qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

Mainvindow

using System.Windows;

using System;

using HelixToolkit.Wpf;

using System.Windows.Media.Media3D;

using System.Windows.Threading;

using System.Globalization;

using System.Threading;

namespace pipe

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public class Cleaner

{

public double \_flow\_rate;

public double \_diameter;

public double \_optimal\_speed;

public double \_hydro\_resistance;

public double d\_Flow\_rate

{

get { return \_flow\_rate; }

set { \_flow\_rate = value; }

}

public double d\_Diameter

{

get { return \_diameter; }

set { \_diameter = value; }

}

public double d\_Optimal\_speed

{

get { return \_optimal\_speed; }

set { \_optimal\_speed = value; }

}

public double d\_Hydro\_resistance

{

get { return \_hydro\_resistance; }

set { \_hydro\_resistance = value; }

}

public double Calculate\_real\_speed()

{

return (d\_Flow\_rate) / ((Math.PI / 4) \* Math.Pow(d\_Diameter, 2) \* 3600);

}

public double Calculate\_hydro\_resistance(double k1, double k2, double e500)

{

return k1\*k2\*e500;

}

public double Calcualte\_diameter(double flow\_rate, double optimal\_speed)

{

return Math.Sqrt(flow\_rate / (0.785 \* optimal\_speed \* 3600 \* 1));

}

public double Calculate\_cyclon\_total\_area(double flow\_rate, double optimal\_speed)

{

return Math.Round(flow\_rate / (optimal\_speed \* 3600),3);

}

public double Calculate\_speed\_deviation(double real\_speed, double optimal\_speed)

{

return (Math.Abs(real\_speed - optimal\_speed) \* 100) / optimal\_speed;

}

}

public partial class MainWindow : Window

{

public double real\_density = 0;//действительная плотность

public double mT = 0;//действительноя вязкость

private CultureInfo myCulture;

public MainWindow()

{

InitializeComponent();

//для разделителя целой и дробной части

myCulture = new CultureInfo("en-US") { /\* Тут можно доопределить специфику культуры \*/ };

Thread.CurrentThread.CurrentCulture = myCulture;

Thread.CurrentThread.CurrentUICulture = myCulture;

txb\_flow\_rate.Text = "8000";

txb\_optimal\_speed.Text = "3.5";

txb\_e500.Text = "155";

txb\_flow\_temp.Text = "135";

btn\_create.IsEnabled = false;

DispatcherTimer timer = new DispatcherTimer();

timer.Interval = new TimeSpan(0, 0, 1);

timer.Tick += timer\_Tick;

timer.Start();

}

private void timer\_Tick( object sender, EventArgs e )

{

txb\_Memory.Text = string.Format("{0:0.00} MB",GC.GetTotalMemory(true) / 1024.0 / 1024.0);

}

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

{

txb\_results.Clear();

Cleaner test = new Cleaner();

test.d\_Optimal\_speed = double.Parse(txb\_optimal\_speed.Text);

test.d\_Flow\_rate = double.Parse(txb\_flow\_rate.Text);

test.d\_Hydro\_resistance = double.Parse(txb\_e500.Text);

test.d\_Diameter = test.Calcualte\_diameter(test.d\_Flow\_rate,test.d\_Optimal\_speed);//расчет диаметра

double d\_real\_speed = Math.Round(test.Calculate\_real\_speed(), 3);

double d\_hydro\_resistance = test.Calculate\_hydro\_resistance(1, 0.93, test.d\_Hydro\_resistance);

double d\_delta\_pressure = d\_hydro\_resistance \* real\_density \* Math.Pow(d\_real\_speed,2)/2;

double d\_CycTotalArea = test.Calculate\_cyclon\_total\_area(test.d\_Flow\_rate, test.d\_Optimal\_speed);

double d\_delta\_speed = test.Calculate\_speed\_deviation(d\_real\_speed, test.d\_Optimal\_speed);

string s\_result = String.Format(

"Действительная скорость = {0}, м/с \nКоэффициент гидравлического сопротивления одиночного циклона = {1} \nПотери давления в циклоне = {2}, Па \nДиаметр циклона = {3}, м \nPT = {4} кг/м3 \nμТ = {5}, Па\*с \nF = {6} м^2",

d\_real\_speed, d\_hydro\_resistance, Math.Round(d\_delta\_pressure, 5), Math.Round(test.d\_Diameter, 3), Math.Round(real\_density,3), Math.Round(mT,6) ,d\_CycTotalArea );

string s\_result1 = String.Format("\nΔw = {0}, %", Math.Round(d\_delta\_speed,3));

txb\_results.Text += s\_result + s\_result1;

this.DataContext = new Create\_main\_model(test.d\_Diameter);// создание модели

view1.ZoomExtents();

cuttingGroup1.CuttingPlanes.Clear();

cuttingGroup1.CuttingPlanes.Add(new Plane3D(new Point3D(0, 0, -4.25 \* test.d\_Diameter), new Vector3D(0, 0, 1)));

//btn\_create.IsEnabled = false;

d\_hydro\_resistance = 0;

d\_delta\_pressure = 0;

real\_density = 0;

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

experement\_window exp\_windows = new experement\_window();

exp\_windows.Show();

btn\_flow.IsEnabled = false;

}

private void About\_Program\_Click(object sender, RoutedEventArgs e)

{

AboutWindow form = new AboutWindow();

form.Show();

}

private void About\_Process\_Click(object sender, RoutedEventArgs e)

{

AboutProcess form = new AboutProcess();

form.Show();

}

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

{

GRAPH gr = new GRAPH();

gr.Show();

}

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

{

view1.ZoomExtents();

}

}

}

<Window x:Class="pipe.MainWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

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

xmlns:ht="clr-namespace:HelixToolkit.Wpf;assembly=HelixToolkit.Wpf"

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

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

xmlns:HelixToolkit="clr-namespace:HelixToolkit.Wpf;assembly=HelixToolkit.Wpf" Title="Модель" Height="600"

Width="930" TextElement.Foreground="{DynamicResource MaterialDesignBody}"

Background="{DynamicResource {x:Static SystemColors.ControlLightBrushKey}}"

FontFamily="pack://application:,,,/MaterialDesignThemes.Wpf;component/Resources/Roboto/#Roboto"

Icon="Resources/favicon.ico">

<Grid ShowGridLines="True">

<!--Разметка сетки GUI -->

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1.6\*"></ColumnDefinition>

<ColumnDefinition Width="auto"></ColumnDefinition>

<!--Колонка для разделителя-->

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<GridSplitter Grid.Row="0" Grid.Column="1" Grid.RowSpan="2" Width="5" VerticalAlignment="Stretch"

HorizontalAlignment="Center">

</GridSplitter>

<DockPanel Grid.Row="0" Grid.Column="0" Name="dp1">

<Menu DockPanel.Dock="Top" Background="DarkGray">

<MenuItem Header="Настройки">

<MenuItem Header="Чувствительность увеличения ">

<StackPanel Orientation="Horizontal">

<Slider Value="{Binding ZoomSensitivity, ElementName=view1}" Minimum="0.1" Maximum="5"

Width="200" />

<TextBlock Text="{Binding ZoomSensitivity, StringFormat='{}{0:0.0}', ElementName=view1}" />

</StackPanel>

</MenuItem>

</MenuItem>

<MenuItem Header="Камера">

<MenuItem IsCheckable="True" Header="Центрировать" Click="MenuItem\_Click" />

<MenuItem IsCheckable="True" Header="Orthographic"

IsChecked="{Binding Orthographic, ElementName=view1}" />

<MenuItem Header="Расстояние до ближней плоскости">

<TextBox Text="{Binding Camera.NearPlaneDistance, ElementName=view1}"></TextBox>

</MenuItem>

<MenuItem Header="Расстояние до дальней плоскости">

<TextBox Text="{Binding Camera.FarPlaneDistance, ElementName=view1}"></TextBox>

</MenuItem>

<MenuItem IsCheckable="True" Header="Вращание вокруг курсора"

IsChecked="{Binding RotateAroundMouseDownPoint, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Увеличение вокруг курсора"

IsChecked="{Binding ZoomAroundMouseDownPoint, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Бесконечное вращение"

IsChecked="{Binding InfiniteSpin, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="IsInertiaEnabled"

IsChecked="{Binding IsInertiaEnabled, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="IsPanEnabled"

IsChecked="{Binding IsPanEnabled, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Передвижение вкл."

IsChecked="{Binding IsMoveEnabled, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Вращение вкл."

IsChecked="{Binding IsRotationEnabled, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Увеличение вкл."

IsChecked="{Binding IsZoomEnabled, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Показать поле зрения"

IsChecked="{Binding IsChangeFieldOfViewEnabled, ElementName=view1}" />

<MenuItem Header="Чувствительность вращения">

<StackPanel Orientation="Horizontal">

<Slider Value="{Binding RotationSensitivity, ElementName=view1}" Minimum="0.1" Maximum="5"

Width="200" />

<TextBlock Text="{Binding RotationSensitivity, StringFormat='{}{0:0.0}', ElementName=view1}" />

</StackPanel>

</MenuItem>

<MenuItem Header="ZoomSensitivity">

<StackPanel Orientation="Horizontal">

<Slider Value="{Binding ZoomSensitivity, ElementName=view1}" Minimum="0.1" Maximum="5"

Width="200" />

<TextBlock Text="{Binding ZoomSensitivity, StringFormat='{}{0:0.0}', ElementName=view1}" />

</StackPanel>

</MenuItem>

<MenuItem Header="CameraInertiaFactor">

<StackPanel Orientation="Horizontal">

<Slider Value="{Binding CameraInertiaFactor, ElementName=view1}" Minimum="0.7" Maximum="1"

Width="200" />

<TextBlock

Text="{Binding CameraInertiaFactor, StringFormat='{}{0:0.000}', ElementName=view1}" />

</StackPanel>

</MenuItem>

</MenuItem>

<MenuItem Header="Вид">

<MenuItem IsCheckable="True" Header="Показать систему координат"

IsChecked="{Binding ShowCoordinateSystem, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Показать куб вращения"

IsChecked="{Binding ShowViewCube, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Информация о камере"

IsChecked="{Binding ShowCameraInfo, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Количество полигонов"

IsChecked="{Binding ShowTriangleCountInfo,ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Поле зрения"

IsChecked="{Binding ShowFieldOfView, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Частота кадров"

IsChecked="{Binding ShowFrameRate, ElementName=view1}" />

<MenuItem IsCheckable="True" Header="Цель камеры"

IsChecked="{Binding ShowCameraTarget, ElementName=view1}">

<Separator Height="100" Margin="0" Width="100" />

</MenuItem>

<MenuItem Header="Подпись">

<TextBox Text="{Binding Title, ElementName=view1}" Width="200" />

</MenuItem>

<MenuItem Header="Подпись 1">

<TextBox Text="{Binding SubTitle, ElementName=view1}" Width="200" />

</MenuItem>

</MenuItem>

<MenuItem x:Name="AboutMenu" Header="Дополнительно">

<MenuItem x:Name="About\_Program" Header="О программе" Click="About\_Program\_Click" />

<MenuItem x:Name="About\_Process" Header="Описание процесса" Click="About\_Process\_Click" />

<MenuItem x:Name="Grap" Header="Графики" Click="Grap\_Click" />

</MenuItem>

</Menu>

<!--Порт вывода-->

<ht:HelixViewport3D x:Name="view1" ZoomExtentsWhenLoaded="True" ShowCoordinateSystem="True"

UpDownPanSensitivity="-5" Background="White">

<ht:HelixViewport3D.DefaultCamera>

<PerspectiveCamera Position="1,1,3" LookDirection="-1,-1,-1" UpDirection="0,0,25" FieldOfView="60"

NearPlaneDistance="0.1" />

</ht:HelixViewport3D.DefaultCamera>

<ht:DefaultLights />

<ht:CuttingPlaneGroup x:Name="cuttingGroup1">

<ModelVisual3D Content="{Binding Model}" />

<!--

<ht:CuttingPlaneGroup.CuttingPlanes>

<ht:Plane3D Position="0,0,5" Normal="0,0,-1"/>

</ht:CuttingPlaneGroup.CuttingPlanes>

-->

</ht:CuttingPlaneGroup>

</ht:HelixViewport3D>

</DockPanel>

<Button x:Name="btn\_create" Content="Построить и рассчитать" Grid.Column="2" HorizontalAlignment="Left"

Margin="10,385,0,0" VerticalAlignment="Top" Width="192" Click="button\_Click\_1" IsDefault="True"

Height="46" />

<Button x:Name="btn\_flow" Grid.Column="2" HorizontalAlignment="Left" Margin="10,331,0,0" VerticalAlignment="Top"

Width="192" Click="button1\_Click" Height="45">

<TextBlock TextWrapping="Wrap">

Состав входного потока

</TextBlock>

</Button>

<TextBox x:Name="txb\_flow\_rate" Grid.Column="2" HorizontalAlignment="Left" Height="26" Margin="144,42,0,0"

TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Background="White" />

<Label x:Name="lbl\_flow\_rate" HorizontalAlignment="Left" Grid.Column="2" Margin="10,35,0,0"

VerticalAlignment="Top">

<TextBlock TextWrapping="Wrap" Width="125">

Производительность, м^3/ч:

</TextBlock>

</Label>

<Label x:Name="lbl\_optimal\_velocity" Grid.Column="2" HorizontalAlignment="Left" Margin="10,70,0,0"

VerticalAlignment="Top">

<TextBlock TextWrapping="Wrap" Width="80">

Оптимальная скорость, м/с:

</TextBlock>

</Label>

<TextBox x:Name="txb\_optimal\_speed" Grid.Column="2" HorizontalAlignment="Left" Height="26" Margin="144,75,0,0"

TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Background="White" />

<TextBox x:Name="txb\_results" Grid.Column="2" IsReadOnly="true" HorizontalAlignment="Left" Height="141"

Margin="10,183,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="337" Background="White">

</TextBox>

<Label x:Name="e500" Grid.Column="2" HorizontalAlignment="Left" Margin="10,110,0,0" VerticalAlignment="Top">

<TextBlock TextWrapping="Wrap" Width="100">

<!-- Коэффициент гидравлического сопротивления циклона диаметром 500мм: -->

E500:

</TextBlock>

</Label>

<TextBox x:Name="txb\_e500" Grid.Column="2" HorizontalAlignment="Left" Height="26" Margin="144,104,0,0"

TextWrapping="Wrap" VerticalAlignment="Top" Width="120" Background="White" />

<Label x:Name="lbl\_flow\_temp" Grid.Column="2" HorizontalAlignment="Left" Margin="10,141,0,0"

VerticalAlignment="Top">

<TextBlock TextWrapping="Wrap" Width="80">

Температура газовй смеси:

</TextBlock>

</Label>

<TextBox x:Name="txb\_flow\_temp" Grid.Column="2" HorizontalAlignment="Left" Height="26" Margin="144,147,0,0"

TextWrapping="Wrap" VerticalAlignment="Top" Width="120" Background="White" />

<TextBox x:Name="txb\_Memory" Grid.Column="2" HorizontalAlignment="Left" Height="26" Margin="148,537,0,0"

TextWrapping="Wrap" VerticalAlignment="Top" Width="67" />

<Label x:Name="lbl\_Memory" Content="Использовано памяти:" Grid.Column="2" HorizontalAlignment="Left"

Margin="9,538,0,0" VerticalAlignment="Top" Height="26" />

</Grid>

</Window>