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

В. В. Горланов

*(подпись) (И.О. Фамилия)*

Е.С. Деревягин

*(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**

**СОДЕРЖАНИЕ**

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №1-3 3](#_Toc120708914)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №4-6 12](#_Toc120708915)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №7-9 23](#_Toc120708916)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №10-12 34](#_Toc120708917)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №13-18 40](#_Toc120708918)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №19-20 53](#_Toc120708919)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №21-27 65](#_Toc120708920)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №28 77](#_Toc120708921)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №29 81](#_Toc120708922)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №30 85](#_Toc120708923)

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

# ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №1-3

**Тема: Разработка страницы авторизации пользователей. Создание словаря стилей**

Студент: В.В. Горланов,

Е.С. Деревягин

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.

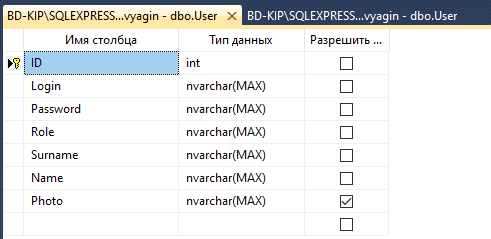


Рисунок 1. Скриншот проекта таблицы User

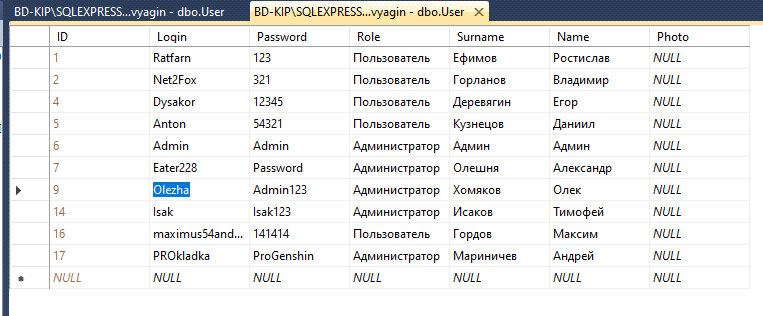


Рисунок 2. Скриншот содержимого таблицы User

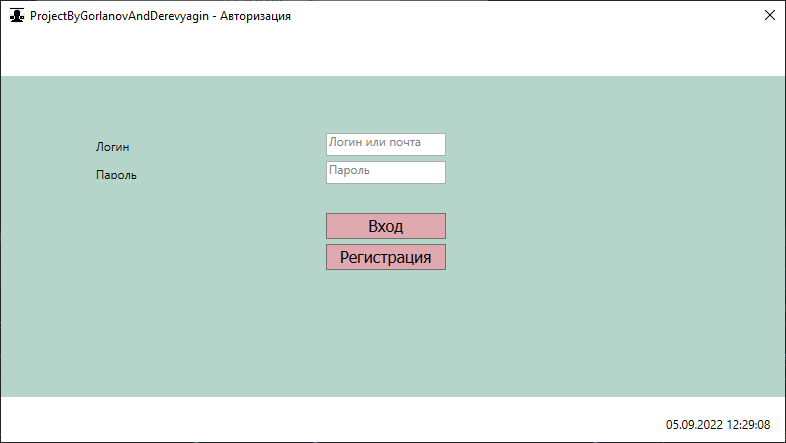


Рисунок 3. Демонстрация работы авторизации

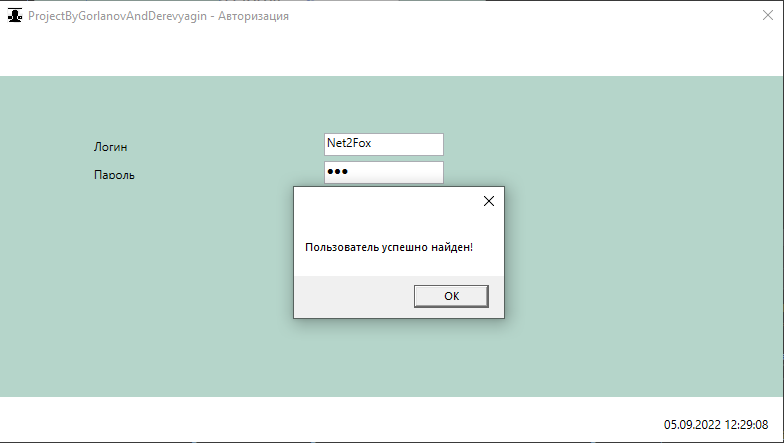


Рисунок 4. Демонстрация работы авторизации

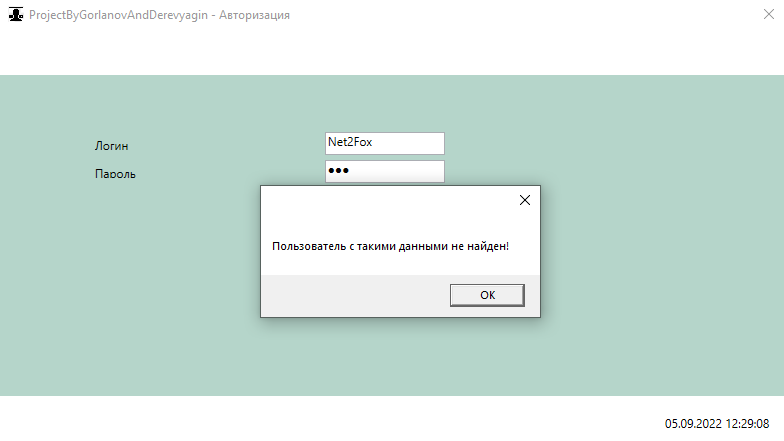


Рисунок 5. Демонстрация работы авторизации

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6. Демонстрация работы всплывающих подсказок

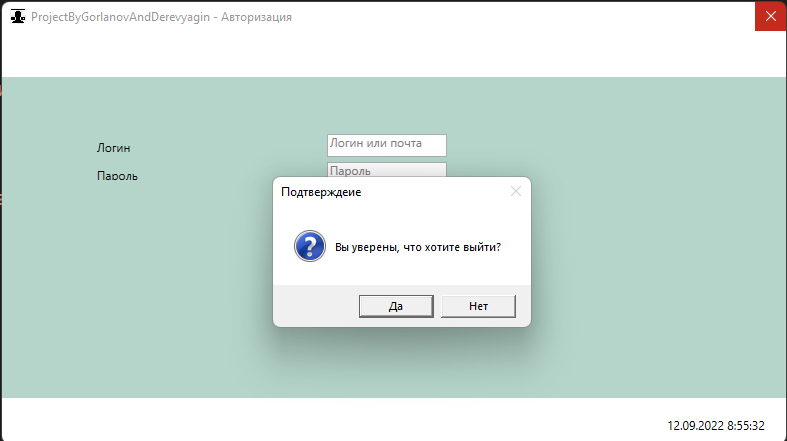


Рисунок 7. Подтверждение выхода из программы

<Window x:Class="\_619\_Gorlanov\_Derevyagin.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:\_619\_Gorlanov\_Derevyagin"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" Closing="Window\_Closing" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="45"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="45"/>

</Grid.RowDefinitions>

<Frame x:Name="MainFrame" Source="Pages/AuthPage.xaml" HorizontalAlignment="Left" Height="329" Grid.Row="1" VerticalAlignment="Top" Width="792" NavigationUIVisibility="Hidden" Navigated="MainFrame\_Navigated"/>

<TextBlock x:Name="TextBlock\_DateTime" Margin="0,0,10,10" Grid.Row="2" TextWrapping="Wrap" Text="DateTime" Height="16" VerticalAlignment="Bottom" HorizontalAlignment="Right" Width="109"/>

<Button x:Name="ButtonBack" Content="Назад" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top" Width="75" Height="25" Click="ButtonBack\_Click"/>

</Grid>

</Window>

Листинг 1. Файл MainWindow.xaml

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Forms;

using System.Windows.Navigation;

namespace \_619\_Gorlanov\_Derevyagin

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

TextBlock\_DateTime.Text = DateTime.Now.ToString();

}

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

{

DialogResult res = System.Windows.Forms.MessageBox.Show("Вы уверены, что хотите выйти?", "Подтверждеие", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (res != System.Windows.Forms.DialogResult.Yes)

{

e.Cancel = true;

}

}

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

{

if (MainFrame.CanGoBack)

{

MainFrame.GoBack();

}

}

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

{

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

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

if (page is Pages.AuthPage)

ButtonBack.Visibility = Visibility.Hidden;

else

ButtonBack.Visibility = Visibility.Visible;

}

}

}

Листинг 2. Файл MainWindow.xaml.cs

<Page x:Class="\_619\_Gorlanov\_Derevyagin.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\_Gorlanov\_Derevyagin.Pages"

mc:Ignorable="d"

d:DesignHeight="300" d:DesignWidth="300"

Title="Авторизация">

<Grid Background="#b5d5ca">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="34\*"/>

<ColumnDefinition Width="89\*"/>

<ColumnDefinition Width="177\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="52\*"/>

<RowDefinition Height="193\*"/>

<RowDefinition Height="55\*"/>

</Grid.RowDefinitions>

<TextBox x:Name="TextBoxLogin" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120" Grid.Row="1" Grid.Column="2" TextChanged="TextBoxLogin\_TextChanged">

<TextBox.ToolTip>

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

</TextBox.ToolTip>

</TextBox>

<PasswordBox x:Name="PasswordBox" HorizontalAlignment="Left" Margin="0,28,0,0" VerticalAlignment="Top" Width="120" Height="23" Grid.Row="1" Grid.Column="2" PasswordChanged="PasswordBox\_PasswordChanged">

<PasswordBox.ToolTip>

<ToolTip Content="Введите пароль"/>

</PasswordBox.ToolTip>

</PasswordBox>

<Label Content="Пароль" HorizontalAlignment="Left" Margin="0,28,0,0" VerticalAlignment="Top" Height="23" Width="89" Grid.Row="1" Grid.Column="1"/>

<Label Content="Логин" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="89" Grid.Row="1" Grid.Column="1"/>

<TextBlock Text="Логин или почта" Foreground="Gray" x:Name="txtHintLogin" VerticalAlignment="Top" Width="92" Grid.Column="2" Grid.Row="1" HorizontalAlignment="Left" Height="13" Margin="3,0,0,0"/>

<TextBlock Text="Пароль" Foreground="Gray" x:Name="txtHintPassword" VerticalAlignment="Top" Width="41" Grid.Column="2" Grid.Row="1" HorizontalAlignment="Left" Height="16" Margin="3,28,0,0"/>

<Button x:Name="LoginButton" Content="Вход" Grid.Column="2" HorizontalAlignment="Left" Margin="0,80,0,0" Grid.Row="1" VerticalAlignment="Top" Width="120" Height="26" Background="#FFE0A9AF" Click="LoginButton\_Click">

<Button.ToolTip>

<ToolTip Content="Кнопка для входа"/>

</Button.ToolTip>

</Button>

<Button x:Name="RegisterButton" Content="Регистрация" Grid.Column="2" HorizontalAlignment="Left" Margin="0,111,0,0" Grid.Row="1" VerticalAlignment="Top" Width="120" Height="26" Background="#FFE0A9AF">

<Button.ToolTip>

<ToolTip Content="Кнопка для регистрации"/>

</Button.ToolTip>

</Button>

</Grid>

</Page>

Листинг 3. Файл AuthPage.xaml

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace \_619\_Gorlanov\_Derevyagin.Pages

{

/// <summary>

/// Логика взаимодействия для AuthPage.xaml

/// </summary>

public partial class AuthPage : Page

{

public AuthPage()

{

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 (PasswordBox.Password.Length > 0)

{

txtHintPassword.Visibility = Visibility.Hidden;

}

}

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

{

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

{

MessageBox.Show("Введите логин и пароль!");

return;

}

using (var db = new Entities())

{

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

if (user == null)

{

MessageBox.Show("Пользователь с такими данными не найден!");

return;

}

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

switch(user.Role)

{

case "Администратор":

NavigationService?.Navigate(new AdminPage());

break;

case "Пользователь":

NavigationService?.Navigate(new UserPage());

break;

}

}

}

}

}

Листинг 4. Файл AuthPage.xaml.cs

<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="#b5d5ca"/>

<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>

</ResourceDictionary>

Листинг 5. Файл Dictionary.xaml

<Application x:Class="\_619\_Gorlanov\_Derevyagin.App"

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

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

xmlns:local="clr-namespace:\_619\_Gorlanov\_Derevyagin"

StartupUri="MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Dictionary.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

Листинг 6. Файл App.xaml

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: В.В. Горланов,

Е.С. Деревягин

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.

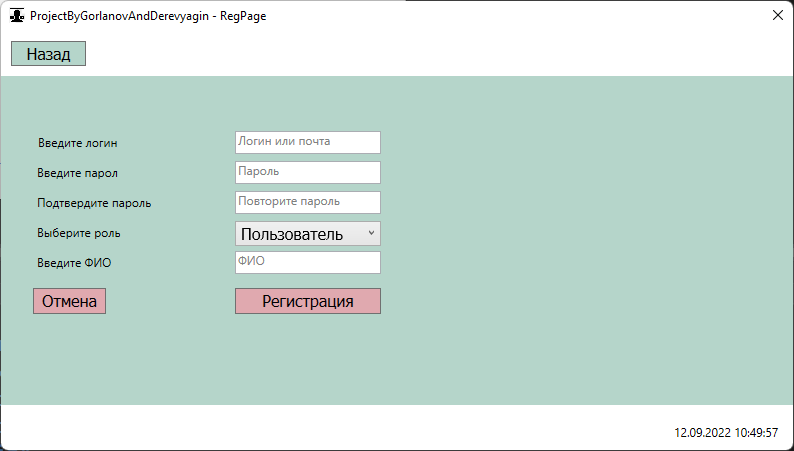


Рисунок 1. Страница регистрации

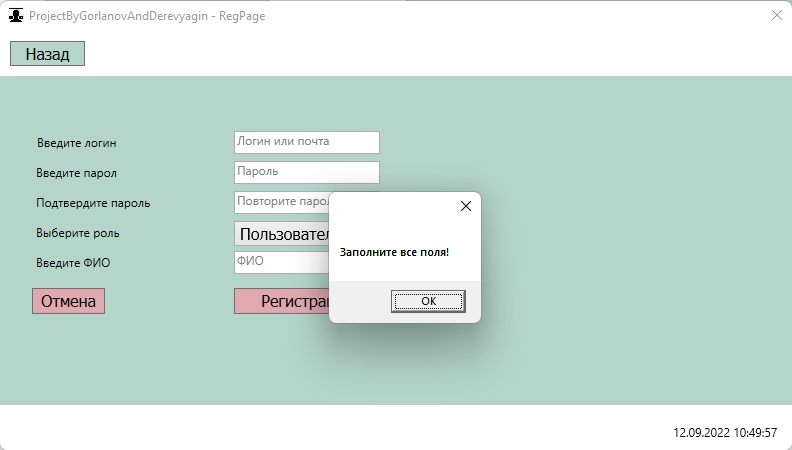


Рисунок 2. Демонстрация окна, если не заполнить поля

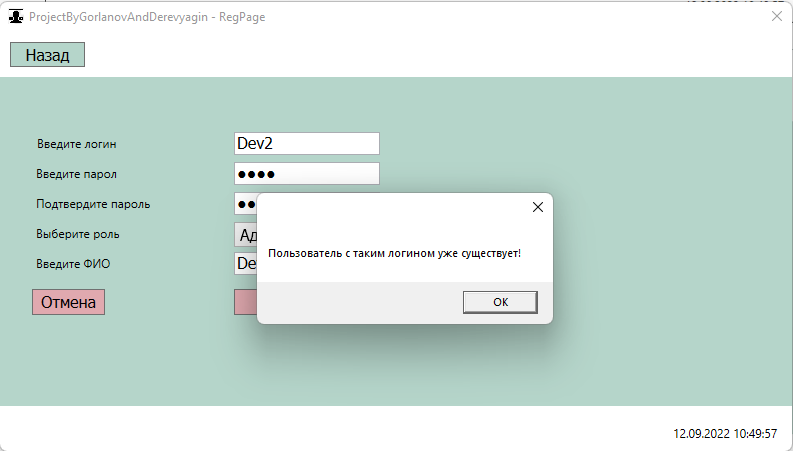


Рисунок 3. Проверка на существующего пользователя

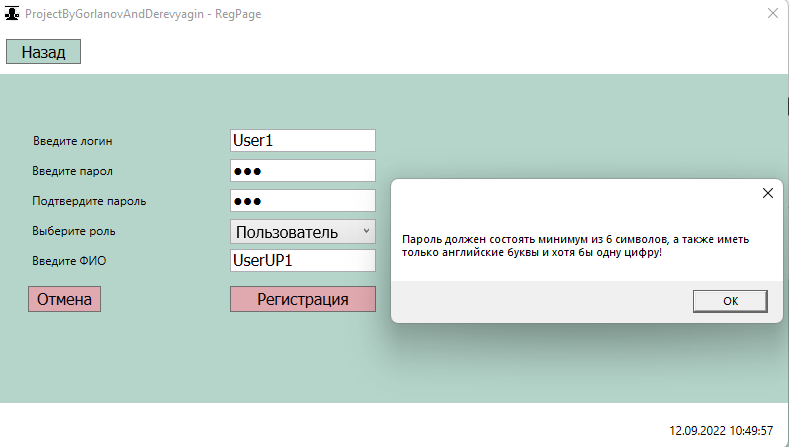


Рисунок 4. Проверка пароля

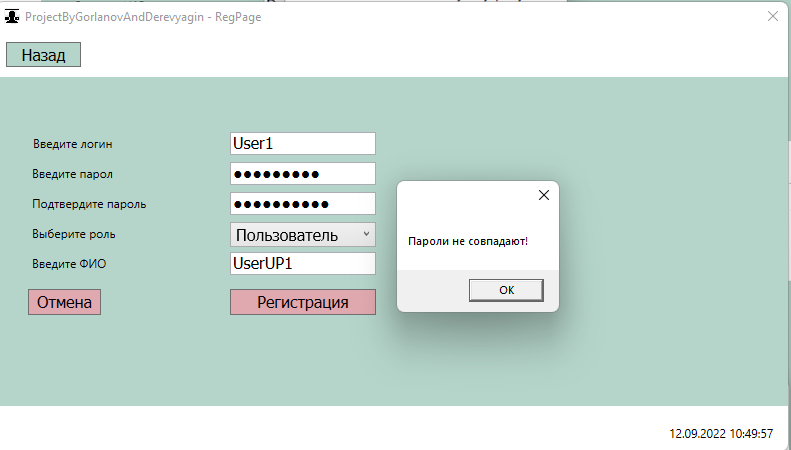


Рисунок 5. Проверка на совпадение паролей

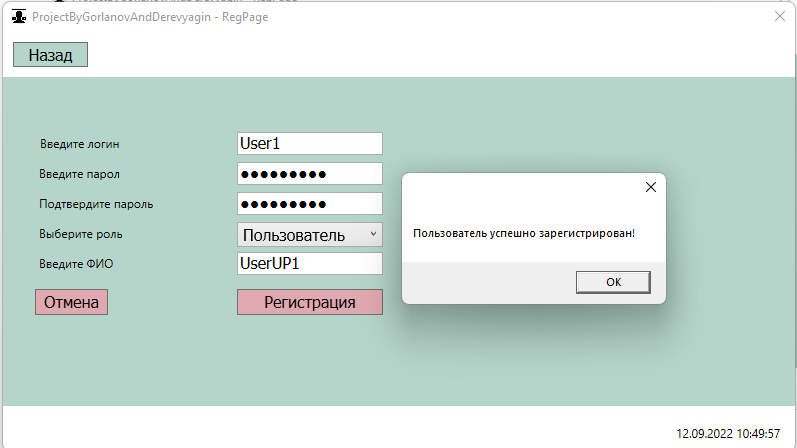


Рисунок 6. Успешная регистрация пользователя

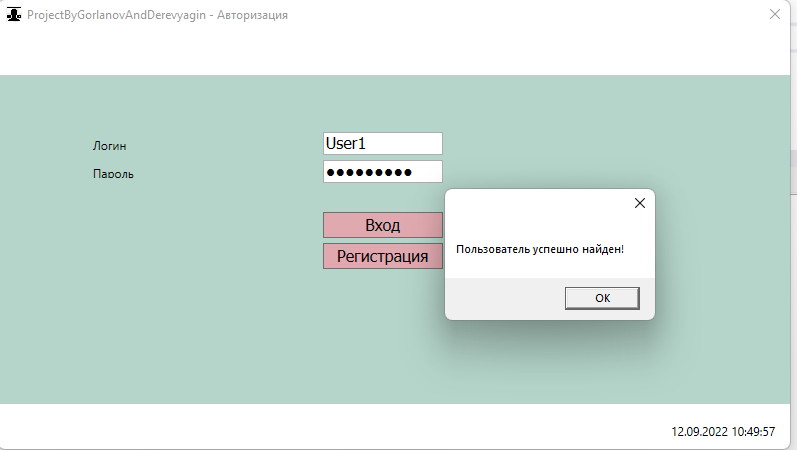


Рисунок 7. Вход под только что созданным пользователем

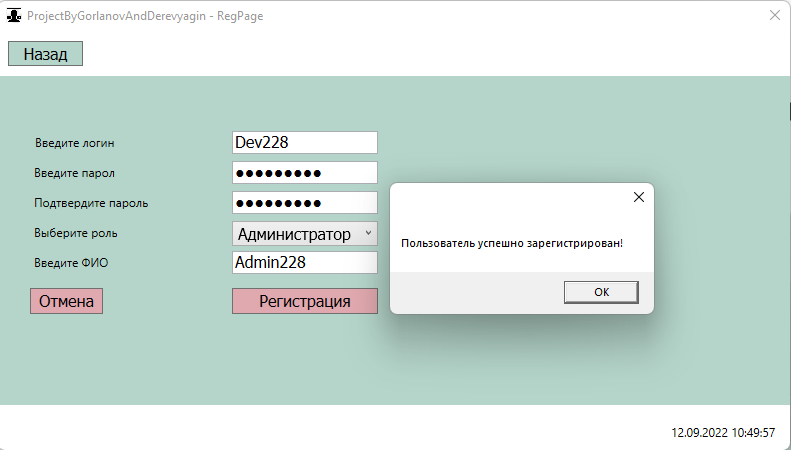


Рисунок 8. Регистрация админа

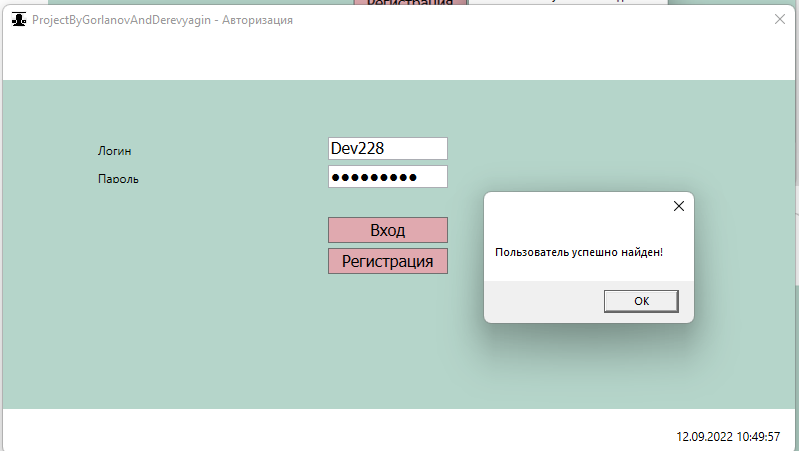


Рисунок 9. Вход под только что созданным админом

Изображение выглядит как стол

Автоматически созданное описание

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>

</ResourceDictionary>

Листинг 1. Файл Dictionary.xaml

<Page x:Class="\_619\_Gorlanov\_Derevyagin.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"

xmlns:local="clr-namespace:\_619\_Gorlanov\_Derevyagin.Pages"

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="50"/>

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

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

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

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

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="ButtonEdit" Content="Редактировать"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Grid.Row="1" x:Name="AddButton" Content="Добавить" Background="#FFE0A9AF" Margin="10,9,0,9" Height="Auto" HorizontalAlignment="Left" Width="100"/>

<Button Grid.Row="1" x:Name="DelButton" Content="Удалить" Background="#FFE0A9AF" Margin="0,9,10,9" Height="Auto" HorizontalAlignment="Right" Width="100"/>

</Grid>

</Page>

Листинг 2. Файл AdminPage.xaml

using System.Linq;

using System.Windows.Controls;

namespace \_619\_Gorlanov\_Derevyagin.Pages

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public AdminPage()

{

InitializeComponent();

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

}

}

}

Листинг 3. Файл AdminPage.xaml.cs

<Page x:Class="\_619\_Gorlanov\_Derevyagin.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\_Gorlanov\_Derevyagin.Pages"

mc:Ignorable="d"

d:DesignHeight="300" d:DesignWidth="420"

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.Linq;

using System.Text.RegularExpressions;

using System.Windows;

using System.Windows.Controls;

namespace \_619\_Gorlanov\_Derevyagin.Pages

{

/// <summary>

/// Логика взаимодействия для RegPage.xaml

/// </summary>

public partial class RegPage : Page

{

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 EntitiesHome())

{

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;

}

User userObject = new User

{

FIO = TextBoxFIO.Text,

Login = TextBoxLogin.Text,

Password = PasswordBox.Password,

Role = ComboBoxRole.Text

};

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: В.В. Горланов,

Е.С. Деревягин

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.

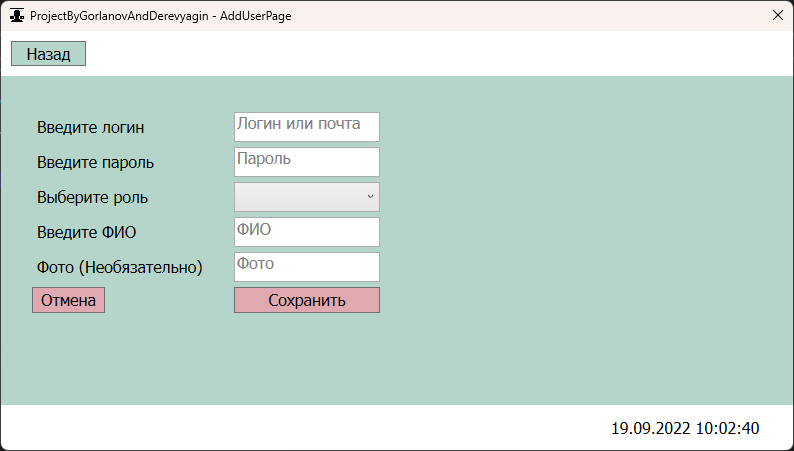


Рисунок 1. Страница добавления пользователя

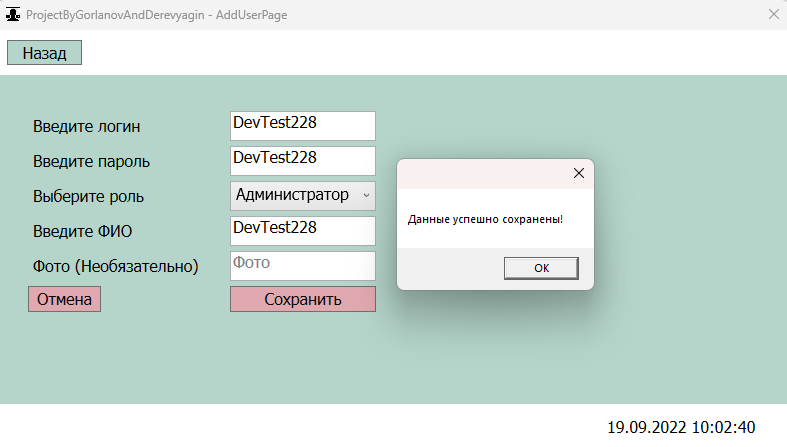


Рисунок 2. Проверка работы добавления нового пользователя

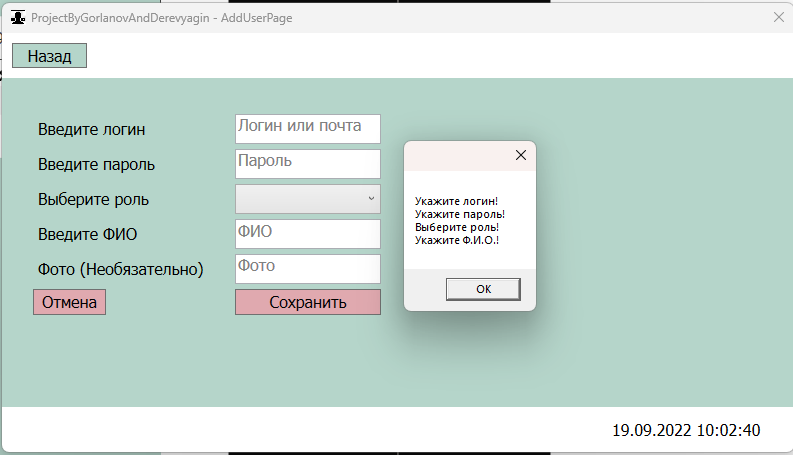


Рисунок 3. Проверка заполнения полей

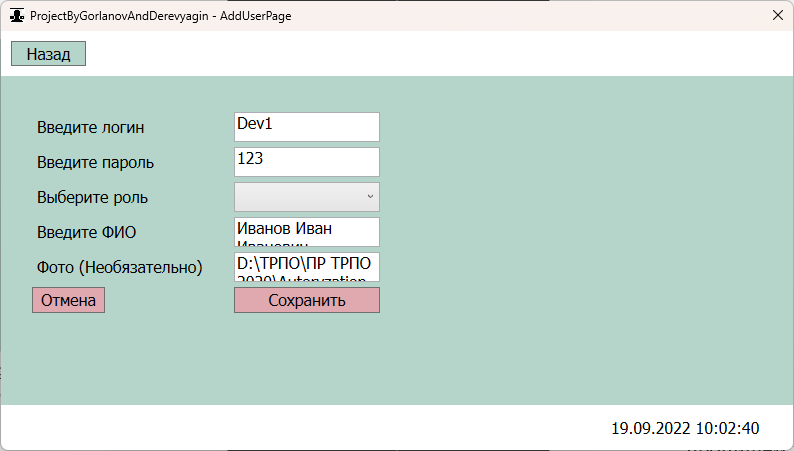


Рисунок 4. Страница редактирования пользователя

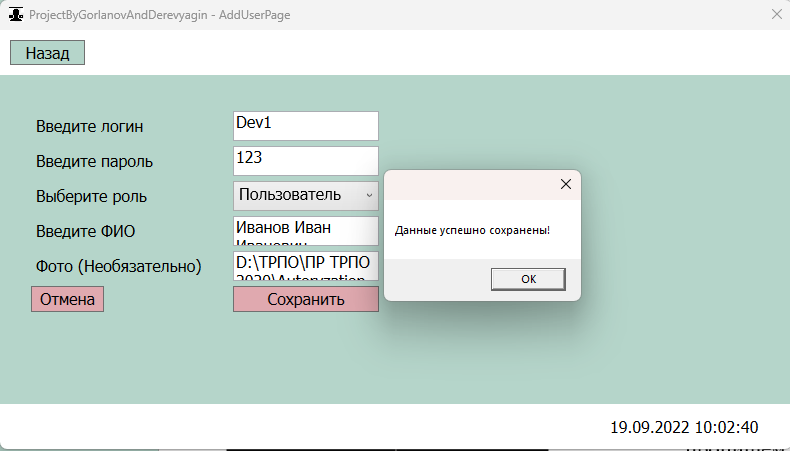


Рисунок 5. Проверка работы редактирования пользователя

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6. Удаление пользователя

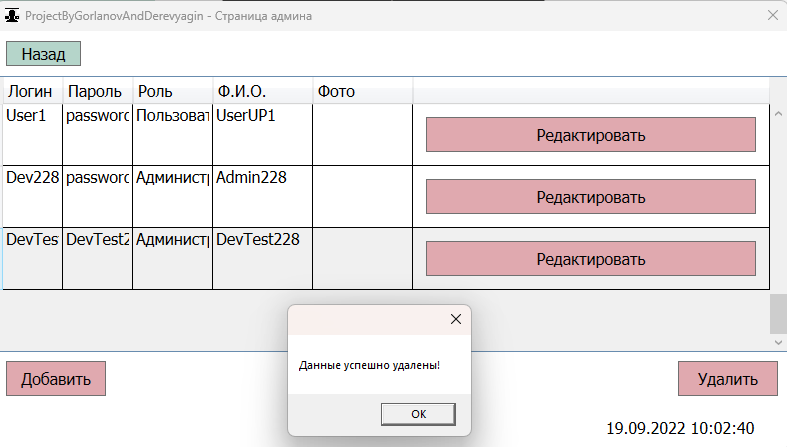


Рисунок 7. Проверка удаления пользователя

Изображение выглядит как стол

Автоматически созданное описание

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\_Gorlanov\_Derevyagin.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\_Gorlanov\_Derevyagin.Pages"

mc:Ignorable="d"

d:DesignHeight="300" d:DesignWidth="420"

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}" 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 FIO}" 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;

using System.Text;

using System.Windows;

using System.Windows.Controls;

namespace \_619\_Gorlanov\_Derevyagin.Pages

{

/// <summary>

/// Логика взаимодействия для AddUserPage.xaml

/// </summary>

public partial class AddUserPage : Page

{

private User \_currentUser = new User();

public AddUserPage(User selectedUser)

{

InitializeComponent();

if(selectedUser != null)

{

\_currentUser = selectedUser;

}

DataContext = \_currentUser;

}

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 (PasswordBox.Text.Length > 0)

{

txtHintPassword.Visibility = Visibility.Hidden;

}

}

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

{

txtHintFIO.Visibility = Visibility.Visible;

if (TextBoxFIO.Text.Length > 0)

{

txtHintFIO.Visibility = Visibility.Hidden;

}

}

private void TextBoxPhoto\_TextChanged(object sender, RoutedEventArgs e)

{

txtHintPhoto.Visibility = Visibility.Visible;

if (TextBoxPhoto.Text.Length > 0)

{

txtHintPhoto.Visibility = Visibility.Hidden;

}

}

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

{

NavigationService?.Navigate(new AdminPage());

}

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

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentUser.Login))

{

errors.AppendLine("Укажите логин!");

}

if (string.IsNullOrWhiteSpace(\_currentUser.Password))

{

errors.AppendLine("Укажите пароль!");

}

if ((\_currentUser.Role == null) || (ComboBoxRole.Text == ""))

{

errors.AppendLine("Выберите роль!");

}

else

{

\_currentUser.Role = ComboBoxRole.Text;

}

if (string.IsNullOrWhiteSpace(\_currentUser.FIO))

{

errors.AppendLine("Укажите Ф.И.О.!");

}

//Проверяем переменную errors на наличие ошибок

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

//Добавляем в объект User новую запись

if (\_currentUser.ID == 0)

{

EntitiesHome.GetContext().User.Add(\_currentUser);

}

try

{

EntitiesHome.GetContext().SaveChanges();

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

}

catch (Exception ex)

{

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

}

}

}

}

Листинг 3. Файл AddUserPage.xaml.cs

<Page x:Class="\_619\_Gorlanov\_Derevyagin.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"

xmlns:local="clr-namespace:\_619\_Gorlanov\_Derevyagin.Pages"

mc:Ignorable="d"

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

Title="Страница админа" IsVisibleChanged="Page\_IsVisibleChanged">

<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 FIO}" 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 4. Файл AdminPage.xaml

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace \_619\_Gorlanov\_Derevyagin.Pages

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public AdminPage()

{

InitializeComponent();

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

}

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

{

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

{

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

DataGridUser.ItemsSource = EntitiesHome.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

{

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

EntitiesHome.GetContext().SaveChanges();

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

DataGridUser.ItemsSource = EntitiesHome.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: В.В. Горланов,

Е.С. Деревягин

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.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1. Данные в БД

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Страница пользователя

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Страница пользователя с фотографиями пользователей

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Проверка работы сортировки по пользователям

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Проверка работы сортировки по ФИО

<Page x:Class="\_619\_Gorlanov\_Derevyagin.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\_Gorlanov\_Derevyagin.Pages"

mc:Ignorable="d"

d:DesignHeight="500" d:DesignWidth="1200"

Title="Страница пользователя">

<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="TextBoxFIO" VerticalContentAlignment="Center" Margin="0 10 20 10" HorizontalAlignment="Left" Height="30" TextWrapping="Wrap" VerticalAlignment="Top" Width="134" Grid.Row="0" TextChanged="TextBoxFIO\_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>

<CheckBox x:Name="CheckBoxRole" VerticalContentAlignment="Center" Margin="0 10 20 10" Content="Только по пользователям" HorizontalAlignment="Center" Height="30" VerticalAlignment="Center" Width="212" Grid.Row="0" Checked="CheckBoxRole\_Checked" Unchecked="CheckBoxRole\_Unchecked">

<CheckBox.ToolTip>

<ToolTip Content="Фильтрация по роли"/>

</CheckBox.ToolTip>

</CheckBox>

<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>/Resources/DefaultPhoto.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock VerticalAlignment="Center" TextAlignment="Center" Width="450" Grid.Row="0" TextWrapping="Wrap" Text="{Binding FIO}" 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.Linq;

using System.Windows;

using System.Windows.Controls;

namespace \_619\_Gorlanov\_Derevyagin.Pages

{

/// <summary>

/// Логика взаимодействия для UserPage.xaml

/// </summary>

public partial class UserPage : Page

{

public UserPage()

{

InitializeComponent();

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

ListUser.ItemsSource = currentUsers;

ComboBoxTypeSort.SelectedIndex = 0;

CheckBoxRole.IsChecked = false;

UpdateUsers();

}

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

{

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()

{

//загружаем всех пользователей в список

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

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

currentUsers = currentUsers.Where(x => x.FIO.ToLower().Contains(TextBoxFIO.Text.ToLower())).ToList();

//обрабатываем фильтр по одной роли пользователей

if (CheckBoxRole.IsChecked.Value)

currentUsers = currentUsers.Where(x => x.Role.Contains("Пользователь")).ToList();

//осуществляем сортировку в зависимости от выбора пользователя

if (ComboBoxTypeSort.SelectedIndex == 0)

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

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

}

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

{

ComboBoxTypeSort.SelectedIndex = 0;

CheckBoxRole.IsChecked = false;

TextBoxFIO.Text = null;

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: Реализация диаграмм в приложении WPF. Программная работа с документами Word и Excel с помощью библиотек Microsoft.Office.Interop.Word и Microsoft.Office.Interop.Excel**

Студент: В.В. Горланов,

Е.С. Деревягин

Группа: ИСИП-619

Преподаватель: Т.Г. Аксёнова

Дата: 03.10.2022

**Цель работы:** приобрести практические навыки реализации диаграмм, формирования текстовых документов Microsoft Word и электронных таблиц Microsoft Excel в приложениях WPF.

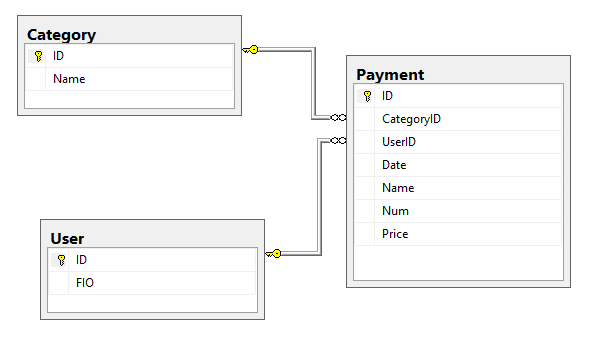
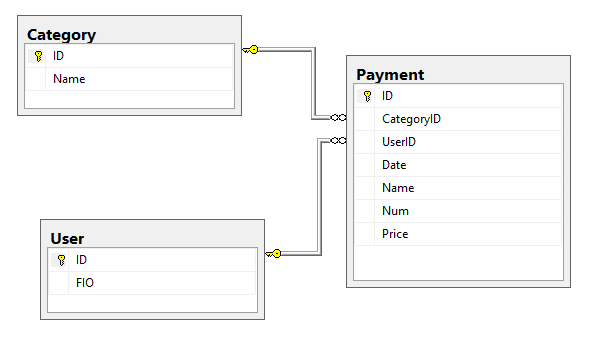


Рисунок 1. ER-диаграмма

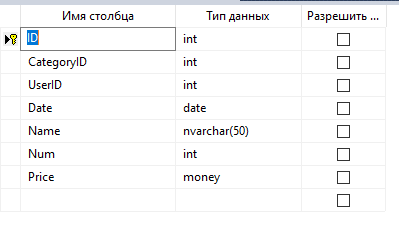


Рисунок 2. Проект таблицы Payments

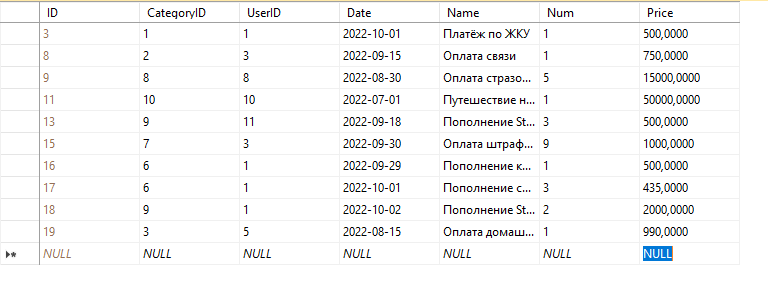


Рисунок 3. Данные в таблице Payments

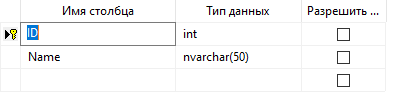


Рисунок 4. Проект таблицы Category

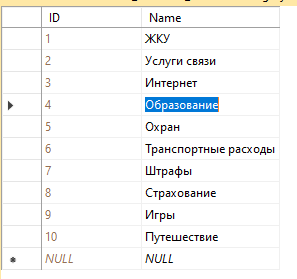


Рисунок 5. Данные в таблице Category

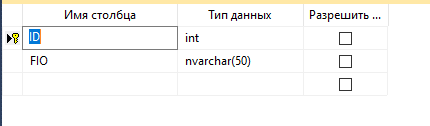


Рисунок 6. Проект таблицы User

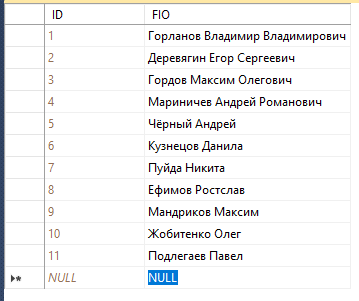


Рисунок 7. Данные в таблице User

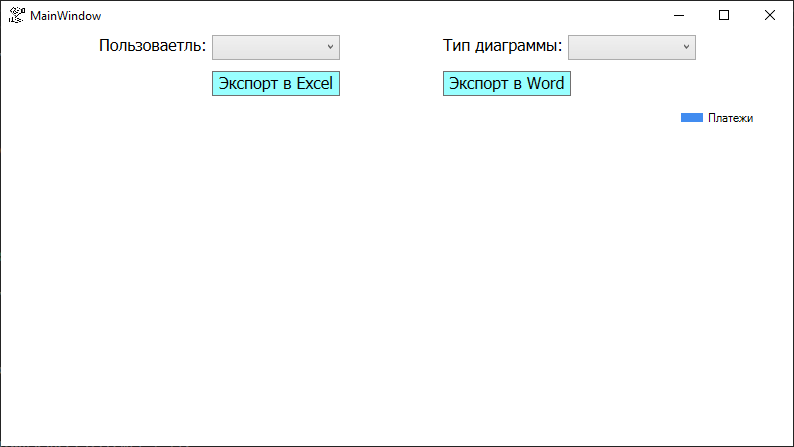


Рисунок 8. Скриншот программы

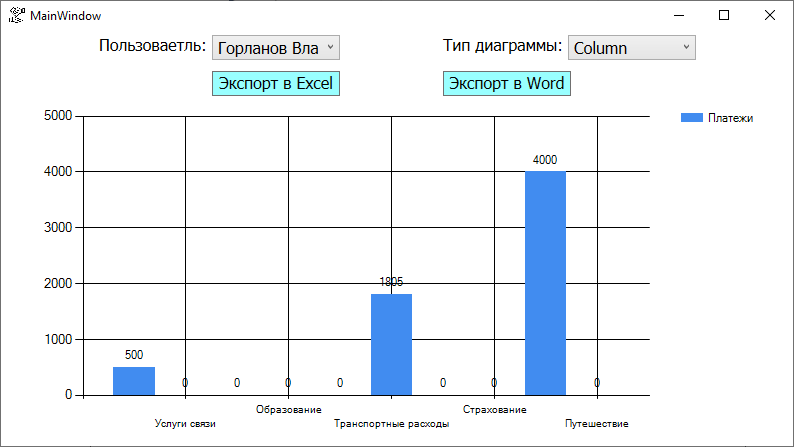


Рисунок 9. Скриншот работы программы 1

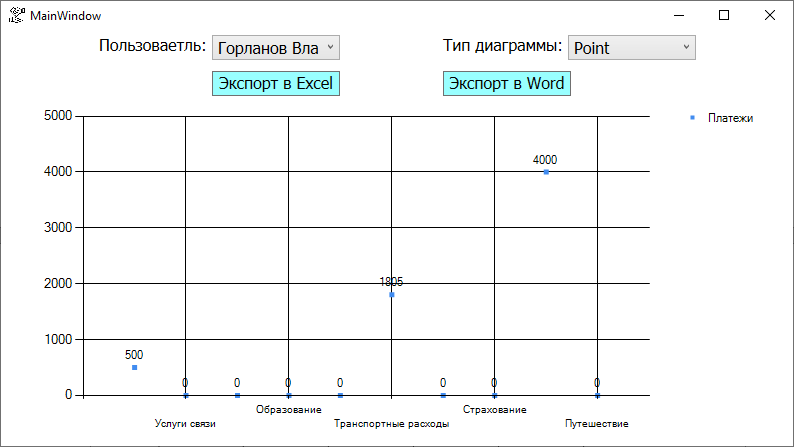


Рисунок 10. Скриншот работы программы 2

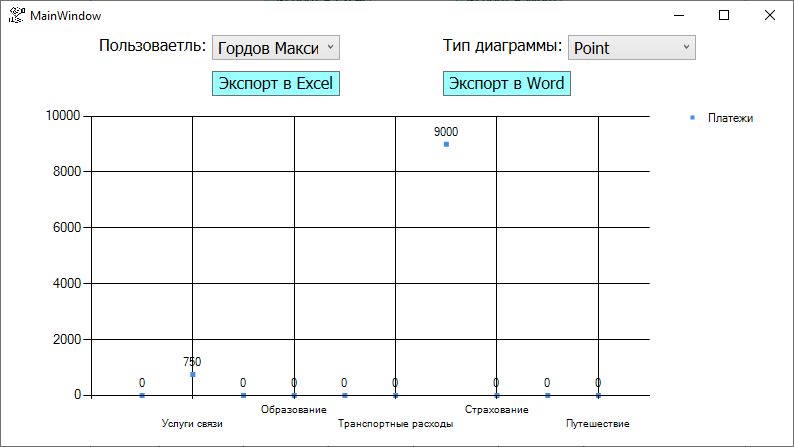


Рисунок 11. Скриншот работы программы 3

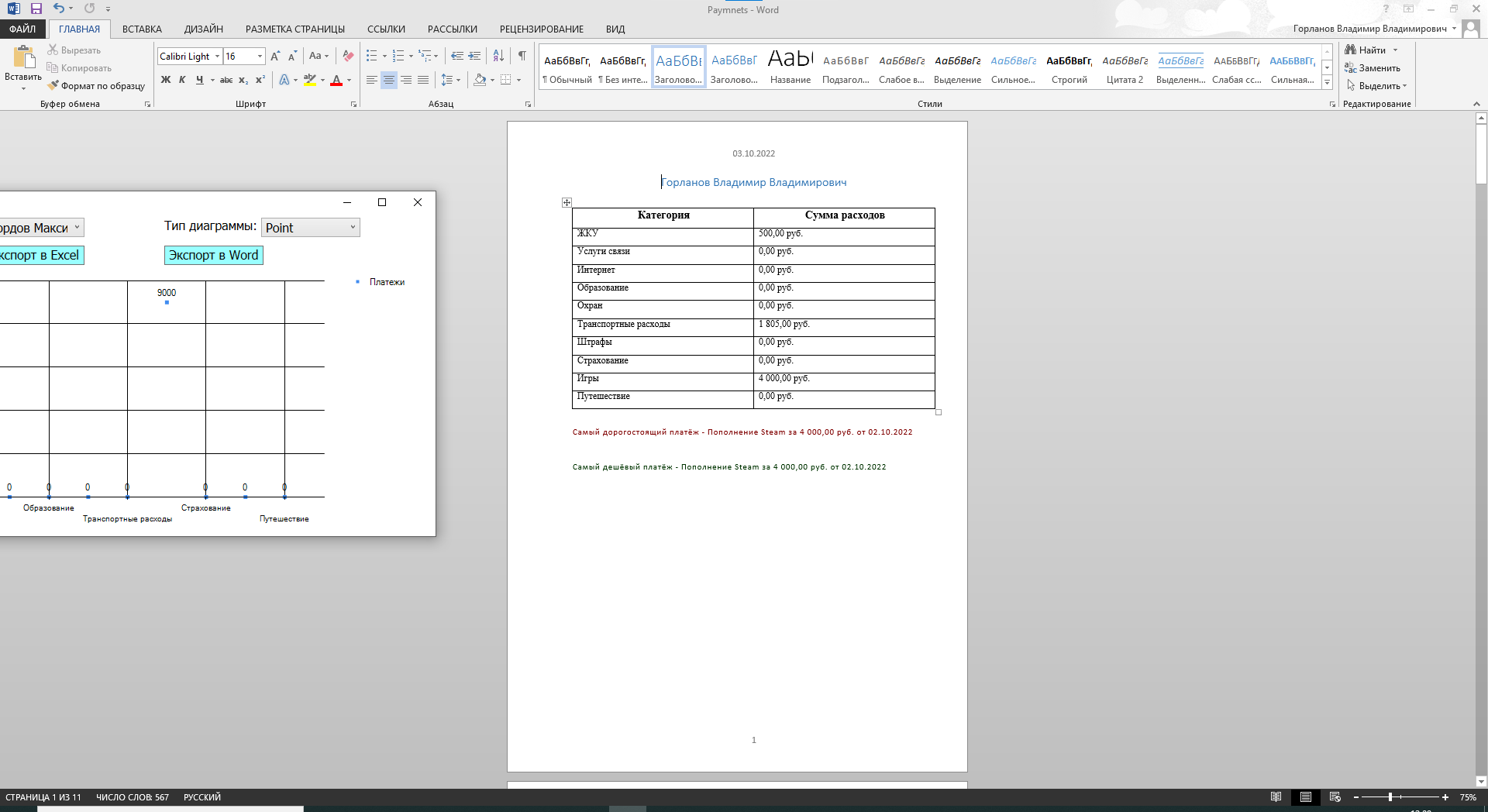


Рисунок 12. Скриншот экспорта в Word

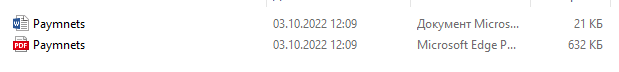


Рисунок 13. Скриншот сохранения файла Word

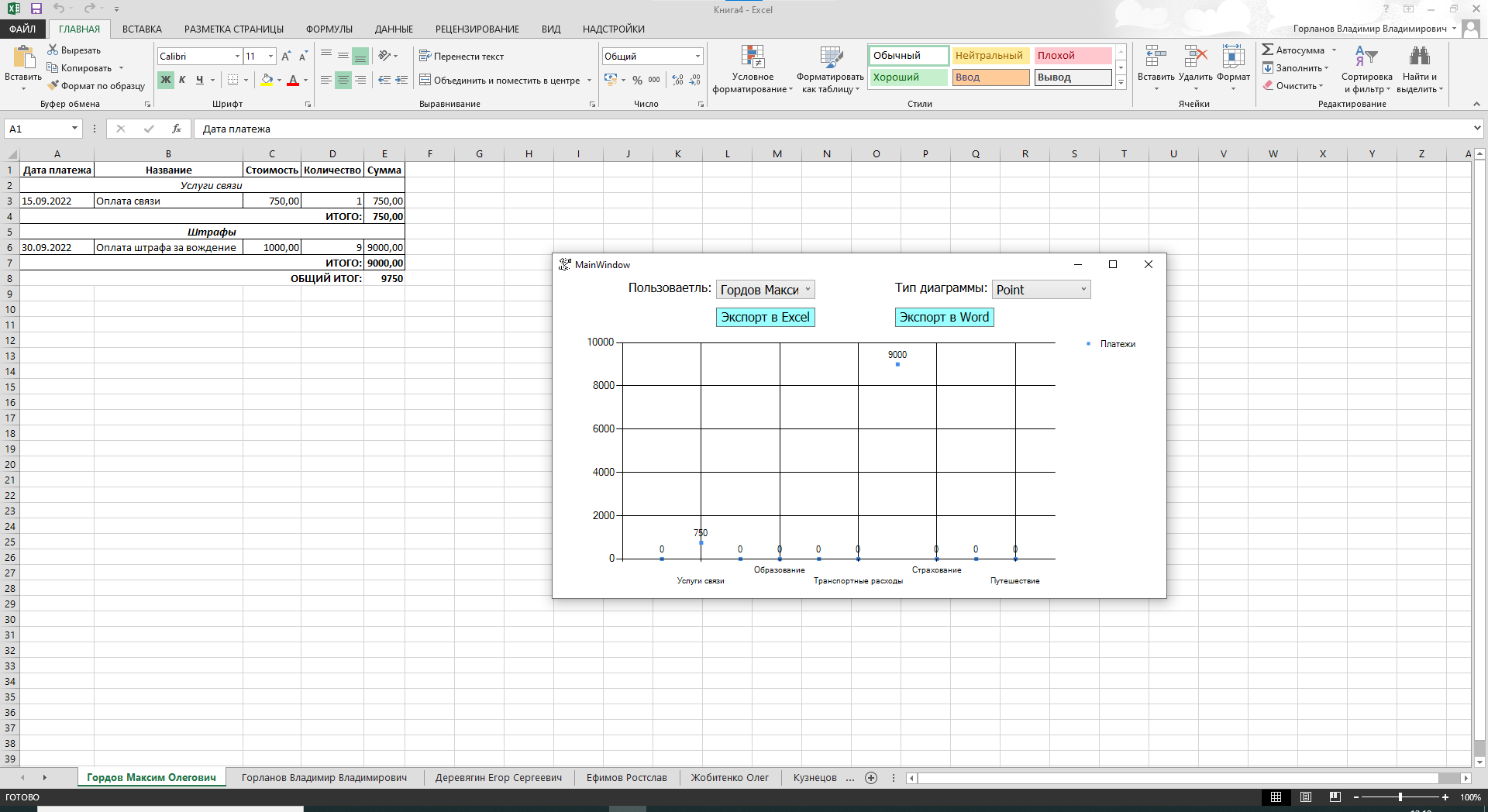


Рисунок 14. Скриншот экспорта в Excel

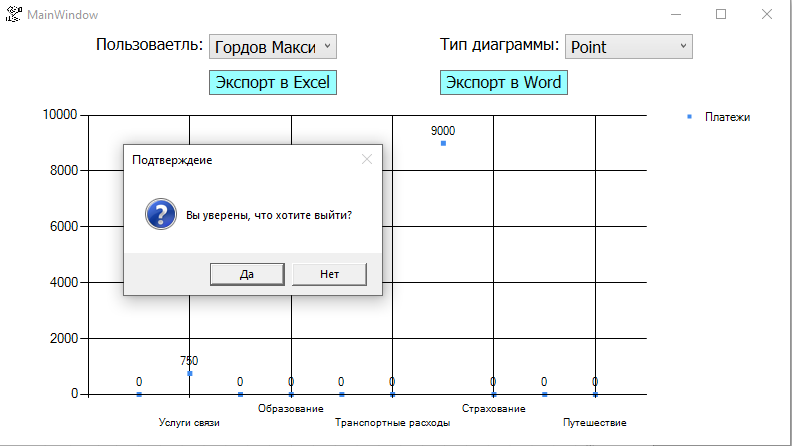


Рисунок 15. Скриншот выхода из программы

<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="25"/>

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

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

<Setter Property="Background" Value="#99ffff"/>

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

</Style>

<Style TargetType="TextBox">

<Setter Property="Height" Value="25"/>

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

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

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

</Style>

<Style TargetType="ComboBox">

<Setter Property="Height" Value="25"/>

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

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

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

</Style>

<Style TargetType="TextBlock">

<Setter Property="Height" Value="25"/>

<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

<Application x:Class="Payment\_619Gorlanov.App"

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

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

xmlns:local="clr-namespace:Payment\_619Gorlanov"

StartupUri="MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Dictionary.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

Листинг 2. Файл App.xaml

<Window x:Class="Payment\_619Gorlanov.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:charts="clr-namespace:System.Windows.Forms.DataVisualization.Charting;assembly=System.Windows.Forms.DataVisualization"

xmlns:local="clr-namespace:Payment\_619Gorlanov"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" SizeToContent="WidthAndHeight" Closing="Window\_Closing" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60\*"/>

<RowDefinition Height="350"/>

</Grid.RowDefinitions>

<Button x:Name="ExcelButton" Grid.Row="0" Content="Экспорт в Excel" HorizontalAlignment="Left" Margin="211,40,0,0" VerticalAlignment="Top" Width="128" Click="ExcelButton\_Click"/>

<Button x:Name="WordButton" Grid.Row="0" Content="Экспорт в Word" HorizontalAlignment="Left" Margin="442,40,0,0" VerticalAlignment="Top" Width="128" Click="WordButton\_Click"/>

<ComboBox x:Name="CmbUser" Grid.Row="0" SelectionChanged="UpdateChart" DisplayMemberPath="FIO" HorizontalAlignment="Left" VerticalAlignment="Top" Width="128" Margin="211,4,0,0"/>

<ComboBox x:Name="CmbDiagram" Grid.Row="0" SelectionChanged="UpdateChart" HorizontalAlignment="Left" VerticalAlignment="Top" Width="128" Margin="567,4,0,0"/>

<TextBlock Grid.Row="0" HorizontalAlignment="Left" Margin="98,4,0,0" TextWrapping="Wrap" Text="Пользоваетль:" VerticalAlignment="Top"/>

<TextBlock Grid.Row="0" HorizontalAlignment="Left" Margin="442,4,0,0" TextWrapping="Wrap" Text="Тип диаграммы:" VerticalAlignment="Top"/>

<WindowsFormsHost Grid.Row="1" HorizontalAlignment="Left" Height="350" VerticalAlignment="Top" Width="792">

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

<charts:Chart.Legends>

<charts:Legend>

</charts:Legend>

</charts:Chart.Legends>

</charts:Chart>

</WindowsFormsHost>

</Grid>

</Window>

Листинг 3. Файл MainWindows.xaml

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace Payment\_619Gorlanov

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private Entities \_context = new Entities();

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));

}

private void UpdateChart(object sender, SelectionChangedEventArgs e)

{

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

{

Series currentSeries = ChartPayments.Series.FirstOrDefault();

currentSeries.ChartType = currnetType;

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

{

DialogResult res = System.Windows.Forms.MessageBox.Show("Вы уверены, что хотите выйти?", "Подтверждеие", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (res != System.Windows.Forms.DialogResult.Yes)

{

e.Cancel = true;

}

}

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

{

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

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

var application = new Word.Application();

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

foreach (var user in allUser)

{

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

Word.Range userRange = userParagraph.Range;

userRange.Text = user.FIO;

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("N2") + " руб.";

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

cellRange.Font.Size = 12;

}

document.Paragraphs.Add();

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 платёж - {maxPayment.Name} за {(maxPayment.Price \* maxPayment.Num).ToString("N2")} руб. от {maxPayment.Date.ToString("dd.MM.yyyy")}";

maxPaymentParagraph.set\_Style("Подзаголовок");

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

maxPaymentRange.InsertParagraphAfter();

}

document.Paragraphs.Add();

foreach (Word.Section section in document.Sections)

{

Word.Range headerRange = section.Headers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;

headerRange.Fields.Add(headerRange, Word.WdFieldType.wdFieldPage);

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

headerRange.Font.Size = 12;

headerRange.Text = DateTime.Today.ToString("dd.MM.yyyy");

}

foreach (Word.Section section in document.Sections)

{

Word.HeaderFooter ftr = section.Footers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary];

ftr.PageNumbers.RestartNumberingAtSection = true;

ftr.PageNumbers.StartingNumber = 1;

object TotalPages = Word.WdFieldType.wdFieldSectionPages;

object CurrentPage = Word.WdFieldType.wdFieldPage;

Word.Range rngCurrSecFooter = ftr.Range;

rngCurrSecFooter.Fields.Add(rngCurrSecFooter, ref CurrentPage);

Word.Range footerRange = section.Footers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;

footerRange.Font.Size = 12;

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

}

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("N2")} " + $"руб. от {minPayment.Date.ToString("dd.MM.yyyy")}";

minPaymentParagraph.set\_Style("Подзаголовок");

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

minPaymentRange.InsertParagraphAfter();

}

if (user != allUser.LastOrDefault())

{

document.Words.Last.InsertBreak(Word.WdBreakType.wdPageBreak);

}

application.Visible = true;

document.SaveAs2(@"C:\D\Paymnets.docx");

document.SaveAs2(@"C:\D\Paymnets.pdf", Word.WdExportFormat.wdExportFormatPDF);

}

}

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

{

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

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);

int suma = 0;

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("dd.MM.yyyy");

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++;

suma += Convert.ToInt32(payment.Num) \* Convert.ToInt32(payment.Price);

}

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})";

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

sumaRange.Merge();

sumaRange.Value = "ОБЩИЙ ИТОГ:";

sumaRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

worksheet.Cells[5][startRowIndex + 1].Value = suma;

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

sumaRange.Font.Bold = worksheet.Cells[5][startRowIndex+1].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();

}

application.Visible = true;

}

}

}

}

Листинг 4. Файл MainWindow.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 №19-20

**Тема: Разработка Desktop-приложения**

Студент: В.В. Горланов,

Е.С. Деревягин

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

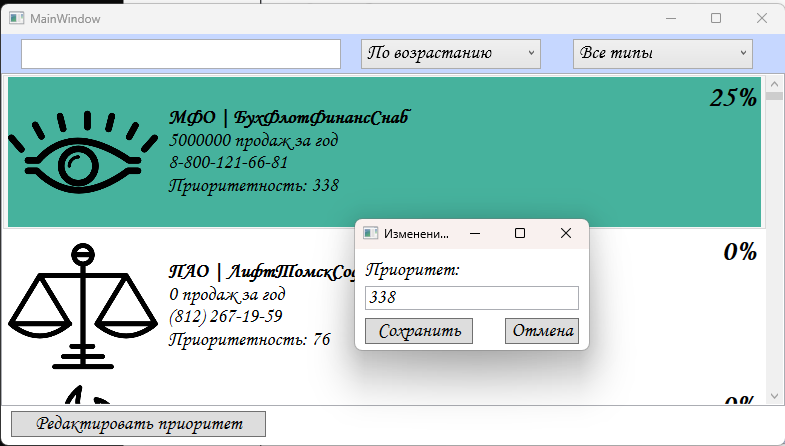
Автоматически созданное описание

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 №21-27

**Тема: Разработка Android-приложения**

Студент: В.В. Горланов,

Е.С. Деревягин

Группа: ИСИП-619

Преподаватель: П.А. Костиков

Дата: 31.10.2022 – 07.11.2022

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

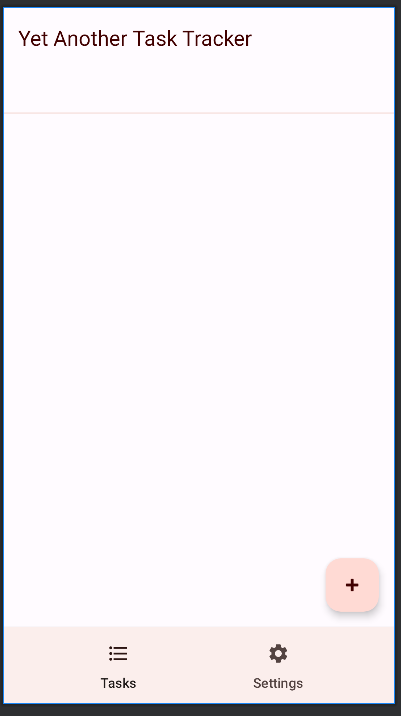


Рисунок 1. Итоговый вид MainActivity

Изображение выглядит как квадрат

Автоматически созданное описание

Рисунок 2. Итоговый вид ListFragment

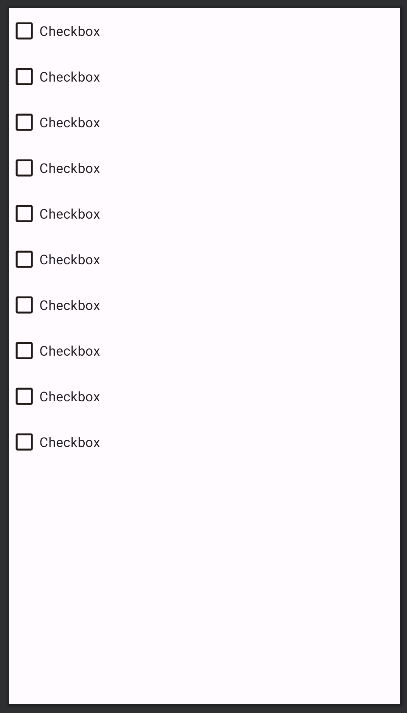


Рисунок 3. Итоговый вид TaskFragment

<?xml version="1.0" encoding="utf-8"?>

<layout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools">

<androidx.constraintlayout.widget.ConstraintLayout

android:id="@+id/content"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<com.google.android.material.appbar.AppBarLayout

android:id="@+id/appBar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

tools:ignore="MissingConstraints">

<com.google.android.material.appbar.MaterialToolbar

android:id="@+id/toolbar"

app:title="@string/app\_full\_name"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

app:titleTextColor="?attr/colorOnPrimaryContainer" />

</com.google.android.material.appbar.AppBarLayout>

<androidx.fragment.app.FragmentContainerView

android:id="@+id/fragmentContainerView"

android:name="androidx.navigation.fragment.NavHostFragment"

android:layout\_width="0dp"

android:layout\_height="0dp"

app:defaultNavHost="true"

app:layout\_constraintBottom\_toTopOf="@+id/bottomNavigation"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/appBar"

app:navGraph="@navigation/nav" />

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottomNavigation"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/colorOnSurfaceInverse"

app:layout\_constraintBottom\_toBottomOf="parent"

app:menu="@menu/bottom\_navigation\_menu"

tools:layout\_editor\_absoluteX="0dp"

app:itemRippleColor="@null"/>

</androidx.constraintlayout.widget.ConstraintLayout>

</layout>

Листинг 1. Файл activity\_main.xml

package ru.net2fox.trackerapp

import android.app.Activity

import android.content.SharedPreferences

import android.os.Bundle

import android.util.Log

import androidx.appcompat.app.AppCompatActivity

import androidx.appcompat.app.AppCompatDelegate

import androidx.databinding.DataBindingUtil

import androidx.navigation.fragment.NavHostFragment

import androidx.navigation.ui.setupWithNavController

import ru.net2fox.trackerapp.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding : ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = DataBindingUtil.setContentView(this, R.layout.activity\_main)

val activityPreferences: SharedPreferences = getSharedPreferences("ru.net2fox.trackerapp\_preferences", Activity.MODE\_PRIVATE)

Log.d("TrackerApp", activityPreferences.getString("dark\_theme", "").toString())

if (activityPreferences.getString("dark\_theme", "").toString() == "system") {

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_FOLLOW\_SYSTEM)

} else if (activityPreferences.getString("dark\_theme", "").toString() == "on") { AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_YES)

} else {

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO)

}

val navView = binding.bottomNavigation

val navHostFragment = supportFragmentManager.findFragmentById(R.id.fragmentContainerView) as NavHostFragment

val navController = navHostFragment.navController

navView.setupWithNavController(navController)

}

}

Листинг 2. Файл MainActiviy.kt

<?xml version="1.0" encoding="utf-8"?>

<layout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".ListFragment">

<com.google.android.material.tabs.TabLayout

android:id="@+id/tabs"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:tabInlineLabel="true"

app:tabMode="scrollable"/>

<androidx.viewpager2.widget.ViewPager2

android:id="@+id/viewPager"

android:layout\_width="0dp"

android:layout\_height="0dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/tabs" />

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/fab"

style="?attr/floatingActionButtonStyle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

android:contentDescription="@string/app\_name"

android:src="@android:drawable/ic\_input\_add"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</layout>

Листинг 3. Файл fragment\_list.xml

package ru.net2fox.trackerapp

import android.annotation.SuppressLint

import android.os.Bundle

import android.view.LayoutInflater

import android.view.MotionEvent

import android.view.View

import android.view.ViewGroup

import androidx.appcompat.app.AlertDialog

import androidx.databinding.DataBindingUtil

import androidx.fragment.app.Fragment

import androidx.fragment.app.viewModels

import androidx.lifecycle.Observer

import com.google.android.material.dialog.MaterialAlertDialogBuilder

import com.google.android.material.tabs.TabLayoutMediator

import com.google.android.material.textfield.TextInputEditText

import com.google.android.material.textfield.TextInputLayout

import ru.net2fox.trackerapp.databinding.FragmentListBinding

private const val KEY\_SELECTED\_TAB\_INDEX = "ru.net2fox.trackerapp.SELECTED\_TAB\_INDEX"

class ListFragment : Fragment() {

private lateinit var binding: FragmentListBinding

private lateinit var adapter: ViewPagerAdapter

private val trackerViewModel: TrackerViewModel by viewModels()

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?

): View {

binding = DataBindingUtil.inflate(layoutInflater, R.layout.fragment\_list, container, false)

val view = binding.root

adapter = ViewPagerAdapter(childFragmentManager, lifecycle, trackerViewModel)

binding.viewPager.adapter = adapter

binding.fab.setOnClickListener {

val builder = MaterialAlertDialogBuilder(requireContext())

builder.setTitle(R.string.create\_task\_dialog\_title)

val dialogView: View = inflater.inflate(R.layout.create\_alertdialog, null, false)

dialogView.findViewById<TextInputLayout>(R.id.textInputLayout).hint = getString(R.string.create\_task\_hint)

builder.setView(dialogView)

builder.setPositiveButton(R.string.ok\_dialog\_button, null)

builder.setNegativeButton(R.string.cancel\_dialog\_button, null)

val alertDialog: AlertDialog = builder.create()

alertDialog.show()

alertDialog.getButton(AlertDialog.BUTTON\_POSITIVE).setOnClickListener {

val taskNameEditText: TextInputEditText? = dialogView.findViewById(R.id.editText)

val wantToCloseDialog: Boolean = taskNameEditText?.text.toString().trim().isEmpty()

// Если EditText пуст, отключите закрытие при нажатии на позитивную кнопку

if (!wantToCloseDialog) {

alertDialog.dismiss()

trackerViewModel.getListId(binding.tabs.selectedTabPosition)

?.let { trackerViewModel.addTask(taskNameEditText?.text.toString(), it) }

}

}

}

TabLayoutMediator(binding.tabs, binding.viewPager) { tab, position ->

tab.text = trackerViewModel.getListById(position)?.list?.name

}.attach()

binding.tabs.selectTab(binding.tabs.getTabAt(savedInstanceState?.getInt(KEY\_SELECTED\_TAB\_INDEX) ?: 0))

binding.tabs.addTab(binding.tabs.newTab().setText(R.string.create\_list\_tab))

setTouchListenerToTab()

trackerViewModel.listsWithTasksLiveData.observe(

viewLifecycleOwner,

Observer { listTasks ->

listTasks?.let {

changeDataSet({ })

}

}

)

return view

}

@SuppressLint("NotifyDataSetChanged")

private fun changeDataSet(performChanges: () -> Unit, isNewListAdd: Boolean = false) {

performChanges()

binding.viewPager.adapter!!.notifyDataSetChanged()

if(isNewListAdd)

binding.viewPager.setCurrentItem(trackerViewModel.listSize ?: 0, false)

binding.tabs.addTab(binding.tabs.newTab().setText(R.string.create\_list\_tab))

setTouchListenerToTab()

if(trackerViewModel.listSize == 0) {

binding.fab.visibility = View.INVISIBLE

binding.tabs.setSelectedTabIndicatorHeight(((0 \* resources.displayMetrics.density).toInt()))

}

else if(trackerViewModel.listSize != 0) {

binding.fab.visibility = View.VISIBLE

binding.tabs.setSelectedTabIndicatorHeight(((3 \* resources.displayMetrics.density).toInt()))

}

}

@SuppressLint("ClickableViewAccessibility")

private fun setTouchListenerToTab() {

val tabStrip = binding.tabs.getChildAt(0)

if (tabStrip is ViewGroup) {

val childCount = tabStrip.childCount

for (i in 0 until childCount) {

val tabView = tabStrip.getChildAt(i)

tabView.setOnTouchListener { \_, event ->

if (event.action == MotionEvent.ACTION\_UP){

if(i == binding.tabs.tabCount - 1) {

val builder = MaterialAlertDialogBuilder(requireContext())

builder.setTitle(R.string.create\_list\_dialog\_title)

val dialogView: View = layoutInflater.inflate(R.layout.create\_alertdialog, null, false)

builder.setView(dialogView)

builder.setPositiveButton(R.string.ok\_dialog\_button, null)

builder.setNegativeButton(R.string.cancel\_dialog\_button, null)

val alertDialog: AlertDialog = builder.create()

alertDialog.show() alertDialog.getButton(AlertDialog.BUTTON\_POSITIVE).setOnClickListener {

val listNameEditText: TextInputEditText? = dialogView.findViewById(R.id.editText)

val wantToCloseDialog: Boolean = listNameEditText?.text.toString().trim().isEmpty()

// Если EditText пуст, отключите закрытие при нажатии на позитивную кнопку

if (!wantToCloseDialog) {

alertDialog.dismiss()

trackerViewModel.addList(listNameEditText?.text.toString())

}

}

return@setOnTouchListener true

}

}

return@setOnTouchListener false

}

}

}

}

override fun onSaveInstanceState(outState: Bundle) {

super.onSaveInstanceState(outState)

if(::binding.isInitialized) outState.putInt(KEY\_SELECTED\_TAB\_INDEX, binding.tabs.selectedTabPosition)

}

}

Листинг 4. Файл ListFragment.kt

<?xml version="1.0" encoding="utf-8"?>

<layout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".TaskFragment">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerViewTasks"

android:layout\_width="0dp"

android:layout\_height="0dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:listitem="@layout/item\_task" />

</androidx.constraintlayout.widget.ConstraintLayout>

</layout>

Листинг 5. Файл fragment\_task.xml

package ru.net2fox.trackerapp

import android.annotation.SuppressLint

import android.os.Bundle

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.CheckBox

import androidx.databinding.DataBindingUtil

import androidx.fragment.app.Fragment

import androidx.fragment.app.viewModels

import androidx.lifecycle.Observer

import androidx.recyclerview.widget.LinearLayoutManager

import androidx.recyclerview.widget.RecyclerView

import ru.net2fox.trackerapp.database.Task

import ru.net2fox.trackerapp.databinding.FragmentTaskBinding

private const val KEY\_LIST\_ID = "ru.net2fox.trackerapp.LIST\_ID"

class TaskFragment : Fragment() {

private lateinit var tasks: List<Task>

private lateinit var binding: FragmentTaskBinding

private lateinit var adapter: TaskRecyclerViewAdapter

private val taskDetailViewModel: TaskDetailViewModel by viewModels()

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?

): View {

binding = DataBindingUtil.inflate(layoutInflater, R.layout.fragment\_task, container, false)

val view = binding.root

binding.recyclerViewTasks.layoutManager = LinearLayoutManager(context)

//val divider = MaterialDividerItemDecoration(view.context, LinearLayoutManager.VERTICAL)

//binding.recyclerViewTasks.addItemDecoration(divider)

adapter = TaskRecyclerViewAdapter(taskDetailViewModel)

binding.recyclerViewTasks.adapter = adapter

taskDetailViewModel.listTasksLiveData.observe(

viewLifecycleOwner,

Observer { tasks ->

tasks?.let {

this.tasks = tasks.tasks

changeDataSet { }

}

}

)

val listId = arguments?.getInt(KEY\_LIST\_ID) ?: throw IllegalStateException()

taskDetailViewModel.loadTasks(listId)

return view

}

@SuppressLint("NotifyDataSetChanged")

private fun changeDataSet(performChanges: () -> Unit) {

performChanges()

binding.recyclerViewTasks.adapter!!.notifyDataSetChanged()

//setTouchListenerToItem()

}

fun setTouchListenerToItem() {

}

private inner class TaskViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {

private lateinit var task: Task

private val checkBoxView: CheckBox = itemView.findViewById(R.id.checkBox)

fun bind(task: Task) {

this.task = task

checkBoxView.text = task.name

checkBoxView.isChecked = task.isDone

checkBoxView.setOnCheckedChangeListener { buttonView, isChecked ->

task.isDone = checkBoxView.isChecked

taskDetailViewModel.saveTask(task)

}

}

}

private inner class TaskRecyclerViewAdapter(private val taskDetailViewModel: TaskDetailViewModel) : RecyclerView.Adapter<TaskViewHolder>() {

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): TaskViewHolder {

val itemView = LayoutInflater.from(parent.context).inflate(R.layout.item\_task, parent, false)

return TaskViewHolder(itemView)

}

override fun onBindViewHolder(holder: TaskViewHolder, position: Int) {

val task = taskDetailViewModel.listTasksLiveData.value?.tasks?.get(position)

if (task != null) {

holder.bind(task)

}

}

override fun getItemCount(): Int {

return taskDetailViewModel.listTasksLiveData.value?.tasks?.size ?: 0

}

}

companion object {

fun newInstance(param1: Int) =

TaskFragment().apply {

arguments = Bundle().apply {

putInt(KEY\_LIST\_ID, param1)

}

}

}

}

Листинг 6. Файл TaskFramgent.kt

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

# ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №28

**Тема: Работа с базами данных. Создание, модификация и удаление таблиц**

Студент: В.В. Горланов,

Е.С. Деревягин

Группа: ИСИП-619

Преподаватель: И.В. Сибирев

Дата: 14.11.2022

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

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Создание базы данных в MS SQL

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2. Создание таблицы Customers

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Создание таблицы Orders через запрос

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Результат выполнения запроса на создание таблицы Orders

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5. Таблица Orders

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6. Создание диаграммы базы данных

Изображение выглядит как текст, снимок экрана, внутренний

Автоматически созданное описание

Рисунок 7. Создание связей между таблицами

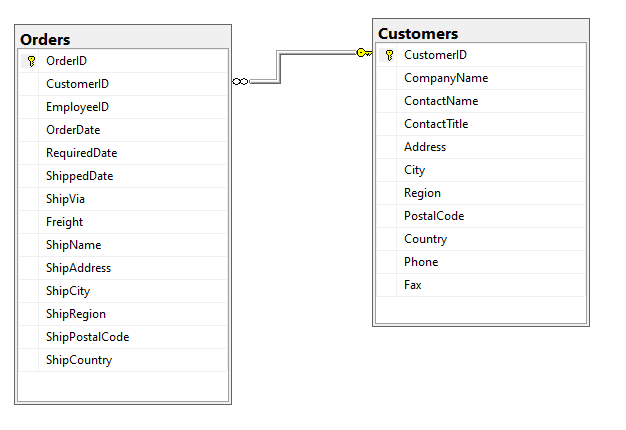


Рисунок 8. Связь между таблицами Orders и Customers

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

# ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №29

**Тема: Работа с базами данных. Создание запросов и представлений, использование механизма объединений**

Студент: В.В. Горланов,

Е.С. Деревягин

Группа: ИСИП-619

Преподаватель: И.В. Сибирев

Дата: 14.11.2022

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

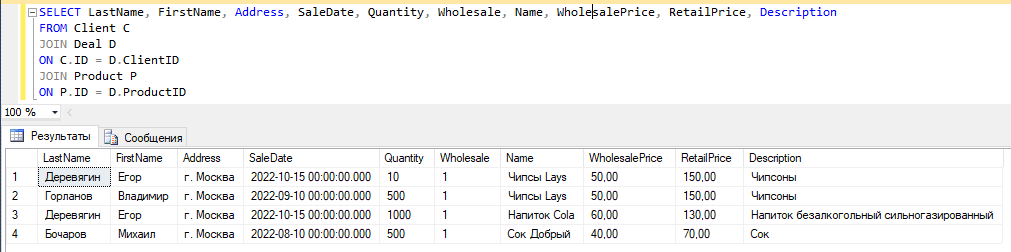
****

Рисунок 1. Вывод всей информации из БД в удобном виде

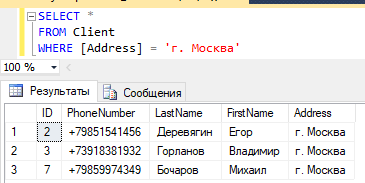


Рисунок 2. Запрос получения клиентов из определённого города

Изображение выглядит как стол

Автоматически созданное описание

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

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4. Запрос вывод только оптовых сделок

Изображение выглядит как стол

Автоматически созданное описаниеРисунок 5. Вывод фамилии и имени в верхнем регистре

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6. Вывод клиентов, купившие определённый товар

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 7. Вывод самого дорогого товара

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Количество сделок за последний месяц

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

# ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №30

**Тема: Работа с базами данных. Изучение механизмов обеспечения целостности данных SQL**

Студент: В.В. Горланов,

Е.С. Деревягин

Группа: ИСИП-619

Преподаватель: И.В. Сибирев

Дата: 21.11.2022

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

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