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 5)

{

currentAgents = currentAgents.OrderBy(x => x.Discount).ToList();

}

}

else

{

if (ComboBoxTypeFilter.SelectedIndex == 1)

{

currentAgents = currentAgents.OrderByDescending(x => x.AgentType.Name).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 2)

{

currentAgents = currentAgents.OrderByDescending(x => x.Name).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 3)

{

currentAgents = currentAgents.OrderByDescending(x => x.Quantity).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 4)

{

currentAgents = currentAgents.OrderByDescending(x => x.Priority).ToList();

}

else if (ComboBoxTypeFilter.SelectedIndex == 5)

{

currentAgents = currentAgents.OrderByDescending(x => x.Discount).ToList();

}

}

}

if (ListAgent != null)

{

ListAgent.ItemsSource = currentAgents;

}

//осуществляем сортировку в зависимости от выбора пользователя

//if (ComboBoxTypeSort.SelectedIndex == 0)

// ListAgent.ItemsSource = currentAgents.OrderBy(x => x.AgentType.Name).ToList();

//else ListAgent.ItemsSource = currentAgents.OrderByDescending(x => x.AgentType.Name).ToList();

}

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

{

if (ListAgent.SelectedItem != null)

EditPriorityButton.Visibility = Visibility.Visible;

else

EditPriorityButton.Visibility = Visibility.Hidden;

}

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

{

EditPriorityWindow editPriorityWindow = new EditPriorityWindow((Agent)ListAgent.SelectedItem);

editPriorityWindow.Owner = this;

editPriorityWindow.Closed += EditPriorityWindow\_Closed;

editPriorityWindow.ShowDialog();

}

private void EditPriorityWindow\_Closed(object sender, EventArgs e)

{

if ((sender as EditPriorityWindow).DialogResult == true)

{

var index = agents.IndexOf((Agent)ListAgent.SelectedItem);

agents[index].Priority = (sender as EditPriorityWindow).priority;

Entities.GetContext().Agent.ToList()[index].Priority = (sender as EditPriorityWindow).priority;

Entities.GetContext().SaveChanges();

ListAgent.ItemsSource = null;

ListAgent.ItemsSource = agents;

}

}

}

}

Листинг 2. Файл MainWindow.xaml.cs

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

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

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="Monotype Corsiva"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="FontFamily" Value="Monotype Corsiva"/>

</Style>

<Style TargetType="ComboBox">

<Setter Property="FontFamily" Value="Monotype Corsiva"/>

</Style>

</ResourceDictionary>

Листинг 3. Файл Dictionary.xaml

<Application x:Class="Gorlanov\_619.App"

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

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

xmlns:local="clr-namespace:Gorlanov\_619"

StartupUri="MainWindow.xaml">

<Application.Resources>

<ResourceDictionary Source="Dictionary.xaml"/>

</Application.Resources>

</Application>

Листинг 4. Файл App.xaml

<Window x:Class="Gorlanov\_619.EditPriorityWindow"

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:Gorlanov\_619"

mc:Ignorable="d"

Title="Изменение приоритета" Height="140" Width="250" WindowStartupLocation="CenterOwner">

<Grid>

<Button x:Name="SaveButton" Content="Сохранить" HorizontalAlignment="Left" Margin="10,0,0,6" VerticalAlignment="Bottom" Width="108" Click="SaveButton\_Click" FontSize="20"/>

<Button x:Name="CancelButton" Content="Отмена" HorizontalAlignment="Right" Margin="0,0,10,6" VerticalAlignment="Bottom" Width="74" Click="CancelButton\_Click" FontSize="20"/>

<TextBox x:Name="PriorityTextBox" Height="24" Margin="10,37,10,0" TextWrapping="Wrap" Text="TextBox" VerticalAlignment="Top" FontSize="20"/>

<TextBlock Margin="10,10,10,0" TextWrapping="Wrap" Text="Приоритет:" VerticalAlignment="Top" FontSize="20"/>

</Grid>

</Window>

Листинг 5. Файл EditPriorityWindow.xaml

using System;

using System.Windows;

namespace Gorlanov\_619

{

/// <summary>

/// Логика взаимодействия для EditPriorityWindow.xaml

/// </summary>

public partial class EditPriorityWindow : Window

{

internal int priority;

public EditPriorityWindow(Agent agent)

{

InitializeComponent();

PriorityTextBox.Text = agent.Priority.ToString();

}

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

{

priority = Convert.ToInt32(PriorityTextBox.Text);

DialogResult = true;

}

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

{

DialogResult = false;

}

}

}

Литсинг 6. Файл EditPriorityWindow.xaml.cs

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №21-24**

**Тема:** **РАЗРАБОТКА ПРОИЛОЖЕНИЯ ДЛЯ ОТСЛЕЖИВАНИЯ ВАЛЮТ**

Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: П.А. Костиков

Дата: 05.09.2022

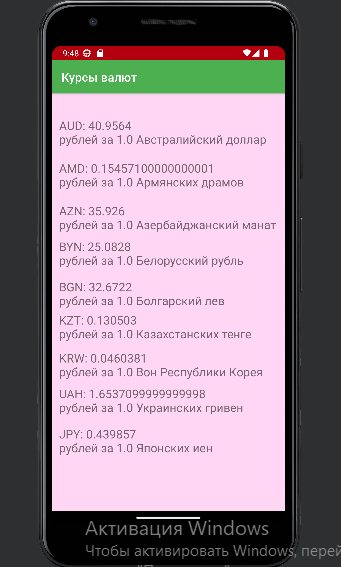


Рис.1 Работающие приложение

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:background="@color/material\_dynamic\_tertiary90"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 tools:layout\_editor\_absoluteX="94dp"  
 tools:layout\_editor\_absoluteY="38dp">  
  
 <TextView  
 android:id="@+id/KRWTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="432dp"  
 android:text="KRW: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/JPYTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="560dp"  
 android:text="JPY: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/tempTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="40dp"  
 android:text="AUD: "  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/AZNTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="184dp"  
 android:text="AZN: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/AMDTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="112dp"  
 android:text="AMD: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/UAHTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="492dp"  
 android:text="UAH: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/KZTTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="368dp"  
 android:text="KZT: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/BGNTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="312dp"  
 android:text="BGN: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/BYNTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="244dp"  
 android:text="BYN: "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Файл 1. Activity main

package com.Kuz  
  
import android.annotation.SuppressLint  
import android.os.Bundle  
import android.util.Log  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
import com.android.volley.Request  
import com.android.volley.RequestQueue  
import com.android.volley.toolbox.JsonObjectRequest  
import com.android.volley.toolbox.Volley  
import org.json.JSONException  
  
  
  
class MainActivity : AppCompatActivity() {  
 lateinit var mRequestQueue: RequestQueue // очередь запросов  
  
 lateinit var tempTextView: TextView  
 lateinit var AZNTextView: TextView  
 lateinit var BGNTextView: TextView// текстовые поля для отображения данных  
 lateinit var BYNTextView: TextView  
 lateinit var AMDTextView: TextView  
 lateinit var KZTTextView: TextView  
 lateinit var UAHTextView: TextView  
 lateinit var KRWTextView: TextView  
 lateinit var JPYTextView: TextView  
 var ValueAUD = 0.0  
 var NameAUD = toString()  
 var NominalAUD = 0.0  
 var ValueBGN = 0.0  
 var NameBGN = toString()  
 var NominalBGN = 0.0  
 var ValueAZN = 0.0  
 var NameAZN = toString()  
 var NominalAZN = 0.0  
 var ValueBYN = 0.0  
 var NameBYN = toString()  
 var NominalBYN = 0.0  
 var ValueAMD = 0.0  
 var NameAMD = toString()  
 var NominalAMD = 0.0  
 var ValueKZT = 0.0  
 var NameKZT = toString()  
 var NominalKZT = 0.0  
 var ValueUAH = 0.0  
 var NameUAH= toString()  
 var NominalUAH = 0.0  
 var ValueKRW = 0.0  
 var NameKRW= toString()  
 var NominalKRW = 0.0  
 var ValueJPY = 0.0  
 var NameJPY= toString()  
 var NominalJPY = 0.0  
  
 @SuppressLint("MissingInflatedId")  
 override fun onCreate(savedInstanceState: Bundle?) {  
  
 super.onCreate(savedInstanceState)  
  
 val Url =  
 "https://www.cbr-xml-daily.ru/daily\_json.js" //url, из которого мы будем брать JSON-объект  
 setContentView(R.layout.*activity\_main*);  
 tempTextView = findViewById(R.id.*tempTextView*);  
 AZNTextView = findViewById(R.id.*AZNTextView*);  
 BGNTextView = findViewById(R.id.*BGNTextView*);  
 BYNTextView = findViewById(R.id.*BYNTextView*);  
 AMDTextView = findViewById(R.id.*AMDTextView*);  
 KZTTextView = findViewById(R.id.*KZTTextView*);  
 UAHTextView = findViewById(R.id.*UAHTextView*);  
 KRWTextView = findViewById(R.id.*KRWTextView*);  
 JPYTextView = findViewById(R.id.*JPYTextView*);  
  
 mRequestQueue = Volley.newRequestQueue(this);  
 getValuet(Url);  
 setValues()  
  
  
  
  
  
  
 }  
  
 private fun getValuet(url: String) {  
  
 val request = JsonObjectRequest(  
 Request.Method.*GET*, //получение данных  
 url, null, **{** response **->** try {  
 val Valute = response.getJSONObject("Valute")  
 val AZN = Valute.getJSONObject("AZN")  
 val AUD = Valute.getJSONObject("AUD")  
 val BGN = Valute.getJSONObject("BGN")  
 val BYN = Valute.getJSONObject("BYN")  
 val AMD = Valute.getJSONObject("AMD")  
 val KZT = Valute.getJSONObject("KZT")  
 val UAH = Valute.getJSONObject("UAH")  
 val KRW = Valute.getJSONObject("KRW")  
 val JPY = Valute.getJSONObject("JPY")  
 ValueBYN = BYN.getDouble("Value")  
 NameBYN =BYN.getString("Name")  
 ValueAZN = AZN.getDouble("Value")  
 NameAZN =AZN.getString("Name")  
 ValueBGN = BGN.getDouble("Value")  
 NameBGN = BGN.getString("Name")  
 ValueAUD = AUD.getDouble("Value")  
 NameAUD = AUD.getString("Name")  
 ValueAMD = AMD.getDouble("Value")  
 NameAMD = AMD.getString("Name")  
 NominalBGN = BGN.getInt("Nominal").toDouble()  
 NominalAUD = AUD.getInt("Nominal").toDouble()  
 NominalAZN = AZN.getInt("Nominal").toDouble()  
 NominalBYN = BYN.getInt("Nominal").toDouble()  
 NominalAMD = AMD.getInt("Nominal").toDouble()  
 ValueKZT = KZT.getDouble("Value")  
 NameKZT = KZT.getString("Name")  
 NominalKZT = KZT.getInt("Nominal").toDouble()  
 ValueUAH = UAH.getDouble("Value")  
 NameUAH = UAH.getString("Name")  
 NominalUAH = UAH.getInt("Nominal").toDouble()  
 ValueKRW = KRW.getDouble("Value")  
 NameKRW = KRW.getString("Name")  
 NominalKRW = KRW.getInt("Nominal").toDouble()  
 ValueJPY = JPY.getDouble("Value")  
 NameJPY = JPY.getString("Name")  
 NominalJPY = JPY.getInt("Nominal").toDouble()  
 ValueAMD = ValueAMD / NominalAMD  
 ValueBYN = ValueBYN / NominalBYN  
 ValueAZN = ValueAZN / NominalAZN  
 ValueBGN = ValueBGN / NominalBGN  
 ValueAUD = ValueAUD / NominalAUD  
 ValueKZT = ValueKZT / NominalKZT  
 ValueUAH = ValueUAH / NominalUAH  
 ValueKRW = ValueKRW / NominalKRW  
 ValueJPY = ValueJPY / NominalJPY  
 NominalBGN= NominalBGN/ NominalBGN  
 NominalAUD= NominalAUD/NominalAUD  
 NominalAZN= NominalAZN/ NominalAZN  
 NominalBYN=NominalBYN/NominalBYN  
 NominalAMD=NominalAMD/NominalAMD  
 NominalKZT=NominalKZT/NominalKZT  
 NominalUAH=NominalUAH/NominalUAH  
 NominalKRW=NominalKRW/NominalKRW  
 NominalJPY=NominalJPY/NominalJPY  
 setValues() // создадим метод setValues для присваивания значений переменным  
 } catch (e: JSONException) {  
 Log.d("AppLog", "JSON ERROR")  
 e.printStackTrace()  
 }  
 **}**) **{** error **->**// в случае возникновеня ошибки  
 Log.d("AppLog", "API ERROR")  
 error.printStackTrace()  
 **}** mRequestQueue.add(request)  
  
 }  
  
 fun setValues(){  
 tempTextView.setText("AUD: " + ValueAUD +System.getProperty("line.separator")+ "рублей за " +NominalAUD + " " + NameAUD +System.getProperty("line.separator") + " ");  
 AZNTextView.setText("AZN: "+ ValueAZN + System.getProperty("line.separator")+ "рублей за " +NominalAZN+ " " +NameAZN);  
 BGNTextView.setText("BGN: "+ValueBGN + System.getProperty("line.separator")+ "рублей за " + NominalBGN+ " " +NameBGN);  
 BYNTextView.setText("BYN: "+ValueBYN + System.getProperty("line.separator")+ "рублей за " +NominalBYN+ " " +NameBYN)  
 AMDTextView.setText("AMD: "+ValueAMD +System.getProperty("line.separator")+ "рублей за " +NominalAMD+ " " +NameAMD)  
 KZTTextView.setText("KZT: "+ValueKZT +System.getProperty("line.separator")+ "рублей за " +NominalKZT+ " " +NameKZT)  
 UAHTextView.setText("UAH: "+ValueUAH +System.getProperty("line.separator")+ "рублей за " +NominalUAH+ " " +NameUAH)  
 KRWTextView.setText("KRW: "+ValueKRW +System.getProperty("line.separator")+ "рублей за " +NominalKRW+ " " +NameKRW)  
 JPYTextView.setText("JPY: "+ValueJPY +System.getProperty("line.separator")+ "рублей за " +NominalJPY+ " " +NameJPY)  
 }  
}

Файл 2. MainActivity

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №25-32**

**Тема:** **РАЗРАБОТКА СТРАНИЦЫ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ.**

**СОЗДАНИЕ СЛОВАРЯ СТИЛЕЙ**

Студент: Кузнецов Данила

Группа: ИСИП-619

Преподаватель: И.В. Сиберев

Дата: 05.09.2022

Задание 3

Вариант 4

Предметная область: Учебно-методическое

Управление (профессорско-преподавательский состав).

Основные предметно-значимые сущности: Сотрудники, Подразделения, Дисциплины.

Основные предметно-значимые атрибуты сущностей :

- сотрудники – фамилия, имя, отчество, пол, дата рождения, адрес прописки,

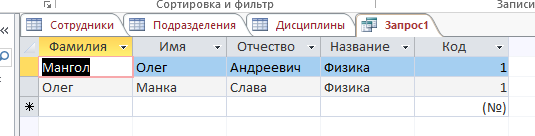
должность, подразделение;

- подразделения – название, вид подразделения;

- дисциплины – название.

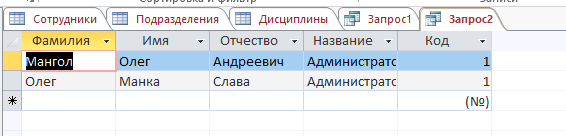
Основные требования к функциям системы:

- выбрать дисциплины, читаемые сотрудниками или определенным сотрудником;



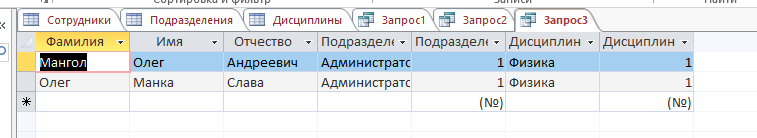
- выбрать список сотрудников по подразделениям или определенному

подразделению;



- выбрать дисциплины, читаемые сотрудниками по подразделениям или

определенному подразделению.



**Цель работы:** приобретение навыков анализа предметной области и построения концептуальной модели.

**Вариант №4. Реализация готовой продукции.**

**Описание предметной области.**

Вы работаете в компании, занимающейся оптово-розничной продажей различных товаров. Вашей задачей является отслеживание финансовой стороны ее работы. Деятельность компании организована следующим образом: компания торгует товарами из определенного спектра. Каждый из этих товаров характеризуется наименованием, оптовой ценой, розничной ценой и справочной информацией. В вашу компанию обращаются покупатели. Для каждого из них вы запоминаете в базе данных стандартные данные (наименование, адрес, телефон, контактное лицо) и составляете по каждой сделке документ, запоминая наряду с покупателем количество купленного им товара и дату покупки.

**Основные сущности.**

Таблица «Клиент»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код клиента | int |
| - | Фамилия | varchar(50) |
| - | Имя | varchar(50) |
| - | Номер телефона | varchar(50) |
| - | Адрес | varcahr(50) |

Таблица «Товар»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код товара | int |
| - | Название | varchar(50) |
| - | Оптовая цена | money |
| - | Розничная цена | money |
| - | Описание | varcahr(MAX) |

Таблица «Сделка»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код сделки | int |
| Внешний ключ | Код товара | int |
| Внешний ключ | Код клиента | int |
| - | Дата продажи | datetime |
| - | Количество | int |
| - | Оптовая продажа | bit |

****

Рисунок 1. ER-диаграмма.

**Перечень возможных запросов.**

Получить список сделок с указанием контактных данных о клиенте, информации о товаре, количестве купленного товара и сумме сделки.

**Цель работы:** ознакомиться со способами создания, модификации, удаления баз.

**Ход работы:**

1. Создание базы данных в MS SQL

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы Customers

Изображение выглядит как стол

Автоматически созданное описание

1. Создание таблицы Orders через запрос

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

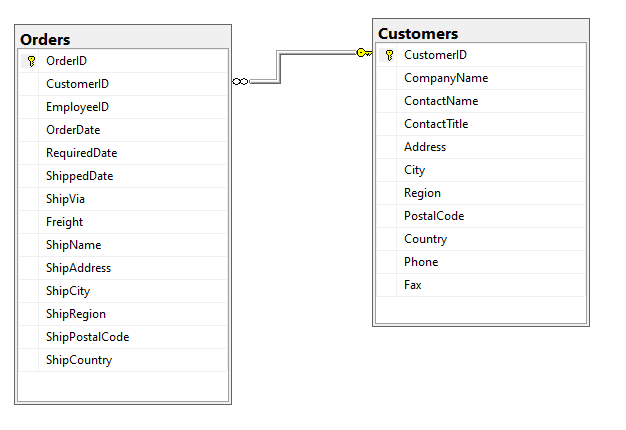
1. Создание диаграммы базы данных

Изображение выглядит как стол

Автоматически созданное описание

1. Создание связей между таблицами

Изображение выглядит как текст, снимок экрана, внутренний

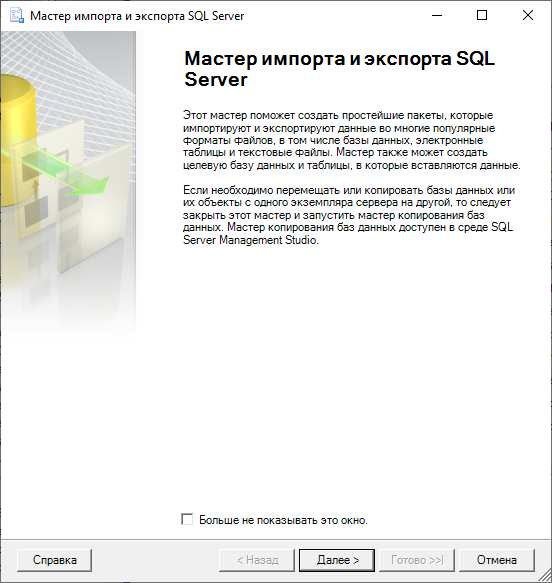
Автоматически созданное описание

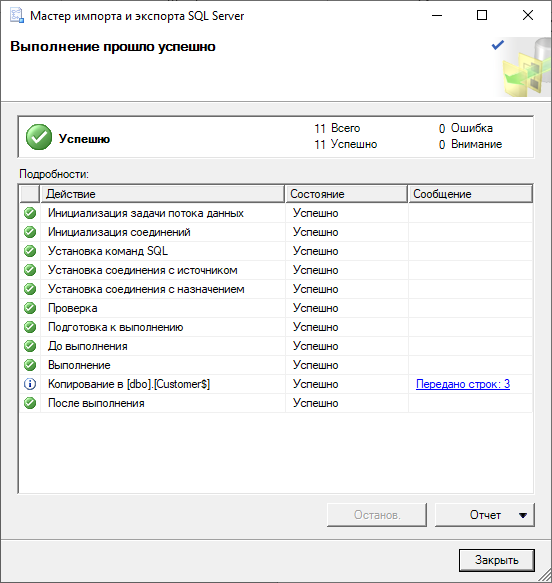
**Вывод:** в результате выполнения данной лабораторной работы мы ознакомились с двумя способами создания базы данных, таблиц и диаграмм, а именно – с помощью графического интерфейса, и с помощью T-SQL.

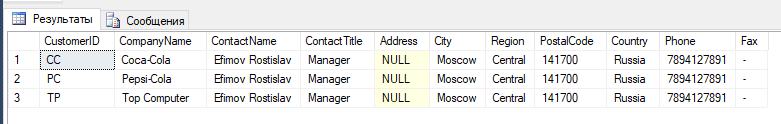
**Цель работы:** освоить методы заполнения таблиц в SQL Server.

**Ход работы:**

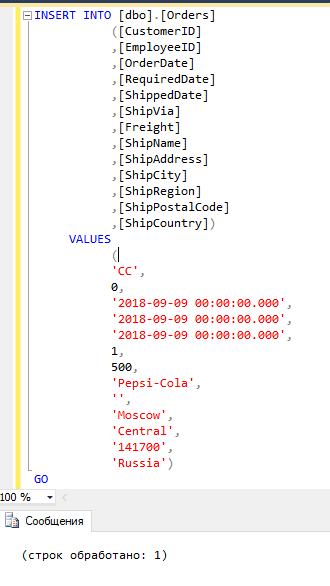
1. Импорт с помощью Management Tool

****

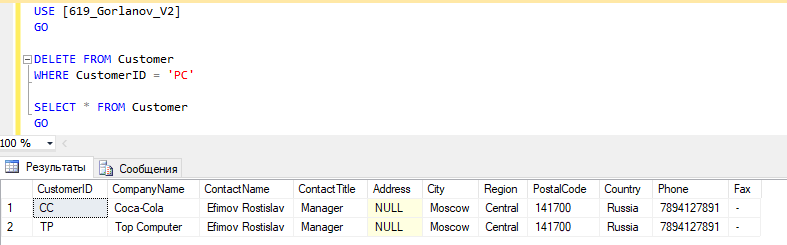
****

****

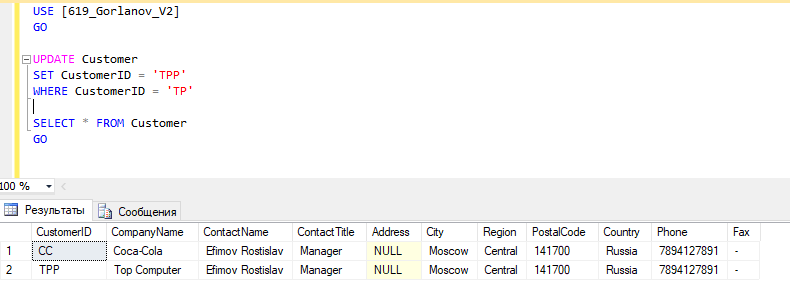
1. Импорт с помощью запроса

****

1. Удаление через запрос

****

1. Изменение строк

****

**Вывод:** в результате выполнения данной лабораторной работы мы ознакомились со способами импорта данных в таблицы, а также со способами обновления и удаления строк.

**Цель работы:** изучить создание запросов и представлений, использование механизма объединений.

**Ход работы:**

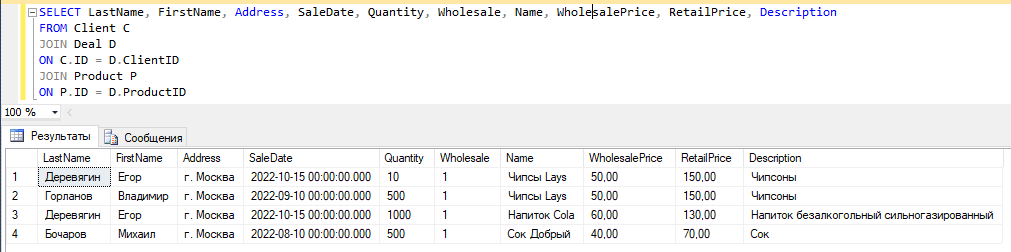
****

Рисунок 1. Вывод всей информации из БД в удобном виде

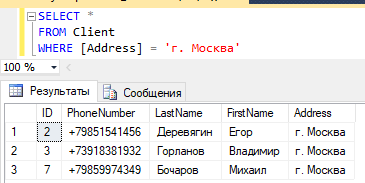


Рисунок 2. Запрос получения клиентов из определённого города

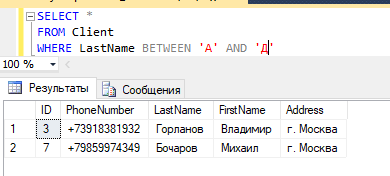


Рисунок 3. Запрос получения клиентов с именем в определённом буквенном диапазоне

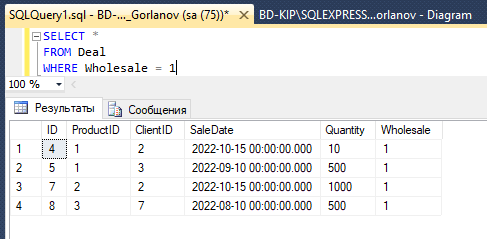
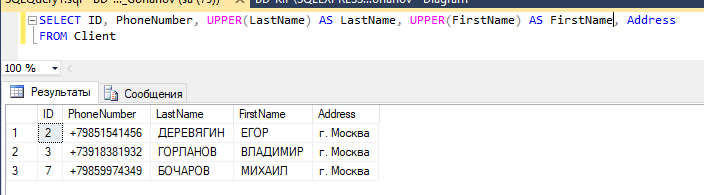


Рисунок 4. Запрос вывод только оптовых сделок

Рисунок 5. Вывод фамилии и имени в верхнем регистре

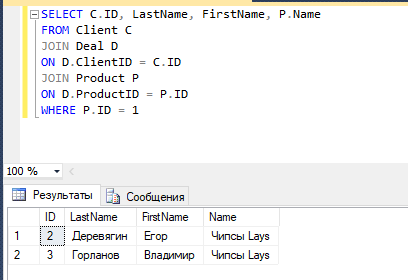
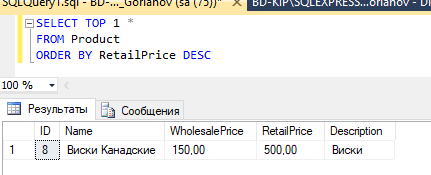
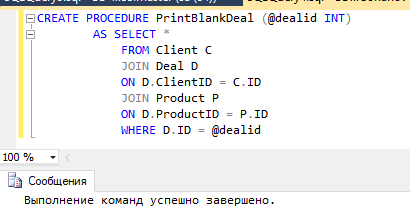


Рисунок 6. Вывод клиентов, купившие определённый товар

  
Рисунок 7. Вывод самого дорогого товара

**Цель работы:** изучение механизма динамических запросов с использованием хранимых процедур и функций

**Ход работы:**

****

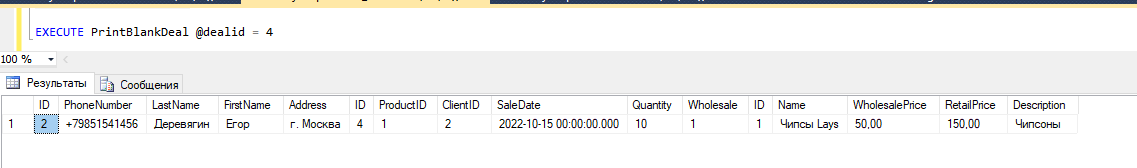
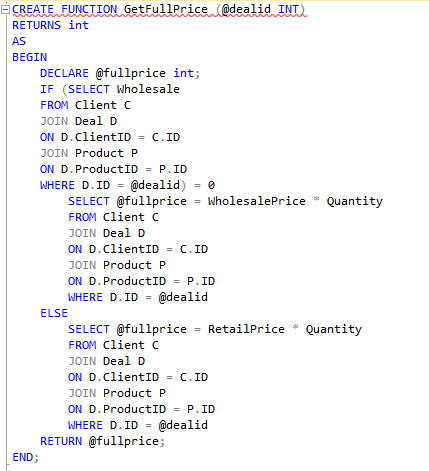
****

Рисунок 1. Печать бланка сделки



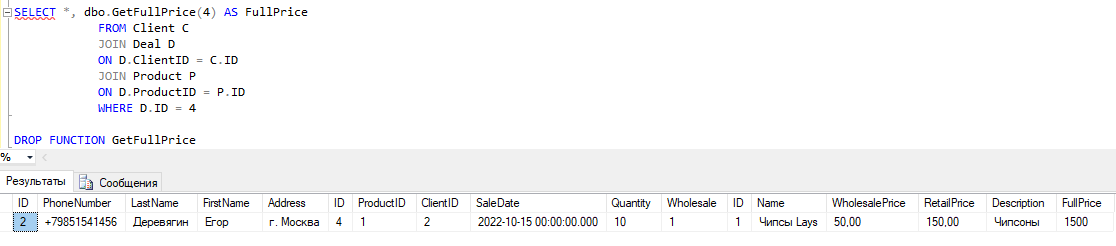


Рисунок 2. Функция расчёта полной стоимости сделки

**Цель работы:** изучение механизмов обеспечения целостности данных SQL

**Ход работы:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Добавление ограничения CHECK

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Триггер на добавление записи

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Демонстрация работы триггера

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Триггер при обновлении данных

Изображение выглядит как текст

Автоматически созданное описание

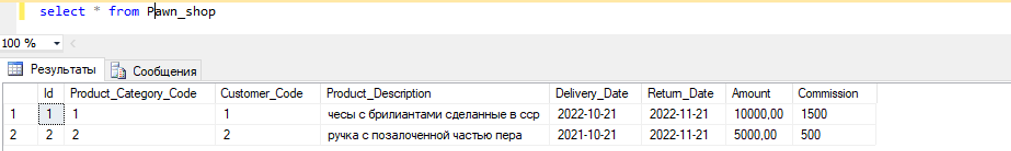
Рисунок 5. Демонстрация работы триггера

Изображение выглядит как текст

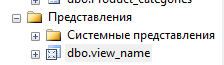
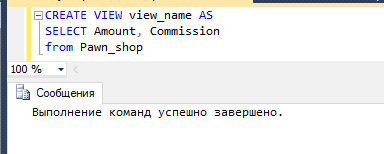
Автоматически созданное описание

Рисунок 6. Триггер удаления данных

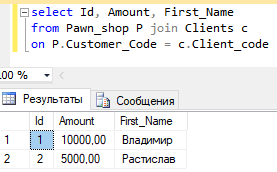
Задание

Способы создания запросов, запросов с подзапросами 

Использование представлений



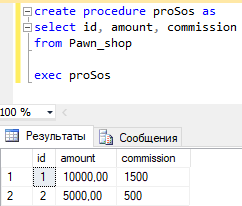
Использование механизма объединений



Цель:

Изучение механизма динамических запросов с использованием

хранимых процедур и функций.



Изображение выглядит как текст

Автоматически созданное описание

Вывод:

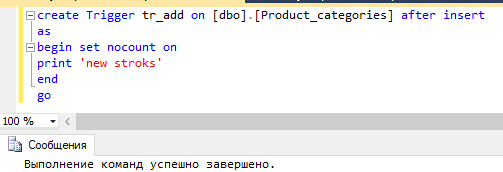
Изучил механизм динамических запросов с использованием

хранимых процедур и функций.

**Цель работы:**

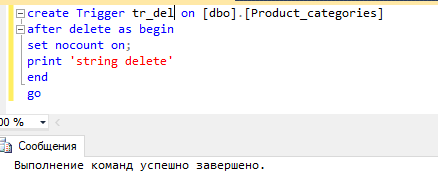
Изучение механизмов обеспечения целостности данных SQL

**Ход работы:** Создаем триггер



Изображение выглядит как текст, стол

Автоматически созданное описание



**Вывод:** Изучил механизмов обеспечения целостности данных SQL