Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

Отчёт по учебной практике

МДК 01.01 «Разработка программных модулей»

Вариант 25 «Абоненты сотового оператора»

**Выполнил(а):**

Суханова Екатерина

студент(ка) группы

ИСП.20А

2022 г.

Оглавление

[1. Привязка SQL Server к Visual Studio. 2](#_Toc121143159)

[1.1. Создание базы данных по предметной области. 2](#_Toc121143160)

[1.2. Привязка базы данных в проект Visual Studio. 2](#_Toc121143161)

[2. Создание интерфейса. 3](#_Toc121143162)

[2.1. Главное окно. 3](#_Toc121143163)

[2.2. Страницы «Тарифы», «Абоненты», «Абоненты и их тарифы». 4](#_Toc121143164)

[2.3. Окна добавления и удаления. 6](#_Toc121143165)

[3. Логика. 7](#_Toc121143166)

[3.1. Главное окно 7](#_Toc121143167)

[3.2. Страницы «Abonents», «Tarifs», «AbonentsAndTarifs». 8](#_Toc121143168)

[3.3. Excel. 10](#_Toc121143169)

[3.3.1. Добавление ссылки. 10](#_Toc121143170)

[3.3.2. Прописываем логику Excel. 10](#_Toc121143171)

[4. Результат: 11](#_Toc121143173)

1. **Привязка SQL Server к Visual Studio.**
   1. **Создание базы данных по предметной области.**

В SQL Server создаём базу данных из нескольких таблиц, имеющих записи. Между ними обязательно должна быть связь.

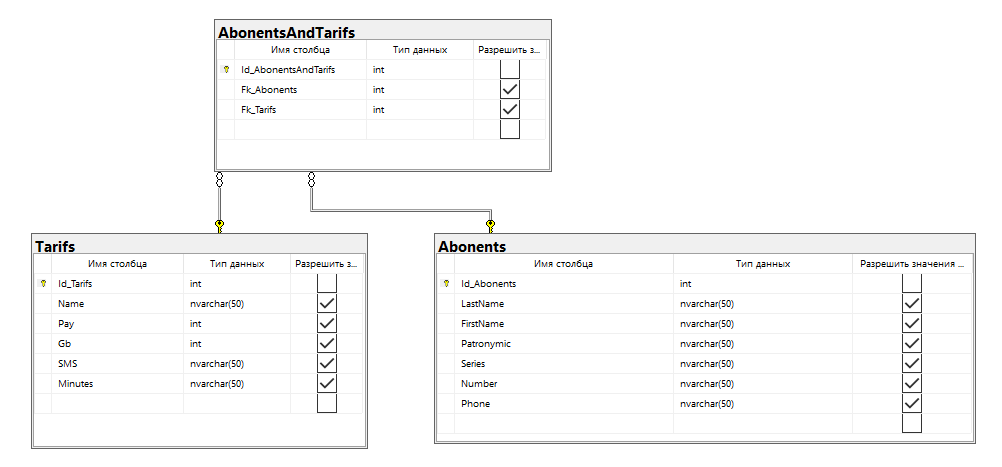


Рис. 1 «Диаграмма базы данных»

Импортируем данные в таблицы из файлов .xlsx, .csv, .txt или добавляем записи самостоятельно.

* 1. **Привязка базы данных в проект Visual Studio.**

Чтобы добавить в проект базу данных нужно нажать ПКМ по названию проекта, затем добавить элемент «Модель ADO.NET EDM». В результате в проекте появится модель с классами, которая соответствует БД.

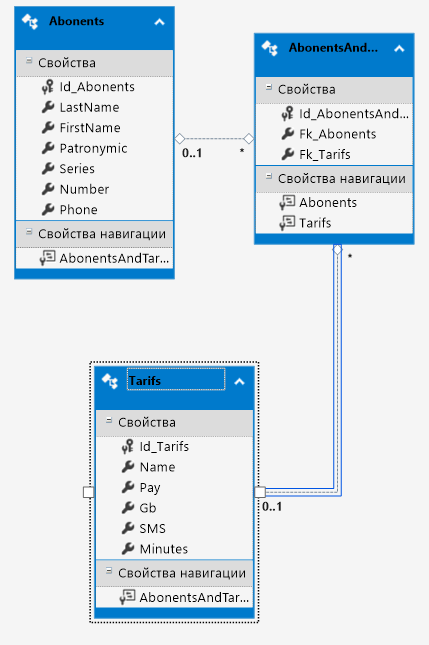


Рис. 2 «Диаграмма базы данных в WPF-проекте»

Пишем код для загрузки страниц в основном окне в MainWindow.xaml.cs:

ClassFrame.frmObj = frmMain;

frmMain.Navigate(new Pages.TarifsPage());

frmMain.Navigate(new Pages.AbonentsPage());

frmMain.Navigate(new Pages.AbonentsAndTarifsPage());

Создаём папку с классом, где будет находится класс с объектом типа Frame для загрузки данных, и подключённая БД из SQL Server и пишем код:

public static Frame frmObj { get; set; }

В MobileOperatorModel.Context.cs прописываем код для работы с полями.

private static MobOperatorEntities \_context;

public MobOperatorEntities()

: base("name=MobOperatorEntities")

{

}

public static MobOperatorEntities GetContext() {

if (\_context == null)

\_context = new MobOperatorEntities();

return \_context;

1. **Создание интерфейса.**
   1. **Главное окно.**

В нем у нас будет название, логотип, место для отображения Frame, кнопки: «Вернуться», «Выйти», «Абоненты», «Абоненты и их тарифы», «Тарифы». Пример создания кнопки, логотипа, таблицы и названия.

<Grid Background="#45AACA">

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Orientation="Horizontal"

HorizontalAlignment="Center" Margin="291,0,291.333,0.333">

<TextBlock Margin="10"

FontSize="20pt"

FontFamily="Boorman Old Style"

VerticalAlignment="Center"

Text="СигналБиг" FontWeight="Light" TextDecorations="Underline">

</TextBlock>

<Image Source="C:\Users\ekate\OneDrive\Рабочий стол\MobileOperator\MobileOperator\Recources\SignalBig.png"

Width="72" Margin="0,0,0,9.333"/>

</StackPanel>

<Button Content="Вернуться"

Height="30"

x:Name="BtnBack"

Width="110"

HorizontalAlignment="Right"

Margin="0,10,163.333,41.333"

Click="BtnBack\_Click"

FontFamily="Bookman Old Style"

Background="#AAF07F" RenderTransformOrigin="-0.56,0.444"/>

<Frame Name="frmMain"

NavigationUIVisibility="Hidden"

ContentRendered="frmMain\_ContentRendered" Foreground="Black" Margin="0,4.667,-0.667,76.667" Grid.RowSpan="2" Grid.Row="1"/>

<StackPanel Grid.Row="2"

Orientation="Horizontal"

HorizontalAlignment="Center">

</StackPanel>

</Grid>

* 1. **Страницы «Тарифы», «Абоненты», «Абоненты и их тарифы».**

Во всех страницах, где будут содержаться таблицы прописываем сверху IsVisibleChanged="Page\_IsVisibleChanged" для отображения записей. Интерфейс у нас будет содержать сортировку, фильтрацию, поиск, редактирование, добавление, удаление, печать в Excel, сброс и подсчёт записей.

<Grid Background="#45AACA">

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center" RenderTransformOrigin="0.572,0.487">

<TextBlock Text="Фильтр по тарифу"

Width="112"

FontFamily="Bookman Old Style"

Margin="5,22,5,19.667"/>

<ComboBox Name="CmbFiltrTarif"

Height="20"

Width="120"

FontFamily="Bookman Old Style"

Margin="5"

SelectionChanged="CmbFiltrTarif\_SelectionChanged"/>

<Button x:Name="BtnResetAll"

Height="25"

Width="70"

FontFamily="Bookman Old Style"

Margin="5"

Content="Сброс"

Click="BtnResetAll\_Click"

Background="#AAF07F"/>

<StackPanel Orientation="Vertical"

VerticalAlignment="Center"

Height="80" Width="202"

Margin="5">

<TextBlock FontFamily="Bookman Old Style"

Text="Сортировка по кол-ву Гб" Margin="0,15,0,0" Width="160"/>

<RadioButton Content="По возрастанию"

FontFamily="Bookman Old Style"

Name="RbUp"

Checked="RbUp\_Checked"/>

<RadioButton Content="По убыванию"

FontFamily="Bookman Old Style"

Name="RbDown"

Checked="RbDown\_Checked"/>

<TextBlock Text="Поиск по фамилии"

FontFamily="Bookman Old Style"

Width="130"

FontSize="13"

Margin="5,20,5,19.667"/>

<TextBox x:Name="TxbSearch"

FontFamily="Bookman Old Style"

Width="100"

Height="18"

TextChanged="TxbSearch\_TextChanged" Margin="0,10,0,19.667" RenderTransformOrigin="0.493,-0.16"/>

</StackPanel>

</StackPanel>

<DataGrid Name="DGridTarifs" Grid.Row="1"

Background="#F89147"

FontFamily="Bookman Old Style"

AutoGenerateColumns="False"

IsReadOnly="True" Width="700">

<DataGrid.Columns>

<DataGridTemplateColumn Width="Auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn

Header="Название" Width="100"

Binding="{Binding Name}"/>

<DataGridTextColumn

Header="Плата" Width="90"

Binding="{Binding Pay}"/>

<DataGridTextColumn

Header="Гб интернета" Width="125"

Binding="{Binding Gb}"/>

<DataGridTextColumn

Header="Наличие смс" Width="\*"

Binding="{Binding SMS}"/>

<DataGridTextColumn

Header="Наличие минут" Width="\*"

Binding="{Binding Minutes}"/>

<DataGridTemplateColumn Width="Auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="BtnEdit"

Click="BtnEdit\_Click"

Content="Редактировать"

Height="30"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="2"

Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Name="BtnNum"

Content="Число записей"

FontFamily="Bookman Old Style"

Margin="10"

Width="100"

Height="30"

Click="BtnNum\_Click" Background="#AAF07F"/>

<TextBlock Text="Записей:"

FontFamily="Bookman Old Style"

VerticalAlignment="Center"/>

<TextBlock x:Name="TxtNum" Width="20"

Margin="2"

FontFamily="Bookman Old Style"

VerticalAlignment="Center"/>

</StackPanel>

</Grid>

* 1. **Окна добавления и удаления.**

Мы используем ComboBox, когда мы выбираем вторичный ключ, в остальных вариантах оставляем TextBox.

<Grid Background="#45AACA">

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<TextBlock HorizontalAlignment="Center"

FontFamily="Bookman Old Style"

VerticalAlignment="Center"

Text="Выберите абонента"></TextBlock>

<ComboBox Name="CmbAbonents"

Grid.Row="0"

FontFamily="Bookman Old Style"

Grid.Column="1"

Height="25"

Width="130"

HorizontalAlignment="Left"

Margin="20"

SelectedValue="{Binding Fk\_Abonents}"/>

<TextBlock HorizontalAlignment="Center"

VerticalAlignment="Center"

FontFamily="Bookman Old Style"

Text="Абонентская плата"

Grid.Row="1"></TextBlock>

<TextBox Grid.Column="1"

Name="TxtPay"

FontFamily="Bookman Old Style"

Text="{Binding Pay}"

Grid.Row="1"

Width="100"

Height="20"

HorizontalAlignment="Left"

Margin="20"

BorderBrush="#FF94AFAD"></TextBox>

<Button Content="Сохранить"

Grid.Row="3"

FontFamily="Bookman Old Style"

Name="BtnSave"

Click="BtnSave\_Click"

Height="30"

Width="118"

HorizontalAlignment="Center" Background="#AAF07F"

Margin="135,9.667,147,80" Grid.ColumnSpan="2"/>

</Grid>

1. **Логика.**
   1. **Главное окно**

Пишем директивы для работы с другими элементами проекта:

using MobileOperator.Classes;

using MobileOperator.Pages;

Это нужно сделать ко всем страницам.

Переход к каждой таблице:

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

{ClassFrame.frmObj.Navigate(new AbonentsAndTarifsPage());}

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

{ClassFrame.frmObj.Navigate(new AbonentsPage());}

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

{ClassFrame.frmObj.Navigate(new TarifsPage());}

Отображение Frame:

private void frmMain\_ContentRendered(object sender, EventArgs e)

{if (frmMain.CanGoBack)

{BtnBack.Visibility = Visibility.Visible; }

else

{BtnBack.Visibility = Visibility.Hidden; }

}

Вернуться на прошлую страницу:

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

{ClassFrame.frmObj.GoBack();}

Закрыть окно:

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

{Close();}

* 1. **Страницы «Abonents», «Tarifs», «AbonentsAndTarifs».**

Пишем контекст в каждом элементе, для работы с таблицей.

private static MobOperatorEntities \_context = new MobOperatorEntities();

Они делаются практически идентично, только меняются названия элементов таблицы.

Код для вывода записей и фильтрации по тарифу:

public TarifsPage()

{InitializeComponent();

DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.ToList();

CmbFiltrTarif.ItemsSource = MobOperatorEntities.GetContext().Tarifs.ToList();

CmbFiltrTarif.SelectedValuePath = "Id\_Tarifs";

CmbFiltrTarif.DisplayMemberPath = "Name";}

Фильтрация по тарифу:

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

{int id = CmbFiltrTarif.SelectedIndex + 1;

DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.Where(x => x.Id\_Tarifs == id).ToList();}

Сброс:

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

{DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.ToList();}

Динамическое отображение данных или изменение данных:

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

{

if (Visibility == Visibility.Visible)

{MobOperatorEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.ToList();

}

}

Сортировка:

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

{

DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.OrderBy(x => x.Gb).ToList();

}

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

{

DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.OrderByDescending(x => x.Gb).ToList();

}

Добавление записи:

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

{

ClassFrame.frmObj.Navigate(new AddEditTarifsPage(null));

}

Удаление записи:

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

{

var tarifForRemoving = DGridTarifs.SelectedItems.Cast<Tarifs>().ToList();

if (MessageBox.Show($"Вы действительно хотите удалить {tarifForRemoving.Count()} данный тариф?",

"Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{try

{MobOperatorEntities.GetContext().Tarifs.RemoveRange(tarifForRemoving);

MobOperatorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

DGridTarifs.ItemsSource = MobOperatorEntities.GetContext().Tarifs.ToList();

catch (Exception ex)

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

}

}

Редактирование записи:

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

{ClassFrame.frmObj.Navigate(new AddEditTarifsPage((sender as Button).DataContext as Tarifs));

Вычисление кол-ва записей:

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

{int countRow = DGridTarifs.Items.Count;

TxtNum.Text = countRow.ToString();}

Поиск:

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

{DGridAbonents.ItemsSource = MobOperatorEntities.GetContext().Abonents.Where(x => x.LastName.ToLower().Contains(TxbSearch.Text.ToLower())).ToList();}

* 1. **Excel.**
     1. **Добавление ссылки.**

Добавляем ссылку на Excel через «Менеджер ссылок».

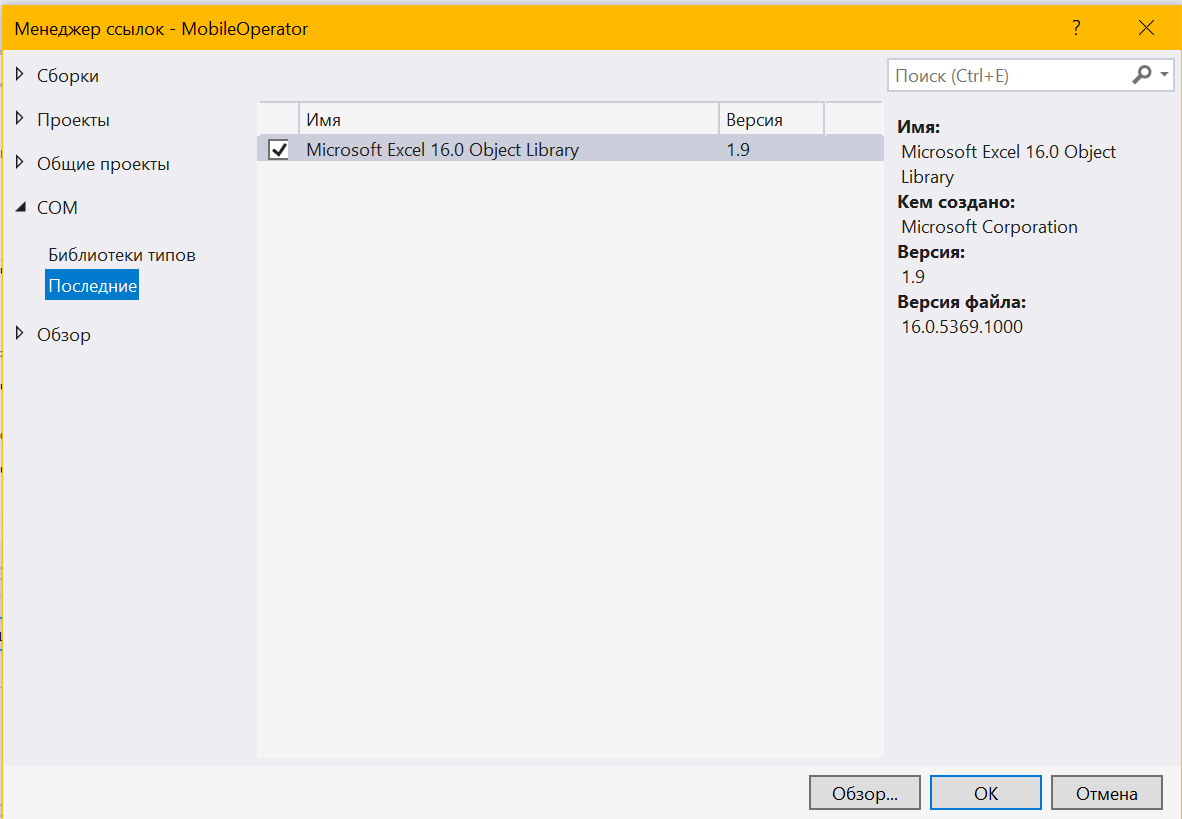


Рис. 3 «Добавление библиотеки Excel»

* + 1. **Прописываем логику Excel.**

Добавляем директиву для работы с Excel.

using Excel = Microsoft.Office.Interop.Excel;

Прописываем код для вывода записей из таблицы в книгу Excel. И одновременно редактируем как документ.

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

{

// объект Excel

var app = new Excel.Application();

// книга

Excel.Workbook wb = app.Workbooks.Add();

// лист

Excel.Worksheet worksheet = app.Worksheets.Item[1];

int indexRows = 3;

Excel.Range TitleRange = worksheet.Range[worksheet.Cells[1][indexRows - 2], worksheet.Cells[7][indexRows - 2]];

TitleRange.Merge();

TitleRange.Value = "Ведомость о всех абонентах и нформация о них";

TitleRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

TitleRange.Font.Italic = true;

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

// ячейки

// 1 - номер столбца

// 2 - номер строки

worksheet.Cells[1][indexRows] = "№";

worksheet.Cells[2][indexRows] = "Фамилия";

worksheet.Cells[3][indexRows] = "Имя";

worksheet.Cells[4][indexRows] = "Отчество";

worksheet.Cells[5][indexRows] = "Серия паспорта";

worksheet.Cells[6][indexRows] = "Номер паспорта";

worksheet.Cells[7][indexRows] = "Телефон";

var printItems = DGridAbonents.Items;

Excel.Range Bold = worksheet.Range[worksheet.Cells[1][indexRows], worksheet.Cells[8][indexRows]];

Bold.Font.Bold = worksheet.Cells[1][indexRows].Font.Bold = true;

Excel.Range Borders = worksheet.Range[worksheet.Cells[1][3], worksheet.Cells[7][14]];

Borders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle =

Borders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle = Excel.XlLineStyle.xlContinuous;

worksheet.Columns.AutoFit();

worksheet.Columns[2].ColumnWidth = 12;

worksheet.Columns[3].ColumnWidth = 12;

worksheet.Columns[4].ColumnWidth = 14;

worksheet.Columns[7].ColumnWidth = 12;

//цикл по данным из таблицы

foreach (Abonents item in printItems)

{

worksheet.Cells[1][indexRows + 1] = indexRows - 2;

worksheet.Cells[2][indexRows + 1] = item.LastName;

worksheet.Cells[3][indexRows + 1] = item.FirstName;

worksheet.Cells[4][indexRows + 1] = item.Patronymic;

worksheet.Cells[5][indexRows + 1] = item.Series;

worksheet.Cells[6][indexRows + 1] = item.Number;

worksheet.Cells[7][indexRows + 1] = item.Phone.ToString();

indexRows++;

}

Excel.Range SignatureRange = worksheet.Range[worksheet.Cells[1][indexRows + 3], worksheet.Cells[3][indexRows + 3]];

SignatureRange.Merge();

SignatureRange.Value = "Подпись:";

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

SignatureRange.Font.Italic = true;

Excel.Range SealRange = worksheet.Range[worksheet.Cells[5][indexRows + 3], worksheet.Cells[7][indexRows + 3]];

SealRange.Merge();

SealRange.Value = "Печать:";

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

SealRange.Font.Italic = true;

worksheet.Range[worksheet.Cells[1][3], worksheet.Cells[7][14]].Font.Name = "Times New Roman";

// показать Excel

app.Visible = true;

}

1. **Результат:**



Рис. 4 «Отображение главного окна с абонентами и их услугами»



Рис. 5 «Отображение главного окна с тарифами»



Рис. 6 «Отображение главного окна с абонентами»



Рис. 7 «Добавление абонента»



Рис. 8 «Добавление абоненту тариф»



Рис. 9 «Добавление тарифа»



Рис. 10 «Печать абонентов и информацию о них в Excel»