Российская Федерация

ПРОГРАММА ДЛЯ ЭВМ

ПРОГРАММА ТЕПЛОФИЗИЧЕСКОГО РАСЧЕТА ТОПЛИВНОГО БАКА СПГ

Фрагменты исходного текста программы

Листов 9

Авторы:

Смелик Анатолий Анатольевич

Вдовичев Антон Андреевич

Горшков Сергей Николаевич

Российская Федерация

Анапа

2020 г

РАСПЕЧАТКА ИСХОДНОГО ТЕКСТА ПРОГРАММЫ

Файл MainWindow.xaml

```
<Window x:Class="TermoPhysCalc.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:TermoPhysCalc"
    mc:Ignorable="d"
    Title="Теплофизический расчет" SizeToContent="WidthAndHeight"
    Closing="MainWindow_OnClosing"
    Loaded="MainWindow_OnLoaded">
  <Grid>
    <ScrollViewer CanContentScroll="True" HorizontalScrollBarVisibility="Auto"
VerticalScrollBarVisibility="Auto">
      <Grid>
        <Grid.ColumnDefinitions>
          <ColumnDefinition Width="Auto" />
          <ColumnDefinition Width="Auto" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
          <RowDefinition Height="*" />
          <RowDefinition Height="Auto" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0" Grid.Column="0">
          <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
          </Grid.RowDefinitions>
          <Expander Grid.Row="0">
            <Expander.Header>
              <Label>Геометрические значения</Label>
            </Expander.Header>
            <Grid>
              <Grid.RowDefinitions>
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
              </Grid.RowDefinitions>
              <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto" />
                <ColumnDefinition Width="*" />
              </Grid.ColumnDefinitions>
              <Label Grid.Row="0" Grid.Column="0" Content="delta1 (толщина наружной стенки, м) = " />
              <TextBox Grid.Row="0" Grid.Column="1" Name="delta1_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="1" Grid.Column="0" Content="delta2 (толщина слоя изоляции, м) = " />
              <TextBox Grid.Row="1" Grid.Column="1" Name="delta2_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="2" Grid.Column="0" Content="delta3 (толщина внутренней стенки, м) = " />
              <TextBox Grid.Row="2" Grid.Column="1" Name="delta3_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="3" Grid.Column="0" Content="S (площадь плоской стенки, м^2) = " />
              <TextBox Grid.Row="3" Grid.Column="1" Name="S_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="4" Grid.Column="0"
                  Content="d_экв (эквивалентный диаметр топливного бака, м) = " />
              <TextBox Grid.Row="4" Grid.Column="1" Name="d_ekv_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="5" Grid.Column="0" Content="l (длина топливного бака, м) = " />
              <TextBox Grid.Row="5" Grid.Column="1" Name="l_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
            </Grid>
          </Expander>
          <Expander Grid.Row="1">
            <Expander.Header>
              <Label>Физические свойства</Label>
            </Expander.Header>
            <Grid>
              <Grid.RowDefinitions>
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
              </Grid.RowDefinitions>
              <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto" />
                <ColumnDefinition Width="*" />
              </Grid.ColumnDefinitions>
              <Label Grid.Row="0" Grid.Column="0" Content="tg1 (температура наружного воздуха, град) = "
/>
              <TextBox Grid.Row="0" Grid.Column="1" Name="tg1_txt" Margin="5"
                    PreviewTextInput="TextBoxInt_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="1" Grid.Column="0"
                   Content="tg2 (температура криогенной жидкости, град) = " />
              <TextBox Grid.Row="1" Grid.Column="1" Name="tg2_txt" Margin="5"
                    PreviewTextInput="TextBoxInt_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="2" Grid.Column="0"
                   Content="betta (температурный коэффициент объемного расширения для воздуха) = " />
              <TextBox Grid.Row="2" Grid.Column="1" Name="betta_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="3" Grid.Column="0"
                   Content="nu_ж1 (коэффициент кинематической вязкости воздуха, м^2/сек) = " />
              <TextBox Grid.Row="3" Grid.Column="1" Name="nug1_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="4" Grid.Column="0"
                   Content="nu_ж2 (коэффициент кинематической вязкости криогенной жидкости, м^2/сек) =
" />
              <TextBox Grid.Row="4" Grid.Column="1" Name="nug2_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="5" Grid.Column="0"
                   Content="w_ж2 (скорость движения криогенной жидкости, м/сек) = " />
              <TextBox Grid.Row="5" Grid.Column="1" Name="wg2_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="6" Grid.Column="0"
                   Content="c_p (удельная теплоемкость криогенной жидкости, кДж/(кг*град)) = " />
              <TextBox Grid.Row="6" Grid.Column="1" Name="cp_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="7" Grid.Column="0"
                   Content="lambda_ж2 (коэффициент теплопроводности криогенной жидкости,
кДж/(кг*град)) = " />
              <TextBox Grid.Row="7" Grid.Column="1" Name="lambdag2_txt" Margin="5"
                    PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="8" Grid.Column="0"
                  Content="lambda3 (коэффициент теплопроводности наружной стенки, кДж/(кг*град)) = " />
              <TextBox Grid.Row="8" Grid.Column="1" Name="lambda3_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="9" Grid.Column="0"
                  Content="lambda_изл (коэффициент теплопроводности слоя изоляции, кДж/(кг*град)) = " />
              <TextBox Grid.Row="9" Grid.Column="1" Name="lambda_izl_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="10" Grid.Column="0"
                  Content="lambda1 (коэффициент теплопроводности внутренней стенки, кДж/(кг*град)) = "
/>
              <TextBox Grid.Row="10" Grid.Column="1" Name="lambda1_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="11" Grid.Column="0"
                  Content="tc1 (температура среды на границе наружной стенки, град) = " />
              <TextBox Grid.Row="11" Grid.Column="1" Name="tc1_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
              <Label Grid.Row="12" Grid.Column="0"
                  Content="tc4 (температура криогенной жидкости на границе внутренней стенки, град) = " />
              <TextBox Grid.Row="12" Grid.Column="1" Name="tc4_txt" Margin="5"
                   PreviewTextInput="TextBoxDouble_PreviewTextInput" MinWidth="30" />
            </Grid>
          </Expander>
        </Grid>
        <Grid Grid.Row="0" Grid.Column="1">
          <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
          </Grid.RowDefinitions>
          <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto" />
            <ColumnDefinition Width="*" />
          </Grid.ColumnDefinitions>
          <Label Grid.Column="0" Grid.Row="0" Content="q (плотность теплового потока, Вт/м^2) = " />
          <Label Grid.Column="0" Grid.Row="1" Content="R (термическое сопротивление, (м^2*град)/Вт) = "
/>
          <Label Grid.Column="0" Grid.Row="2"
              Content="tc2 (температура на границе изоляции и наружной стенки, град) = " />
          <Label Grid.Column="0" Grid.Row="3"
              Content="tc3 (температура на границе изоляции и внутренней стенки, град) = " />
          <Label Grid.Column="0" Grid.Row="4" Content="k (коэффициент теплопередачи, Вт/(м^2*град)) = "
/>
          <Label Grid.Column="0" Grid.Row="5" Content="Q (суммарный теплоприток, Вт) = " />

          <Label Grid.Column="1" Name="lbl_q" Grid.Row="0" MinWidth="30" />
          <Label Grid.Column="1" Name="lbl_R" Grid.Row="1" MinWidth="30" />
          <Label Grid.Column="1" Name="lbl_tc2" Grid.Row="2" MinWidth="30" />
          <Label Grid.Column="1" Name="lbl_tc3" Grid.Row="3" MinWidth="30" />
          <Label Grid.Column="1" Name="lbl_k" Grid.Row="4" MinWidth="30" />
          <Label Grid.Column="1" Name="lbl_Q" Grid.Row="5" MinWidth="30" />
        </Grid>
        <Grid Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2">
          <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="Auto" />
          </Grid.ColumnDefinitions>
          <Button Grid.Column="1" Margin="5" Padding="3" Content="Рассчитать" Click="ButtonBase_OnClick"
/>
        </Grid>
```

```
            </Grid>
        </ScrollViewer>
    </Grid>
</Window>
```

Файл MainWindows.xaml.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Windows;
using System.Windows.Input;

namespace TermoPhysCalc
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private const string _fileName = "cache.txt";

        private IDictionary<double, double> _epsOtnTable = new Dictionary<double, double>
        {
            {1, 1.65},
            {2, 1.5},
            {5, 1.34},
            {10, 1.23},
            {15, 1.17},
            {20, 1.13},
            {30, 1.07},
            {40, 1.03}
        };

        private IDictionary<double, double> _reinoldTable = new Dictionary<double, double>
        {
            {2.2 * Math.Pow(10, -3), 2.2},
            {2.3 * Math.Pow(10, -3), 3.6},
            {2.5 * Math.Pow(10, -3), 4.9},
            {3 * Math.Pow(10, -3), 7.5},
            {3.5 * Math.Pow(10, -3), 10},
            {4 * Math.Pow(10, -3), 12.2},
            {5 * Math.Pow(10, -3), 16.5},
            {6 * Math.Pow(10, -3), 20},
            {7 * Math.Pow(10, -3), 24},
            {8 * Math.Pow(10, -3), 27},
            {9 * Math.Pow(10, -3), 30},
            {10 * Math.Pow(10, -3), 33}
        };

        private IDictionary<double, double> _muTableFluid = new Dictionary<double, double>
        {
            {-165.05, 117.38 * Math.Pow(10, -6)},
            {-162.70999999999998, 112.45 * Math.Pow(10, -6)},
            {-160.7, 102.31 * Math.Pow(10, -6)},
            {-158.82999999999998, 104.44 * Math.Pow(10, -6)},
            {-157.14, 101.18 * Math.Pow(10, -6)},
            {-155.63, 98.546 * Math.Pow(10, -6)},
            {-152.24, 92.733 * Math.Pow(10, -6)},
            {-149.32, 88.1 * Math.Pow(10, -6)},
            {-146.72, 84.322 * Math.Pow(10, -6)},
            {-144.44, 75.386 * Math.Pow(10, -6)},
            {-142.28, 73.306 * Math.Pow(10, -6)},
    {-140.28, 71.42 * Math.Pow(10, -6)},
    {-136.78, 68.156 * Math.Pow(10, -6)},
    {-133.62, 65.262 * Math.Pow(10, -6)},
    {-130.8, 62.739 * Math.Pow(10, -6)}
};

private IDictionary<double, double> _prandtlTableAir = new Dictionary<double, double>
{
    {-50, 0.728},
    {-40, 0.728},
    {-30, 0.723},
    {-20, 0.716},
    {-10, 0.712},
    {0, 0.707},
    {10, 0.705},
    {20, 0.703},
    {30, 0.701},
    {40, 0.699},
    {50, 0.698},
    {60, 0.696},
    {70, 0.694},
    {80, 0.692},
    {90, 0.69},
    {100, 0.688},
    {110, 0.687},
    {120, 0.686},
    {130, 0.685},
    {140, 0.684},
    {150, 0.683},
    {160, 0.682},
    {170, 0.682},
    {180, 0.681},
    {190, 0.681},
    {200, 0.68},
    {250, 0.677},
    {300, 0.674},
    {350, 0.676},
    {400, 0.678},
    {450, 0.683},
    {500, 0.687},
    {550, 0.693},
    {600, 0.699},
    {650, 0.703},
    {700, 0.706},
    {750, 0.71},
    {800, 0.713},
    {850, 0.715},
    {900, 0.717},
    {950, 0.718},
    {1000, 0.719},
    {1100, 0.722},
    {1200, 0.724}
};

private double GetPrandtl<TType>(IDictionary<TType, double> table, TType t)
{
    return table[
        table.Keys.Last(kk => Math.Abs((dynamic) kk - t) == table.Keys.Min(k => Math.Abs((dynamic) k - t)))];
}

public MainWindow()
{
    InitializeComponent();
```

```
        }

        private void TextBoxInt_PreviewTextInput(object sender, TextCompositionEventArgs e)
        {
            e.Handled = !(char.IsDigit(e.Text.First()) || e.Text == "-");
        }

        private void TextBoxDouble_PreviewTextInput(object sender, TextCompositionEventArgs e)
        {
            e.Handled = !(char.IsDigit(e.Text.First()) || e.Text == "," || e.Text == "-");
        }

        private double GetDoubleFromText(string txt)
        {
            return double.Parse(string.Join("", txt.Where(x => char.IsDigit(x) || x == '-' || x == ',')));
        }

        private void ButtonBase_OnClick(object sender, RoutedEventArgs e)
        {
            if (string.IsNullOrWhiteSpace(tg1_txt.Text) ||
                string.IsNullOrWhiteSpace(tg2_txt.Text) ||
                string.IsNullOrWhiteSpace(tc1_txt.Text) ||
                string.IsNullOrWhiteSpace(d_ekv_txt.Text) ||
                string.IsNullOrWhiteSpace(wg2_txt.Text) ||
                string.IsNullOrWhiteSpace(nug1_txt.Text) ||
                string.IsNullOrWhiteSpace(nug2_txt.Text) ||
                string.IsNullOrWhiteSpace(betta_txt.Text) ||
                string.IsNullOrWhiteSpace(lambda1_txt.Text) ||
                string.IsNullOrWhiteSpace(lambdag2_txt.Text) ||
                string.IsNullOrWhiteSpace(lambda3_txt.Text) ||
                string.IsNullOrWhiteSpace(lambda_izl_txt.Text) ||
                string.IsNullOrWhiteSpace(lambda1_txt.Text) ||
                string.IsNullOrWhiteSpace(delta1_txt.Text) ||
                string.IsNullOrWhiteSpace(delta2_txt.Text) ||
                string.IsNullOrWhiteSpace(delta3_txt.Text))
            {
                MessageBox.Show("Необходимо ввести все данные!", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
                return;
            }

            const double g = 9.81;
            var tg1 = int.Parse(tg1_txt.Text);
            var tg2 = int.Parse(tg2_txt.Text);
            var tc1 = GetDoubleFromText(tc1_txt.Text);
            var tc4 = GetDoubleFromText(tc4_txt.Text);
            var d_ekv = GetDoubleFromText(d_ekv_txt.Text);
            var l = GetDoubleFromText(l_txt.Text);
            var wg2 = GetDoubleFromText(wg2_txt.Text);
            var nug1 = GetDoubleFromText(nug1_txt.Text);
            var nug2 = GetDoubleFromText(nug2_txt.Text);
            var betta = GetDoubleFromText(betta_txt.Text);
            var lambdag1 = GetDoubleFromText(lambda1_txt.Text);
            var lambdag2 = GetDoubleFromText(lambdag2_txt.Text);
            var lambda3 = GetDoubleFromText(lambda3_txt.Text);
            var lambda_izl = GetDoubleFromText(lambda_izl_txt.Text);
            var lambda1 = GetDoubleFromText(lambda1_txt.Text);
            var delta1 = GetDoubleFromText(delta1_txt.Text);
            var delta2 = GetDoubleFromText(delta2_txt.Text);
            var delta3 = GetDoubleFromText(delta3_txt.Text);
            var S = GetDoubleFromText(S_txt.Text);
            var cp = GetDoubleFromText(cp_txt.Text);
            var mug2 = GetPrandtl(_muTableFluid, tg2);
            var muc4 = GetPrandtl(_muTableFluid, tc4);

            double Nug2;
            var Prg2 = (mug2 * cp) / lambdag2;
            var Prc4 = (muc4 * cp) / lambdag2;

            var Grg1 = (g * betta * Math.Pow(d_ekv, 3)) / nug1 * (tg1 - tc1);
            var Prg1 = GetPrandtl(_prandtlTableAir, tg1);
            var Prc1 = GetPrandtl(_prandtlTableAir, tc1);
            var Nug1 = 0.5 * Math.Pow(Grg1 * Prg1, 0.25) * Math.Pow(Prg1 / Prc1, 0.25);

            var epst = Math.Pow(Prg2 / Prc4, 0.25);
            var Reg2 = (wg2 * d_ekv) / nug2;
            var K0 = GetPrandtl(_reinoldTable, Reg2);
            var eps_otn = GetPrandtl(_epsOtnTable, l / d_ekv);
            if (Reg2 < 2300)
            {
                Nug2 = 4 * epst;
            }
            else if (Reg2 > 2300 && Reg2 < 10_000)
            {
                Nug2 = K0 * Math.Pow(Prg2, 0.43) * epst * eps_otn;
            }
            else
            {
                if (l / d_ekv < 50)
                    Nug2 = 0.021 * Math.Pow(Reg2, 0.8) * Math.Pow(Prg2, 0.43) * epst * (1 + 2 * (d_ekv / l));
                else
                    Nug2 = 0.021 * Math.Pow(Reg2, 0.8) * Math.Pow(Prg2, 0.43) * epst;
            }

            var alpha1 = (Nug1 * lambdag1) / d_ekv;
            var alpha2 = (Nug2 * lambdag2) / d_ekv;
            var q = (tg1 - tg2) /
                    (1 / alpha1 + delta3 / lambda3 + delta2 / lambda_izl + delta1 / lambda1 + 1 / alpha2);
            var R = (tg1 - tg2) / q;
            var k = 1 / R;
            var tc2 = tc1 - q * 1 / alpha1;
            var tc3 = tc2 - q * delta2 / lambda_izl;
            var Q = q * S;

            lbl_q.Content = q;
            lbl_R.Content = R;
            lbl_k.Content = k;
            lbl_tc2.Content = tc2;
            lbl_tc3.Content = tc3;
            lbl_Q.Content = Q;
        }

        private void MainWindow_OnClosing(object sender, CancelEventArgs e)
        {
            var strToSave = "";
            strToSave += "delta1_txt=" + delta1_txt.Text + "\n";
            strToSave += "delta2_txt=" + delta2_txt.Text + "\n";
            strToSave += "delta3_txt=" + delta3_txt.Text + "\n";
            strToSave += "S_txt=" + S_txt.Text + "\n";
            strToSave += "d_ekv_txt=" + d_ekv_txt.Text + "\n";
            strToSave += "l_txt=" + l_txt.Text + "\n";
            strToSave += "tg1_txt=" + tg1_txt.Text + "\n";
            strToSave += "tg2_txt=" + tg2_txt.Text + "\n";
            strToSave += "betta_txt=" + betta_txt.Text + "\n";
            strToSave += "nug1_txt=" + nug1_txt.Text + "\n";
            strToSave += "nug2_txt=" + nug2_txt.Text + "\n";
            strToSave += "wg2_txt=" + wg2_txt.Text + "\n";
            strToSave += "cp_txt=" + cp_txt.Text + "\n";
            strToSave += "lambdag2_txt=" + lambdag2_txt.Text + "\n";
            strToSave += "lambda3_txt=" + lambda3_txt.Text + "\n";
            strToSave += "lambda_izl_txt=" + lambda_izl_txt.Text + "\n";
            strToSave += "lambda1_txt=" + lambda1_txt.Text + "\n";
            strToSave += "tc1_txt=" + tc1_txt.Text + "\n";
            strToSave += "tc4_txt=" + tc4_txt.Text;
            File.WriteAllText(_fileName, strToSave);
        }

        private void MainWindow_OnLoaded(object sender, RoutedEventArgs e)
        {
            if (!File.Exists(_fileName)) return;
            var cache = File.ReadAllLines(_fileName);
            if (!cache.Any()) return;
            delta1_txt.Text = cache[0].Split('=')[0] == "delta1_txt" ? cache[0].Split('=')[1] : "";
            delta2_txt.Text = cache[1].Split('=')[0] == "delta2_txt" ? cache[1].Split('=')[1] : "";
            delta3_txt.Text = cache[2].Split('=')[0] == "delta3_txt" ? cache[2].Split('=')[1] : "";
            S_txt.Text = cache[3].Split('=')[0] == "S_txt" ? cache[3].Split('=')[1] : "";
            d_ekv_txt.Text = cache[4].Split('=')[0] == "d_ekv_txt" ? cache[4].Split('=')[1] : "";
            l_txt.Text = cache[5].Split('=')[0] == "l_txt" ? cache[5].Split('=')[1] : "";
            tg1_txt.Text = cache[6].Split('=')[0] == "tg1_txt" ? cache[6].Split('=')[1] : "";
            tg2_txt.Text = cache[7].Split('=')[0] == "tg2_txt" ? cache[7].Split('=')[1] : "";
            betta_txt.Text = cache[8].Split('=')[0] == "betta_txt" ? cache[8].Split('=')[1] : "";
            nug1_txt.Text = cache[9].Split('=')[0] == "nug1_txt" ? cache[9].Split('=')[1] : "";
            nug2_txt.Text = cache[10].Split('=')[0] == "nug2_txt" ? cache[10].Split('=')[1] : "";
            wg2_txt.Text = cache[11].Split('=')[0] == "wg2_txt" ? cache[11].Split('=')[1] : "";
            cp_txt.Text = cache[12].Split('=')[0] == "cp_txt" ? cache[12].Split('=')[1] : "";
            lambdag2_txt.Text = cache[13].Split('=')[0] == "lambdag2_txt" ? cache[13].Split('=')[1] : "";
            lambda3_txt.Text = cache[14].Split('=')[0] == "lambda3_txt" ? cache[14].Split('=')[1] : "";
            lambda_izl_txt.Text = cache[15].Split('=')[0] == "lambda_izl_txt" ? cache[15].Split('=')[1] : "";
            lambda1_txt.Text = cache[16].Split('=')[0] == "lambda1_txt" ? cache[16].Split('=')[1] : "";
            tc1_txt.Text = cache[17].Split('=')[0] == "tc1_txt" ? cache[17].Split('=')[1] : "";
            tc4_txt.Text = cache[18].Split('=')[0] == "tc4_txt" ? cache[18].Split('=')[1] : "";
        }
    }
}
```