AddWindow

Title="Главное окно" Height="295" Width="362" MaxHeight="295" MaxWidth="362" MinHeight="295" MinWidth="362" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button Content="Выход" x:Name="exit" HorizontalAlignment="Left" Height="34" Margin="10,220,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

<Button Content="Просмотр" Name="chek" HorizontalAlignment="Left" Height="43" Margin="62,58,0,0" VerticalAlignment="Top" Width="236" Click="chek\_Click"/>

<Button Content="Добавление" Name="add" HorizontalAlignment="Left" Height="43" Margin="62,106,0,0" VerticalAlignment="Top" Width="236" Click="add\_Click"/>

<Button Name="PlanRass" Content="План рассроч." HorizontalAlignment="Left" Margin="62,154,0,0" VerticalAlignment="Top" Width="116" Height="43" Click="PlanRass\_Click"/>

<Button x:Name="PlanEdin" Content="План един." HorizontalAlignment="Left" Margin="178,154,0,0" VerticalAlignment="Top" Width="120" Height="43" Click="PlanEdin\_Click"/>

</Grid>

public partial class AddWindow : Window

{

public AddWindow()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

}

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

{

this.Close();

}

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

{

new Windows.ProVid().Show();

this.Close();

}

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

{

ChekWindow taskWindow = new ChekWindow();

taskWindow.Show();

this.Close();

}

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

{

new PlanPerMonthe().Show();

this.Close();

}

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

{

new PlanPerMonthEdin().Show();

this.Close();

}

}

}

ChekWindow

Title="Окно добавления" Height="313.25" Width="270.583" MaxHeight="313.25" MaxWidth="270.583" MinHeight="313.25" MinWidth="270.583" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<Label Content="Добавление: " HorizontalAlignment="Left" Margin="84,14,0,0" VerticalAlignment="Top"/>

<Button Content="В рассрочку" x:Name="rass" HorizontalAlignment="Left" Height="42" Margin="10,46,0,0" VerticalAlignment="Top" Width="236" Click="rass\_Click"/>

<Button Content="Единвоременный платёж" x:Name="edin" HorizontalAlignment="Left" Height="42" Margin="10,94,0,0" VerticalAlignment="Top" Width="236" Click="edin\_Click"/>

<Button Content="Вид страхования" x:Name="Vid" HorizontalAlignment="Left" Height="42" Margin="10,142,0,0" VerticalAlignment="Top" Width="236" Click="Vid\_Click"/>

<Button Content="Страхователь" x:Name="Str" HorizontalAlignment="Left" Height="42" Margin="10,190,0,0" VerticalAlignment="Top" Width="236" Click="Str\_Click"/>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Height="34" Margin="10,237,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

</Grid>

public partial class ChekWindow : Window

{

public ChekWindow()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

}

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

{

AddWindow taskWindow = new AddWindow();

taskWindow.Show();

this.Close();

}

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

{

new RessrochWindow().Show();

this.Close();

}

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

{

new EdinWindow().Show();

this.Close();

}

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

{

Windows.VidWindow taskWindow = new Windows.VidWindow();

taskWindow.Show();

this.Close();

}

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

{

new StrahWindow().Show();

this.Close();

}

}

}

EdinWindow

Title="Окно добавления с единовременным платежом" Height="381.5" Width="800" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<Button Content="Назад" x:Name="exit1" HorizontalAlignment="Left" Height="34" Margin="10,302,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

<ComboBox x:Name="cmdkt" HorizontalAlignment="Left" Height="26" Margin="205,16,0,0" VerticalAlignment="Top" Width="518">

</ComboBox>

<Button Content="Добавить" x:Name="addbt1" Height="34" Margin="690,302,0,0" VerticalAlignment="Top" Click="addbt\_Click" HorizontalAlignment="Left" Width="92"/>

<Label Content="Вид страховки\*" HorizontalAlignment="Left" Margin="430,47,0,0" VerticalAlignment="Top"/>

<ComboBox Name="indstr" HorizontalAlignment="Left" Height="26" Margin="544,47,0,0" VerticalAlignment="Top" Width="180"/>

<Label Content="Страхователь\*" HorizontalAlignment="Left" Margin="96,16,0,0" VerticalAlignment="Top"/>

<Label Content="Номер договора\*" HorizontalAlignment="Left" Margin="78,47,0,0" VerticalAlignment="Top"/>

<Label Content="Дата начала договора\*" HorizontalAlignment="Left" Margin="48,78,0,0" VerticalAlignment="Top"/>

<Label Content="Дата окончания договора\*" HorizontalAlignment="Left" Margin="26,109,0,0" VerticalAlignment="Top"/>

<Label Content="Начальная страховая премия\*" HorizontalAlignment="Left" Margin="8,140,0,0" VerticalAlignment="Top"/>

<Label Content="Оплаченая премия\*" HorizontalAlignment="Left" Margin="403,78,0,0" VerticalAlignment="Top"/>

<Label Content="Банк посредник" HorizontalAlignment="Left" Margin="421,109,0,0" VerticalAlignment="Top"/>

<Label Content="Премичание" HorizontalAlignment="Left" Margin="100,211,0,0" VerticalAlignment="Top"/>

<TextBox Name="nomer\_dogovora" HorizontalAlignment="Left" Height="26" Margin="205,47,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="99/99/9999" Name="date\_start" HorizontalAlignment="Left" Height="26" Margin="205,78,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="99/99/9999" Name="date\_finish" HorizontalAlignment="Left" Height="26" Margin="205,109,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox Name="premiaNach" HorizontalAlignment="Left" Height="26" Margin="205,140,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox Name="premiaFin" HorizontalAlignment="Left" Height="26" Margin="544,78,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox Name="bank" HorizontalAlignment="Left" Height="26" Margin="544,109,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox Name="prim" HorizontalAlignment="Left" Height="70" Margin="205,211,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="518"/>

<Label Content="Страховая\*" HorizontalAlignment="Left" Margin="108,171,0,0" VerticalAlignment="Top"/>

<Label Content="Объект страховой" HorizontalAlignment="Left" Margin="409,140,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="objectStr" HorizontalAlignment="Left" Height="26" Margin="544,140,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<ComboBox x:Name="strahovayaCombo" HorizontalAlignment="Left" Margin="205,171,0,0" VerticalAlignment="Top" Width="179" Height="26"/>

</Grid>

</Window>

public partial class EdinWindow : Window

{

public EdinWindow()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

string[] strahovatelCB = db.strahovately

.Select(c => c.FIO\_Namenov)

.ToArray();

for (int i = 0; i < strahovatelCB.Length; i++)

{

cmdkt.Items.Add(strahovatelCB[i]);

}

string[] strahovaya = db.strahovaya

.Select(c => c.naimenovStrahovaya)

.ToArray();

for (int i = 0; i < strahovaya.Length; i++)

{

strahovayaCombo.Items.Add(strahovaya[i]);

}

string[] str = db.strahovki

.Select(c => c.index\_str)

.ToArray();

for (int i = 0; i < str.Length; i++)

{

indstr.Items.Add(str[i]);

}

cmdkt.SelectedIndex = 0;

indstr.SelectedIndex = 0;

strahovayaCombo.SelectedIndex = 0;

}

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

{

new ChekWindow().Show();

this.Close();

}

public static kursacEntities1 db = new kursacEntities1();

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

{

try

{

if (nomer\_dogovora.Text == "" || date\_start.Text == "\_\_.\_\_.\_\_\_\_" || date\_finish.Text == "\_\_.\_\_.\_\_\_\_" || premiaNach.Text == "" || premiaFin.Text == "")

{

System.Windows.MessageBox.Show("Одно или несколько обязательных полей не заполнено!", "Ошибка");

}

else

{

// ID по ФИО

int ID = db.strahovately

.Where(c => c.FIO\_Namenov == cmdkt.Text)

.Select(c => c.idstrahovately)

.FirstOrDefault();

//ID для новой записи

int[] IDoplata = db.edinovr\_oplata

.Select(c => c.idedinovr\_oplata)

.ToArray();

int OpLaTa = Convert.ToInt32(IDoplata.Length) + 1;

// ID по Страховой

int IDstrahovaya = db.strahovaya

.Where(c => c.naimenovStrahovaya == strahovayaCombo.Text)

.Select(c => c.idstrahovaya)

.FirstOrDefault();

var oplata = new edinovr\_oplata() //Добавление в базу

{

idedinovr\_oplata = OpLaTa,

idagent = Convert.ToInt32(1),

index\_str = indstr.Text,

idstrahovately = Convert.ToInt32(ID),

nomer\_dogovora = nomer\_dogovora.Text,

data\_nachala = DateTime.Parse(date\_start.Text),

data\_okonchanya = DateTime.Parse(date\_finish.Text),

nachisl\_str\_perem = Convert.ToDecimal(premiaNach.Text),

oplach\_premya = Convert.ToDecimal(premiaFin.Text),

bank\_posr = bank.Text,

primechanya = prim.Text,

objecti = objectStr.Text,

idstrahovaya = Convert.ToInt32(IDstrahovaya),

};

db.edinovr\_oplata.Add(oplata); // Сохранение запроса

db.SaveChanges(); // Сохранние данных в базе

MessageBox.Show("Добавлено", "Успех");

new AddWindow().Show();

this.Close();

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: "+Convert.ToString(ex),"Ошибка, обратитесь к системному администратору");

}

}

}

}

PlanPerMonthEdin

Title="План на месяц" Height="450" Width="800">

<Grid>

<DataGrid x:Name="grid1" AutoGenerateColumns="False" Margin="10,49,10,0"

CanUserAddRows="False"

CanUserDeleteRows="False"

IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding strahovately.FIO\_Namenov}" Header="ФИО/Наименование" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding index\_str}" Header="Индекс" Width="40"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_nachala, StringFormat=MM/dd/yyyy}" Header="Дата начала" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding objecti}" Header="Объект" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding strahovaya.naimenovStrahovaya}" Header="Страховая" Width="100"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_okonchanya, StringFormat=MM/dd/yyyy}" Header="Дата окончания" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding nomer\_dogovora}" Header="Номер договора" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding oplach\_premya}" Header="Оплаченная премия" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding bank\_posr}" Header="Банк поср." Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding primechanya}" Header="Примечание" Width="SizeToCells"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Margin="10,10,0,0" Width="95" Click="exit\_Click" VerticalAlignment="Top" Height="34"/>

</Grid>

</Window>

public partial class PlanPerMonthEdin : Window

{

kursacEntities1 db = new kursacEntities1();

public PlanPerMonthEdin()

{

try

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

List<edinovr\_oplata> listRsas = db.edinovr\_oplata.ToList();

DateTime DateNow = DateTime.Now;

DateTime DateStart = DateTime.Parse("01/" + DateNow.Month + "/" + DateNow.Year);

DateTime DateFin = DateTime.Parse("30/" + DateNow.Month + "/" + DateNow.Year);

if (DateNow.Day >= 20)

{

DateFin = DateTime.Parse("30/" + (DateNow.Month + 1) + "/" + DateNow.Year);

}

List<edinovr\_oplata> dates = db.edinovr\_oplata.ToList();

dates = db.edinovr\_oplata.Where(c => c.data\_okonchanya >= DateStart && c.data\_okonchanya <= DateFin).ToList();

grid1.ItemsSource = dates.ToList();

}

catch(Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

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

{

new AddWindow().Show();

this.Close();

}

}

}

PlanPerMonthRass

Title="План на месяц" Height="450" Width="800" Icon="/КурсовойПроект;component/AppData/BestIco\_1.ico">

<Grid>

<DataGrid x:Name="grid1" AutoGenerateColumns="False" Margin="10,49,10,0"

CanUserAddRows="False"

CanUserDeleteRows="False"

CanUserResizeRows="False"

>

<DataGrid.Columns>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding strahovately.FIO\_Namenov}" Header="ФИО/Наименование" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding index\_str}" Header="Индекс" Width="60"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_nachala, StringFormat=MM/dd/yyyy}" Header="Дата начала" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_okonchanya, StringFormat=MM/dd/yyyy}" Header="Дата окончания" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding nomer\_dogovora}" Header="Номер договора" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding summa\_platezh}" Header="Общая сумма" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding strahovaya.naimenovStrahovaya}" Header="Страховая" Width="90"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding objecti}" Header="Объект" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding bank\_posr}" Header="Банк поср." Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding primechanya}" Header="Примечание" Width="SizeToCells"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top" Width="92" Height="34" Click="exit\_Click"/>

</Grid>

</Window>

public partial class PlanPerMonthe : Window

{

kursacEntities1 db = new kursacEntities1();

public PlanPerMonthe()

{

try

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

List<rassrochka> listRsas = db.rassrochka.ToList();

DateTime dt = DateTime.Now;

DateTime x1 = DateTime.Parse("01/" + dt.Month + "/" + dt.Year);

DateTime x2 = DateTime.Parse("29/" + dt.Month + "/" + dt.Year);

if (dt.Day >= 20)

{

x2 = DateTime.Parse("29/" + (dt.Month + 1) + "/" + dt.Year);

}

List<rassrochka> dates = db.rassrochka.ToList();

dates = db.rassrochka.Where(c => c.data\_okonchanya >= x1 && c.data\_okonchanya <= x2).ToList();

grid1.ItemsSource = dates.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

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

{

new AddWindow().Show();

this.Close();

}

}

}

ProsmotrEdin

Title="Окно просмотра с единовременной оплатой" Height="600" Width="917" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<DataGrid Name="grid1" AutoGenerateColumns="False" Margin="10,67,0,0"

CanUserAddRows="False"

CanUserDeleteRows="True"

HorizontalAlignment="Left" Width="889"

>

<DataGrid.Columns>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding strahovately.FIO\_Namenov}" Header="ФИО/Наименование" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding index\_str}" Header="Индекс" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding data\_nachala, StringFormat=MM/dd/yyyy}" Header="Дата начала" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding strahovaya.naimenovStrahovaya}" Header="Страховая" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding objecti}" Header="Объект" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding data\_okonchanya, StringFormat=MM/dd/yyyy}" Header="Дата окончания" Width="60"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding nomer\_dogovora}" Header="Номер договора" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding nachisl\_str\_perem}" Header="Начальная премия" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding oplach\_premya}" Header="Оплаченная премия" Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding bank\_posr}" Header="Банк поср." Width="80"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding primechanya}" Header="Примечание" Width="80"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Margin="398,28,0,0" Width="92" Click="exit\_Click" VerticalAlignment="Top" Height="34"/>

<Button Content="Рассрочка" x:Name="rass" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="10,28,0,0" Click="rass\_Click\_1"/>

<Button Content="Страхователи" x:Name="strah" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="107,28,0,0" Click="strah\_Click"/>

<Button Content="Страховки" x:Name="Страховки" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="204,28,0,0" Click="Страховки\_Click"/>

<Button Content="Изменить" x:Name="change" HorizontalAlignment="Left" Margin="301,28,0,0" VerticalAlignment="Top" Width="92" Height="34" Click="change\_Click"/>

<Label Content="Страховая" HorizontalAlignment="Left" Margin="501,8,0,0" VerticalAlignment="Top"/>

<ComboBox Name="comboStrahov" HorizontalAlignment="Left" Margin="500,34,0,0" VerticalAlignment="Top" Width="120" SelectionChanged="comboStrahov\_SelectionChanged" Height="24">

<ComboBoxItem>

Все

</ComboBoxItem>

</ComboBox>

<ComboBox x:Name="ComboMonth" HorizontalAlignment="Left" Margin="635,34,0,0" VerticalAlignment="Top" Width="120" SelectionChanged="ComboMonth\_SelectionChanged" Height="24">

<ComboBoxItem Content="Январь"/>

<ComboBoxItem Content="Февраль"/>

<ComboBoxItem Content="Март"/>

<ComboBoxItem Content="Апрель"/>

<ComboBoxItem Content="Май"/>

<ComboBoxItem Content="Июнь"/>

<ComboBoxItem Content="Июль"/>

<ComboBoxItem Content="Август"/>

<ComboBoxItem Content="Сентябрь"/>

<ComboBoxItem Content="Октябрь"/>

<ComboBoxItem Content="Ноябрь"/>

<ComboBoxItem Content="Декабрь"/>

<ComboBoxItem Content="Все"/>

</ComboBox>

<Label Name="labelMonthCheck" Content="Месяц" HorizontalAlignment="Left" Margin="635,10,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

public partial class ProsmotrEdin : Window

{

kursacEntities1 db = new kursacEntities1();

public ProsmotrEdin()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

grid1.ItemsSource = db.edinovr\_oplata.ToList();

// Забиваю комбо со страховой

string[] str = db.strahovaya

.Select(c => c.naimenovStrahovaya)

.ToArray();

for (int i = 0; i < str.Length; i++)

{

comboStrahov.Items.Add(str[i]);

}

comboStrahov.SelectedIndex = 0;

ComboMonth.SelectedIndex = 12;

}

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

{

new ProVid().Show();

this.Close();

}

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

{

new ProsmotrStrahovatel().Show();

this.Close();

}

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

{

new ProsmotrRassroch().Show();

this.Close();

}

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

{

new AddWindow().Show();

this.Close();

}

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

{

db.SaveChanges();

MessageBox.Show("Изменение выполнено успешно!", "Успех");

}

private void SortClass(string SortStrah = "Все", string SortMonth = "12")

{

try

{

List<edinovr\_oplata> SotrtList = db.edinovr\_oplata.ToList();

switch (SortStrah)

{

case "РЕСО":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(1)).ToList();

break;

case "РЕНЕССАНС":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(2)).ToList();

break;

case "РОСГОССТРАХ":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(3)).ToList();

break;

case "ГАЙДЕ":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(4)).ToList();

break;

case "ВСК":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(5)).ToList();

break;

case "Другое":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(6)).ToList();

break;

default:

break;

}

DateTime dt = DateTime.Now;

DateTime x1;

DateTime x2;

switch (SortMonth)

{

case "0":

x1 = DateTime.Parse("01/01/" + dt.Year);

x2 = DateTime.Parse("30/01/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "1":

x1 = DateTime.Parse("01/02/" + dt.Year);

x2 = DateTime.Parse("28/02/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "2":

x1 = DateTime.Parse("01/03/" + dt.Year);

x2 = DateTime.Parse("30/03/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "3":

x1 = DateTime.Parse("01/04/" + dt.Year);

x2 = DateTime.Parse("30/04/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "4":

x1 = DateTime.Parse("01/05/" + dt.Year);

x2 = DateTime.Parse("30/05/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "5":

x1 = DateTime.Parse("01/06/" + dt.Year);

x2 = DateTime.Parse("30/06/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "6":

x1 = DateTime.Parse("01/07/" + dt.Year);

x2 = DateTime.Parse("30/07/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "7":

x1 = DateTime.Parse("01/08/" + dt.Year);

x2 = DateTime.Parse("30/08/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "8":

x1 = DateTime.Parse("01/09/" + dt.Year);

x2 = DateTime.Parse("30/09/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "9":

x1 = DateTime.Parse("01/10/" + dt.Year);

x2 = DateTime.Parse("30/10/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "10":

x1 = DateTime.Parse("01/11/" + dt.Year);

x2 = DateTime.Parse("30/11/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "11":

x1 = DateTime.Parse("01/12/" + dt.Year);

x2 = DateTime.Parse("30/12/" + dt.Year);

SotrtList = db.edinovr\_oplata.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

default:

break;

}

grid1.ItemsSource = SotrtList;

}

catch (Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

private void comboStrahov\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

SortClass(comboStrahov.SelectedItem.ToString(),ComboMonth.SelectedIndex.ToString());

}

private void ComboMonth\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

SortClass(comboStrahov.SelectedItem.ToString(), ComboMonth.SelectedIndex.ToString());

}

}

}

ProsmotrRassroch

Title="Окно просмотра с оплатой в рассрочку" Height="600" Width="917" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<DataGrid x:Name="grid1" AutoGenerateColumns="False" Margin="10,68,10,0"

CanUserAddRows="False"

CanUserDeleteRows="True">

<DataGrid.Columns>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding strahovately.FIO\_Namenov}" Header="ФИО/Наименование" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding index\_str}" Header="Индекс" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_nachala, StringFormat=MM/dd/yyyy}" Header="Дата начала" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_okonchanya, StringFormat=MM/dd/yyyy}" Header="Дата окончания" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding nomer\_dogovora}" Header="Номер договора" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding strahovaya.naimenovStrahovaya}" Header="Страховая" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding objecti}" Header="Объект" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding nachisl\_str\_perem}" Header="Начальная премия" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding oplach\_premya}" Header="Оплаченная премия" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding kol\_platezh}" Header="Кол-во платежей" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding data\_2pl, StringFormat=MM/dd/yyyy}" Header="Дата 2ого платежа" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding data\_3pl, StringFormat=MM/dd/yyyy}" Header="Дата 3ого платежа" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding data\_4pl, StringFormat=MM/dd/yyyy}" Header="Дата 4ого платежа" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding summa\_2pl}" Header="Сумма 2ого пл" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding summa\_3pl}" Header="Сумма 3ого пл" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding summa\_4pl}" Header="Сумма 4ого пл" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding summa\_platezh}" Header="Общая сумма" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding bank\_posr}" Header="Банк поср." Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding primechanya}" Header="Примечание" Width="SizeToCells"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Margin="404,29,0,0" Width="98" Click="exit\_Click" VerticalAlignment="Top" Height="34"/>

<Button Content="Единовременная" x:Name="edin" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="98" Margin="10,29,0,0" Click="edin\_Click"/>

<Button Content="Страхователи" x:Name="strah" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="112,29,0,0" Click="strah\_Click"/>

<Button Content="Страховки" x:Name="Страховки" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="210,29,0,0" Click="Страховки\_Click"/>

<Button Content="Изменить" x:Name="change" HorizontalAlignment="Left" Margin="306,29,0,0" VerticalAlignment="Top" Width="92" Height="34" Click="change\_Click"/>

<Label Content="Страховая" HorizontalAlignment="Left" Margin="512,10,0,0" VerticalAlignment="Top"/>

<ComboBox x:Name="comboStrahov" HorizontalAlignment="Left" Margin="512,41,0,0" VerticalAlignment="Top" Width="120" SelectionChanged="comboStrahov\_SelectionChanged">

<ComboBoxItem Content="Все"/>

</ComboBox>

<ComboBox x:Name="ComboMonth" HorizontalAlignment="Left" Margin="642,41,0,0" VerticalAlignment="Top" Width="120" SelectionChanged="ComboMonth\_SelectionChanged">

<ComboBoxItem Content="Январь"/>

<ComboBoxItem Content="Февраль"/>

<ComboBoxItem Content="Март"/>

<ComboBoxItem Content="Апрель"/>

<ComboBoxItem Content="Май"/>

<ComboBoxItem Content="Июнь"/>

<ComboBoxItem Content="Июль"/>

<ComboBoxItem Content="Август"/>

<ComboBoxItem Content="Сентябрь"/>

<ComboBoxItem Content="Октябрь"/>

<ComboBoxItem Content="Ноябрь"/>

<ComboBoxItem Content="Декабрь"/>

<ComboBoxItem Content="Все"/>

</ComboBox>

<Label x:Name="labelMonthCheck" Content="Месяц" HorizontalAlignment="Left" Margin="642,10,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

public partial class ProsmotrRassroch : Window

{

kursacEntities1 db = new kursacEntities1();

public ProsmotrRassroch()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

grid1.ItemsSource = db.rassrochka.ToList();

// Забиваю комбо со страховой

string[] str = db.strahovaya

.Select(c => c.naimenovStrahovaya)

.ToArray();

for (int i = 0; i < str.Length; i++)

{

comboStrahov.Items.Add(str[i]);

}

comboStrahov.SelectedIndex = 0;

DateTime dk = DateTime.Now;

ComboMonth.SelectedIndex = 12;

}

private void SortClass(string SortStrah = "Все", string SortMonth = "12")

{

try

{

List<rassrochka> SotrtList = db.rassrochka.ToList();

switch (SortStrah)

{

case "РЕСО":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(1)).ToList();

break;

case "РЕНЕССАНС":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(2)).ToList();

break;

case "РОСГОССТРАХ":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(3)).ToList();

break;

case "ГАЙДЕ":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(4)).ToList();

break;

case "ВСК":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(5)).ToList();

break;

case "Другое":

SotrtList = SotrtList.Where(c => c.idstrahovaya.Equals(6)).ToList();

break;

default:

break;

}

DateTime dt = DateTime.Now;

DateTime x1;

DateTime x2;

switch (SortMonth)

{

case "0":

x1 = DateTime.Parse("01/01/" + dt.Year);

x2 = DateTime.Parse("30/01/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "1":

x1 = DateTime.Parse("01/02/" + dt.Year);

x2 = DateTime.Parse("28/02/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "2":

x1 = DateTime.Parse("01/03/" + dt.Year);

x2 = DateTime.Parse("30/03/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "3":

x1 = DateTime.Parse("01/04/" + dt.Year);

x2 = DateTime.Parse("30/04/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "4":

x1 = DateTime.Parse("01/05/" + dt.Year);

x2 = DateTime.Parse("30/05/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "5":

x1 = DateTime.Parse("01/06/" + dt.Year);

x2 = DateTime.Parse("30/06/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "6":

x1 = DateTime.Parse("01/07/" + dt.Year);

x2 = DateTime.Parse("30/07/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "7":

x1 = DateTime.Parse("01/08/" + dt.Year);

x2 = DateTime.Parse("30/08/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "8":

x1 = DateTime.Parse("01/09/" + dt.Year);

x2 = DateTime.Parse("30/09/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "9":

x1 = DateTime.Parse("01/10/" + dt.Year);

x2 = DateTime.Parse("30/10/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "10":

x1 = DateTime.Parse("01/11/" + dt.Year);

x2 = DateTime.Parse("30/11/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

case "11":

x1 = DateTime.Parse("01/12/" + dt.Year);

x2 = DateTime.Parse("30/12/" + dt.Year);

SotrtList = db.rassrochka.Where(c => c.data\_nachala >= x1 && c.data\_nachala <= x2).ToList();

break;

default:

break;

}

grid1.ItemsSource = SotrtList;

}

catch (Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

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

{

AddWindow taskWindow = new AddWindow();

taskWindow.Show();

this.Close();

}

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

{

new ProVid().Show();

this.Close();

}

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

{

new ProsmotrStrahovatel().Show();

this.Close();

}

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

{

new ProsmotrEdin().Show();

this.Close();

}

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

{

db.SaveChanges();

MessageBox.Show("Изменение выполнено успешно!","Успех");

}

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

{

SortClass(comboStrahov.SelectedItem.ToString(), ComboMonth.SelectedIndex.ToString());

}

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

{

SortClass(comboStrahov.SelectedItem.ToString(), ComboMonth.SelectedIndex.ToString());

}

}

}

ProsmotrStrahovatel

Title="Окно просмотра страхователей" Height="600" Width="800" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<DataGrid x:Name="grid1" AutoGenerateColumns="False" Margin="10,49,10,0"

CanUserAddRows="False"

CanUserDeleteRows="False"

CanUserResizeRows="False"

>

<DataGrid.Columns>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding FIO\_Namenov}" Header="ФИО/Наименование" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding data\_rozden, StringFormat=MM/dd/yyyy}" Header="Дата рождения" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding address}" Header="Адрес" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding rab\_telefon}" Header="Рабочий телефон" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding mob\_telefon\_1}" Header="Мобильный 1" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding mob\_telefon\_2}" Header="Мобильный 2" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding dom\_telefon}" Header="Домашний" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="True" Binding="{Binding kategorya}" Header="Категория" Width="SizeToCells"/>

<DataGridTextColumn IsReadOnly="False" Binding="{Binding primechanya}" Header="Примечание" Width="100"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Margin="410,10,0,0" VerticalAlignment="Top" Width="94" Click="exit\_Click" Height="34"/>

<Button Content="Рассрочка" x:Name="rass" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="10,10,0,0" Click="rass\_Click\_1"/>

<Button Content="Единовременная" x:Name="edin" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="98" Margin="108,10,0,0" Click="edin\_Click"/>

<Button Content="Страховки" x:Name="Страховки" HorizontalAlignment="Left" Height="34" VerticalAlignment="Top" Width="92" Margin="210,10,0,0" Click="Страховки\_Click"/>

<Button Content="Изменить" x:Name="change" HorizontalAlignment="Left" Margin="307,10,0,0" VerticalAlignment="Top" Width="98" Height="34" Click="change\_Click"/>

</Grid>

</Window>

kursacEntities1 db = new kursacEntities1();

public ProsmotrStrahovatel()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

grid1.ItemsSource = db.strahovately.ToList();

}

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

{

AddWindow taskWindow = new AddWindow();

taskWindow.Show();

this.Close();

}

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

{

new ProVid().Show();

this.Close();

}

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

{

new ProsmotrEdin().Show();

this.Close();

}

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

{

new ProsmotrRassroch().Show();

this.Close();

}

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

{

db.SaveChanges();

MessageBox.Show("Изменение выполнено успешно!", "Успех");

}

}

}

ProVidstrah

Title="Окно просмотра видов страховок" Height="600" Width="515" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<DataGrid x:Name="grid1" AutoGenerateColumns="False" Margin="10,97,10,0"

CanUserAddRows="False"

CanUserDeleteRows="True">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding index\_str}" Header="Индекс страхователя" Width="SizeToCells" IsReadOnly="True"/>

<DataGridTextColumn Binding="{Binding naimenov\_str}" Header="Наименование" Width="SizeToCells" IsReadOnly="False"/>

<DataGridTextColumn Binding="{Binding kategoria\_str}" Header="Категория" Width="80" IsReadOnly="False"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" x:Name="exit" Margin="10,50,0,0" VerticalAlignment="Top" Height="34" Click="exit\_Click" HorizontalAlignment="Left" Width="92"/>

<Button Content="Рассрочка" x:Name="rass" Height="34" VerticalAlignment="Top" Margin="10,10,0,0" Click="rass\_Click\_1" HorizontalAlignment="Left" Width="92"/>

<Button Content="Единовременная" x:Name="edin" Height="34" VerticalAlignment="Top" Margin="107,10,0,0" Click="edin\_Click" HorizontalAlignment="Left" Width="97"/>

<Button Content="Страхователи" x:Name="strah" Height="34" VerticalAlignment="Top" Margin="209,10,0,0" Click="strah\_Click" HorizontalAlignment="Left" Width="92"/>

<Button Content="Изменить" x:Name="change" Margin="107,50,0,0" VerticalAlignment="Top" Height="34" Click="change\_Click" HorizontalAlignment="Left" Width="97"/>

<Label Content="Категория: " HorizontalAlignment="Left" VerticalAlignment="Top" Margin="234,58,0,0"/>

<ComboBox Name="KategoriCombo" VerticalAlignment="Top" Margin="311,62,38,0" SelectionChanged="KategoriCombo\_SelectionChanged">

<ComboBoxItem>

Все

</ComboBoxItem>

<ComboBoxItem>

Юр. Лица

</ComboBoxItem>

<ComboBoxItem>

Здоровье

</ComboBoxItem>

<ComboBoxItem>

Имущество

</ComboBoxItem>

<ComboBoxItem>

Автомобиль

</ComboBoxItem>

<ComboBoxItem>

Путешествия

</ComboBoxItem>

</ComboBox>

</Grid>

</Window>

public partial class ProVid : Window

{

public ProVid()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

grid1.ItemsSource = db.strahovki.ToList();

KategoriCombo.SelectedIndex = 0;

}

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

{

new AddWindow().Show();

this.Close();

}

public static kursacEntities1 db = new kursacEntities1();

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

{

}

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

{

new ProsmotrEdin().Show();

this.Close();

}

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

{

new ProsmotrStrahovatel().Show();

this.Close();

}

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

{

new ProsmotrRassroch().Show();

this.Close();

}

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

{

try

{

grid1.UpdateLayout(); // Фиксируем изменения внесенные в DataTable

db.SaveChanges();

db.SaveChanges();

MessageBox.Show("Изменение выполнено успешно!", "Успех");

}

catch(Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

private void GetSort(string SortStrah = "0")

{

try

{

List<strahovki> SotrtList = db.strahovki.ToList();

switch (SortStrah)

{

case "1":

SotrtList = db.strahovki.Where(c => c.kategoria\_str.Equals("Юр. Лица")).ToList();

break;

case "2":

SotrtList = db.strahovki.Where(c => c.kategoria\_str.Equals("Здоровье")).ToList();

break;

case "3":

SotrtList = db.strahovki.Where(c => c.kategoria\_str.Equals("Имущество")).ToList();

break;

case "4":

SotrtList = db.strahovki.Where(c => c.kategoria\_str.Equals("Автомобиль")).ToList();

break;

case "5":

SotrtList = db.strahovki.Where(c => c.kategoria\_str.Equals("Путешествия")).ToList();

break;

default:

break;

}

grid1.ItemsSource = SotrtList;

}

catch (Exception ex)

{

System.Windows.MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

private void KategoriCombo\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

GetSort(KategoriCombo.SelectedIndex.ToString());

}

}

}

RessrochWindow

Title="Окно добавления с платежом в рассрочку" Height="500" Width="800" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<ComboBox x:Name="cmdkt" HorizontalAlignment="Left" Height="26" Margin="194,58,0,0" VerticalAlignment="Top" Width="519"/>

<Label Content="Вид страховки\*" HorizontalAlignment="Left" Margin="94,276,0,0" VerticalAlignment="Top"/>

<ComboBox x:Name="indstr" HorizontalAlignment="Left" Height="26" Margin="194,276,0,0" VerticalAlignment="Top" Width="179"/>

<Label Content="Страхователь\*" HorizontalAlignment="Left" Margin="99,58,0,0" VerticalAlignment="Top"/>

<Label Content="Номер договора\*" HorizontalAlignment="Left" Margin="81,90,0,0" VerticalAlignment="Top"/>

<Label Content="Дата начала договора\*" HorizontalAlignment="Left" Margin="51,120,0,0" VerticalAlignment="Top"/>

<Label Content="Дата окончания договора\*" HorizontalAlignment="Left" Margin="29,152,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="nomer\_dogovora" HorizontalAlignment="Left" Height="26" Margin="194,90,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="99/99/9999" x:Name="date\_start" HorizontalAlignment="Left" Height="26" Margin="194,120,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="99/99/9999" x:Name="date\_finish" HorizontalAlignment="Left" Height="26" Margin="194,152,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Button Content="Назад" x:Name="exit1" HorizontalAlignment="Left" Height="34" Margin="10,425,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

<Button Content="Добавить" x:Name="addbt1" HorizontalAlignment="Left" Height="34" Margin="690,425,0,0" VerticalAlignment="Top" Width="92" Click="addbt\_Click"/>

<Label Content="Начисленная страховая премия\*" HorizontalAlignment="Left" Margin="0,182,0,0" VerticalAlignment="Top"/>

<Label Content="Оплаченая премия\*" HorizontalAlignment="Left" Margin="67,214,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="premiaNach" HorizontalAlignment="Left" Height="26" Margin="194,182,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="premiaFin" HorizontalAlignment="Left" Height="26" Margin="194,214,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Label Content="Количество платежей\*" HorizontalAlignment="Left" Margin="46,244,0,0" VerticalAlignment="Top"/>

<Label Content="Дата ожид. платежа 2\*" HorizontalAlignment="Left" Margin="399,88,0,0" VerticalAlignment="Top"/>

<Label Content="Дата ожид. платежа 3" HorizontalAlignment="Left" Margin="397,118,0,0" VerticalAlignment="Top"/>

<Label Content="Дата ожид. платежа 4" HorizontalAlignment="Left" Margin="397,150,0,0" VerticalAlignment="Top"/>

<Label Content="Сумма 2 платежа\*" HorizontalAlignment="Left" Margin="422,180,0,0" VerticalAlignment="Top"/>

<Label Content="Сумма 3 платежа" HorizontalAlignment="Left" Margin="424,212,0,0" VerticalAlignment="Top"/>

<Label Content="Сумма 4 платежа" HorizontalAlignment="Left" Margin="424,242,0,0" VerticalAlignment="Top"/>

<Label Content="Банк посредник" HorizontalAlignment="Left" Margin="429,274,0,0" VerticalAlignment="Top"/>

<Label Content="Примечания" HorizontalAlignment="Left" Margin="108,338,0,0" VerticalAlignment="Top"/>

<Obselete:MaskedTextBox Mask="99/99/9999" x:Name="date2" HorizontalAlignment="Left" Height="26" Margin="534,88,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="99/99/9999" x:Name="date3" HorizontalAlignment="Left" Height="26" Margin="534,118,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="00/00/0000" x:Name="date4" HorizontalAlignment="Left" Height="26" Margin="534,150,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="plat2" HorizontalAlignment="Left" Height="26" Margin="534,180,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="plat3" HorizontalAlignment="Left" Height="26" Margin="534,212,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="plat4" HorizontalAlignment="Left" Height="26" Margin="534,242,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="bank" HorizontalAlignment="Left" Height="26" Margin="534,274,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="primech" HorizontalAlignment="Left" Height="52" Margin="194,338,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="519"/>

<ComboBox x:Name="kol" HorizontalAlignment="Left" Height="26" Margin="194,244,0,0" VerticalAlignment="Top" Text="" Width="179" SelectionChanged="kol\_SelectionChanged">

<ComboBoxItem Content="2"/>

<ComboBoxItem Content="3"/>

<ComboBoxItem Content="4"/>

</ComboBox>

<Label Content="Страховая" HorizontalAlignment="Left" Margin="121,307,0,0" VerticalAlignment="Top"/>

<Label Content="Объект страхования" HorizontalAlignment="Left" Margin="407,305,0,0" VerticalAlignment="Top"/>

<TextBox Name="objectStr" HorizontalAlignment="Left" Height="26" Margin="534,305,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<ComboBox Name="strahovayaCombo" HorizontalAlignment="Left" Margin="194,307,0,0" VerticalAlignment="Top" Width="179" Height="26">

</ComboBox>

</Grid>

</Window>

public partial class RessrochWindow : Window

{

public static kursacEntities1 db = new kursacEntities1();

public RessrochWindow()

{

try

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

string[] strahovatelCB = db.strahovately

.Select(c => c.FIO\_Namenov)

.ToArray();

for (int i = 0; i < strahovatelCB.Length; i++)

{

cmdkt.Items.Add(strahovatelCB[i]);

}

string[] str = db.strahovki

.Select(c => c.index\_str)

.ToArray();

for (int i = 0; i < str.Length; i++)

{

indstr.Items.Add(str[i]);

}

// Наименование

string[] strahovaya = db.strahovaya

.Select(c => c.naimenovStrahovaya)

.ToArray();

for (int i = 0; i < strahovaya.Length; i++)

{

strahovayaCombo.Items.Add(strahovaya[i]);

}

cmdkt.SelectedIndex = 0;

indstr.SelectedIndex = 0;

strahovayaCombo.SelectedIndex = 0;

kol.SelectedIndex = 0;

date3.IsReadOnly = true;

date4.IsReadOnly = true;

}

catch (Exception ex)

{

System.Windows.MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

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

{

try

{

var ras = new rassrochka();

if (nomer\_dogovora.Text == "" || date\_start.Text == "\_\_.\_\_.\_\_\_\_" || date\_finish.Text == "\_\_.\_\_.\_\_\_\_" || premiaNach.Text == "" || premiaFin.Text == "" || date2.Text == "\_\_.\_\_.\_\_\_\_" || plat2.Text == "")

{

System.Windows.MessageBox.Show("Одно или несколько обязательных полей не заполнено!", "Ошибка");

}

else

{

// ID по ФИО

int ID = db.strahovately

.Where(c => c.FIO\_Namenov == cmdkt.Text)

.Select(c => c.idstrahovately)

.FirstOrDefault();

// ID по Страховой

int IDstrahovaya = db.strahovaya

.Where(c => c.naimenovStrahovaya == strahovayaCombo.Text)

.Select(c => c.idstrahovaya)

.FirstOrDefault();

int per = 0;

if (date3.Text == "\_\_.\_\_.\_\_\_\_" && date4.Text != "\_\_.\_\_.\_\_\_\_")

{

System.Windows.MessageBox.Show("Поле 4 платежа не может быть заполнено без поля третьего!");

}

else

{

if (date3.Text != "\_\_.\_\_.\_\_\_\_")

{

per = 1;

var rassroch1 = new rassrochka()

{

idagent = 1,

index\_str = indstr.Text,

idstrahovately = Convert.ToInt32(ID),

nomer\_dogovora = nomer\_dogovora.Text,

data\_nachala = DateTime.Parse(date\_start.Text),

data\_okonchanya = DateTime.Parse(date\_finish.Text),

nachisl\_str\_premya = Convert.ToDecimal(premiaNach.Text),

oplach\_premya = Convert.ToDecimal(premiaFin.Text),

kol\_platezh = Convert.ToInt32(kol.Text),

data\_2pl = DateTime.Parse(date2.Text),

summa\_platezh = Convert.ToDecimal(Convert.ToInt32(plat2.Text) + Convert.ToInt32(plat3.Text)),

data\_3pl = DateTime.Parse(date3.Text),

bank\_posr = bank.Text,

primechanya = primech.Text,

objecti = objectStr.Text,

idstrahovaya = Convert.ToInt32(IDstrahovaya)

};

if (date4.Text == "\_\_.\_\_.\_\_\_\_")

{

db.rassrochka.Add(rassroch1);

db.SaveChanges();

}

if (date4.Text != "\_\_.\_\_.\_\_\_\_")

{

per = 2;

var rassroch3 = new rassrochka()

{

idagent = 1,

index\_str = indstr.Text,

idstrahovately = Convert.ToInt32(ID),

nomer\_dogovora = nomer\_dogovora.Text,

data\_nachala = DateTime.Parse(date\_start.Text),

data\_okonchanya = DateTime.Parse(date\_finish.Text),

nachisl\_str\_premya = Convert.ToDecimal(premiaNach.Text),

oplach\_premya = Convert.ToDecimal(premiaFin.Text),

kol\_platezh = Convert.ToInt32(kol.Text),

data\_2pl = DateTime.Parse(date2.Text),

summa\_platezh = Convert.ToDecimal(Convert.ToInt32(plat2.Text) + Convert.ToInt32(plat3.Text) + Convert.ToInt32(plat4.Text)),

data\_3pl = DateTime.Parse(date3.Text),

data\_4pl = DateTime.Parse(date4.Text),

summa\_4pl = Convert.ToDecimal(plat4.Text),

summa\_3pl = Convert.ToDecimal(plat3.Text),

summa\_2pl = Convert.ToDecimal(plat2.Text),

objecti = objectStr.Text,

idstrahovaya = Convert.ToInt32(IDstrahovaya),

bank\_posr = bank.Text,

primechanya = primech.Text

};

db.rassrochka.Add(rassroch3);

db.SaveChanges();

}

}

else

{

var rassroch = new rassrochka()

{

idagent = 1,

index\_str = indstr.Text,

idstrahovately = Convert.ToInt32(ID),

nomer\_dogovora = nomer\_dogovora.Text,

data\_nachala = DateTime.Parse(date\_start.Text),

data\_okonchanya = DateTime.Parse(date\_finish.Text),

nachisl\_str\_premya = Convert.ToDecimal(premiaNach.Text),

oplach\_premya = Convert.ToDecimal(premiaFin.Text),

kol\_platezh = Convert.ToInt32(kol.Text),

data\_2pl = DateTime.Parse(date2.Text),

summa\_2pl = Convert.ToDecimal(plat2.Text),

summa\_platezh = Convert.ToDecimal(plat2.Text),

bank\_posr = bank.Text,

objecti = objectStr.Text,

idstrahovaya = Convert.ToInt32(IDstrahovaya),

primechanya = primech.Text

};

db.rassrochka.Add(rassroch);

db.SaveChanges();

}

if (per == 1)

{

db.SaveChanges();

}

}

System.Windows.MessageBox.Show("Добавление выполнено успешно!", "Успех!");

}

}

catch (Exception ex)

{

System.Windows.MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

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

{

new ChekWindow().Show();

this.Close();

}

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

{

switch (kol.SelectedItem.ToString())

{

case ("2"):

break;

case ("3"):

date3.IsReadOnly = false;

break;

default:

date3.IsReadOnly = false;

date4.IsReadOnly = false;

break;

}

}

}

}

StrahWindow

Title="Окно добавление страхователя" Height="500" Width="800" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Height="34" Margin="10,425,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

<Label Content="ФИО/Наименование\*" HorizontalAlignment="Left" Margin="44,107,0,0" VerticalAlignment="Top"/>

<Label Content="Дата рождения" HorizontalAlignment="Left" Margin="470,243,0,0" VerticalAlignment="Top"/>

<Label Content="Адрес\*" HorizontalAlignment="Left" Margin="125,138,0,0" VerticalAlignment="Top"/>

<Label Content="Рабочий телефон" HorizontalAlignment="Left" Margin="61,173,0,0" VerticalAlignment="Top"/>

<Label Content="Мобильный телефон 1\*" HorizontalAlignment="Left" Margin="32,208,0,0" VerticalAlignment="Top"/>

<Label Content="Мобильный телефон 2" HorizontalAlignment="Left" Margin="32,247,0,0" VerticalAlignment="Top"/>

<Label Content="Домашний телефон" HorizontalAlignment="Left" Margin="446,177,0,0" VerticalAlignment="Top"/>

<Label Content="Примечание" HorizontalAlignment="Left" Margin="88,278,0,0" VerticalAlignment="Top"/>

<Button Content="Добавление" x:Name="addbt" HorizontalAlignment="Left" Height="34" Margin="690,425,0,0" VerticalAlignment="Top" Width="92" Click="addbt\_Click"/>

<TextBox x:Name="FIO" HorizontalAlignment="Left" Height="30" Margin="187,103,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="564"/>

<Obselete:MaskedTextBox Mask="99/99/9999" x:Name="Datetb" HorizontalAlignment="Left" Height="30" Margin="572,243,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="adresstb" HorizontalAlignment="Left" Height="30" Margin="186,138,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="564"/>

<Obselete:MaskedTextBox Mask="8 (999) 999 99-99" HorizontalAlignment="Left" x:Name="rabteltb" Height="26" Margin="187,173,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<Obselete:MaskedTextBox Mask="8 (999) 999 99-99" MaxLength="64" x:Name="mob1tb" HorizontalAlignment="Left" Height="30" Margin="186,208,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="180"/>

<Obselete:MaskedTextBox Mask="8 (999) 999 99-99" MaxLength="64" x:Name="mob2tb" HorizontalAlignment="Left" Height="30" Margin="187,243,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox MaxLength="64" x:Name="domtel" HorizontalAlignment="Left" Height="30" Margin="571,173,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="179"/>

<TextBox x:Name="prim" HorizontalAlignment="Left" Height="60" Margin="186,278,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="564"/>

<ComboBox Name="kombobomba" HorizontalAlignment="Left" Margin="571,208,0,0" VerticalAlignment="Top" Width="179" Height="30">

<ComboBoxItem Content="юр. лицо"/>

<ComboBoxItem Content="физ. лицо"/>

</ComboBox>

<Label Content="Категория\*" HorizontalAlignment="Left" Margin="500,208,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

public partial class StrahWindow : Window

{

public StrahWindow()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

kombobomba.SelectedIndex = 0;

}

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

{

new ChekWindow().Show();

this.Close();

}

public static kursacEntities1 db = new kursacEntities1();

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

{

try

{

if (FIO.Text == "" || mob1tb.Text == "8 (\_\_\_) \_\_\_ \_\_-\_\_" || adresstb.Text == "")

{

System.Windows.MessageBox.Show("Одно или несколько обязательных полей не заполнено!", "Ошибка");

}

else

{

string mobile1 = ""; string mobile2 = "";

if (mob1tb.Text == "8 (\_\_\_) \_\_\_ \_\_-\_\_")

{

mobile1 = "";

}

else

{

mobile1 = rabteltb.Text;

}

if (mob2tb.Text == "8 (\_\_\_) \_\_\_ \_\_-\_\_")

{

mobile2 = "";

}

else

{

mobile2 = mob2tb.Text;

}

if (Datetb.Text == "\_\_.\_\_.\_\_\_\_")

{

var strahovately1 = new strahovately()

{

FIO\_Namenov = FIO.Text,

address = adresstb.Text,

rab\_telefon = mobile1,

mob\_telefon\_1 = mob1tb.Text,

mob\_telefon\_2 = mobile2,

dom\_telefon = domtel.Text,

kategorya = kombobomba.Text.ToString(),

primechanya = prim.Text

};

db.strahovately.Add(strahovately1);

MessageBox.Show("Добавлено", "Успех");

// Сохранить изменения в базе данных

db.SaveChanges();

}

else

{

var strahovately1 = new strahovately()

{

FIO\_Namenov = FIO.Text,

data\_rozden = DateTime.Parse(Datetb.Text),

address = adresstb.Text,

rab\_telefon = mobile1,

mob\_telefon\_1 = mob1tb.Text,

mob\_telefon\_2 = mobile2,

dom\_telefon = domtel.Text,

kategorya = kombobomba.Text.ToString(),

primechanya = prim.Text

};

db.strahovately.Add(strahovately1);

MessageBox.Show("Добавлено", "Успех");

// Сохранить изменения в базе данных

db.SaveChanges();

}

new AddWindow().Show();

this.Close();

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

}

}

VidWindow

Title="Окно добавления страховки" Height="240" Width="539" Icon="/КурсовойПроект;component/BestIco\_1.ico">

<Grid>

<Button Content="Назад" x:Name="exit" HorizontalAlignment="Left" Height="34" Margin="14,158,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

<TextBox Name="index" HorizontalAlignment="Left" Height="26" Margin="199,37,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="205"/>

<TextBox x:Name="naim" HorizontalAlignment="Left" Height="52" Margin="199,68,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="205"/>

<Label Content="Индекс страхователся\*" HorizontalAlignment="Left" Margin="56,37,0,0" VerticalAlignment="Top" Height="26" Width="136"/>

<Label Content="Наименование\*" HorizontalAlignment="Left" Margin="88,68,0,0" VerticalAlignment="Top" Height="26" Width="98"/>

<Label Content="Категория\*" HorizontalAlignment="Left" Margin="115,125,0,0" VerticalAlignment="Top" Height="26" Width="71"/>

<ComboBox Name="kat" Text="Здоровье" HorizontalAlignment="Left" Margin="199,125,0,0" Height="26" VerticalAlignment="Top" Width="205">

<ComboBoxItem Content="Здоровье"/>

<ComboBoxItem Content="Юр.Лицо"/>

<ComboBoxItem Content="Автомобиль"/>

<ComboBoxItem Content="Имущество"/>

<ComboBoxItem Content="Путешетвия"/>

</ComboBox>

<Button Content="Добавить" x:Name="add" Height="34" Margin="429,158,0,0" VerticalAlignment="Top" Click="add\_Click" HorizontalAlignment="Left" Width="92"/>

</Grid>

</Window>

public partial class VidWindow : Window

{

public static kursacEntities1 db = new kursacEntities1();

public VidWindow()

{

InitializeComponent();

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

kat.SelectedIndex = 0;

}

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

{

new ChekWindow().Show();

this.Close();

}

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

{

try

{

if (index.Text == "" || naim.Text == "")

{

System.Windows.MessageBox.Show("Одно или несколько обязательных полей не заполнено!", "Ошибка");

}

else

{

var strahovki = new strahovki()

{

index\_str = index.Text,

naimenov\_str = naim.Text,

kategoria\_str = kat.Text.ToString()

};

db.strahovki.Add(strahovki);

db.SaveChanges();

MessageBox.Show("Добавлено", "Успех");

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка, обратитесь к системному администратору \n Код ошибки: " + Convert.ToString(ex), "Ошибка, обратитесь к системному администратору");

}

}

}

}

MainWindow

Title="Окно входа" Height="311.5" Width="256" MaxHeight="311.5" MaxWidth="256" MinHeight="311.5" MinWidth="256" Icon="BestIco\_1.ico" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button Content="Вход" Name="login" HorizontalAlignment="Left" Height="34" Margin="141,236,0,0" VerticalAlignment="Top" Width="92" Click="login\_Click"/>

<Button Content="Выход" Name="exit" HorizontalAlignment="Left" Height="34" Margin="10,236,0,0" VerticalAlignment="Top" Width="92" Click="exit\_Click"/>

<TextBox Name="log" Height="26" Margin="10,41,15,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Background="White"/>

<PasswordBox Name="pass" Height="26" Margin="10,151,15,0" VerticalAlignment="Top" Background="White"/>

<Label Name="lbl1" Content="Введите пароль" HorizontalAlignment="Left" Margin="10,120,0,0" VerticalAlignment="Top" Height="26" Width="97"/>

<Label Content="Введите логин" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top" Height="26" Width="89"/>

<Label x:Name="TimerCount" Content="До повторного входа: " HorizontalAlignment="Left" Margin="30,205,0,0" VerticalAlignment="Top" Height="26" Width="97" Visibility="Hidden"/>

</Grid>

</Window>

public partial class MainWindow : Window

{

System.Windows.Threading.DispatcherTimer timer = new System.Windows.Threading.DispatcherTimer();

private void CenterWindowOnScreen()

{

double screenWidth = System.Windows.SystemParameters.PrimaryScreenWidth;

double screenHeight = System.Windows.SystemParameters.PrimaryScreenHeight;

double windowWidth = this.Width;

double windowHeight = this.Height;

this.Left = (screenWidth / 2) - (windowWidth / 2);

this.Top = (screenHeight / 2) - (windowHeight / 2);

}

public MainWindow()

{

InitializeComponent();

}

public static kursacEntities1 db = new kursacEntities1();

int logFail = 0;

private void timerTick(object sender, EventArgs e)

{

login.Visibility = Visibility.Visible;

logFail = 0;

}

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

{

string s = pass.Password;

string abc = "qwertyuiopasdfghjklzxcvbnm123456789";

//abc наш алфавит

int len = 4;

//len - число сдвигов в сторону

int a;

string code = string.Empty;

for (int i = 0; i < s.Length; i++)

{

int pos;

char c = s[i];

pos = abc.IndexOf(c);

a = pos + len;

if (a > 34) a -= 35;

code += abc[a];

}

if (logFail == 3)

{

timer.Tick += new EventHandler(timerTick);

timer.Interval = new TimeSpan(0, 0, 5);

timer.Start();

login.Visibility = Visibility.Hidden;

MessageBox.Show("Ошибка", "Привышено количество вводов", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

if (String.IsNullOrWhiteSpace(log.Text) || (String.IsNullOrWhiteSpace(pass.Password))) // Проверяю есть ли данные

{

MessageBox.Show("Введите данные", "Ошибка");

log.Clear();

pass.Clear();

logFail += 1;

return;

}

var t = MainWindow.db.agent.ToList().Find(c => c.agent\_login == log.Text.Trim() && c.agent\_parol == code); //Создание запроса на поиск

if ((log.Text == "") || (code == "") || (t == null)) // проверка пороля

{

MessageBox.Show("Неверные данные", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

log.Clear(); // Очиска полей

pass.Clear();

logFail += 1;

}

else

{

new Windows.AddWindow().Show(); // Закрытие окна

this.Close();

}

}

}

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

{

this.Close();

}

}

}