МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 3**

по дисциплине: ”Разработка простейшего приложения в визуальной среде WPF”

на тему: ***”Разработка приложения, реализующего циклический вычислительный процесс”***

Вариант 14

Выполнил**:** студент группы 107012019 Кузьмич И.В.

Принял**:** Гурский Н.Н.

Минск 2020

***Цель лабораторной работы:*** изучить средства отладки программ в среде С#. Составить и отладить программу для циклического вычислительного процесса.

## Постановка задачи: Разработать приложение с реализацией циклических вычислений в соответствии с индивидуальным заданием.

***Задание:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 14 | 00.2 | 00.8 |  | 12 |  |

***Код программы:***

***MainWindow.xaml***

<Window x:Class="Laba3.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:Laba3"

mc:Ignorable="d"

Title="Лаб.раб. N3"

Height="500" Width="400"

MaxHeight="1000" MaxWidth="800"

MinHeight="500" MinWidth="400">

<Grid Style="{StaticResource DefaultView}">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="140\*"/>

<RowDefinition Height="350\*"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Grid HorizontalAlignment="Center" VerticalAlignment="Top" Width="200" Height="129" Grid.Row="0" Margin="0,5,0,0" Grid.RowSpan="2">

<Label Content="X1=" Width="35" Height="26" Margin="31,20,0,83" HorizontalAlignment="Left"/>

<TextBox x:Name="textBox1" TextWrapping="NoWrap" Height="20" Margin="66,22,10,85" Text="0.1"/>

<Label Content="X2=" HorizontalAlignment="Left" Margin="31,60,0,43" Width="35" Height="26" RenderTransformOrigin="0.297,0.578"/>

<TextBox x:Name="textBox2" TextWrapping="NoWrap" Height="20" VerticalAlignment="Top" Margin="66,66,10,0" Text="1"/>

</Grid>

<TextBox x:Name="outPanel" Margin="30,0,30,0" TextWrapping="Wrap" VerticalAlignment="Center" Height="260" IsReadOnly="True" Grid.Row="1" Grid.ColumnSpan="2" VerticalScrollBarVisibility="Visible" ScrollViewer.CanContentScroll="True"/>

<Button Content="ПУСК" Margin="0,10,0,20" Width="130" Height="36" Grid.Row="2" Grid.ColumnSpan="2" Click="MainBtnClick"/>

</Grid>

</Window>

***MainWindow.xaml.cs***

using System;

using System.Windows;

using static System.Math;

namespace Laba3

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private double ConvertToDouble(string value)

{

try

{

return Convert.ToDouble(value);

}

catch (System.FormatException ex)

{

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

return 0;

}

}

private void MainBtnClick(object sender, RoutedEventArgs e)

{

double xn = 0, xk = 0;

xn = ConvertToDouble(textBox1.Text);

xk = ConvertToDouble(textBox2.Text);

int k = 12;

double h = (xk - xn) / k;

if (h == 0)

{

h = 1;

}

var YX = Model.FunctionYx(xn, xk, h);

var SX = Model.FunctionSx(xn, xk, h);

string line = "---------------------------\n";

outPanel.Text = "| x | S(x) | Y(x) |\n" + line;

int i = 0;

while (xn <= xk)

{

outPanel.Text +=

$"| {xn:F3} | {SX[i]:F6} | {YX[i]:F6} |\n" + line;

i++;

xn += h;

xn = Round(xn, 6);

}

}

}

}

**Model.cs**

using System.Collections.Generic;

using static System.Math;

namespace Laba3

{

class Model

{

public static List<double> FunctionYx(double xn, double xk, double h)

{

List<double> fList = new List<double>() { };

while (xn <= xk)

{

double yx = (1 / 4.0) \* ((xn + 1) / Sqrt(xn) \* Sinh(Sqrt(xn)) - Cosh(Sqrt(xn)));

fList.Add(yx);

xn += h;

xn = Round(xn, 6);

}

return fList;

}

public static List<double> FunctionSx(double xn, double xk, double h)

{

List<double> sxList = new List<double>() { };

double f, T, sum;

int k = 2;

while (xn <= xk)

{

f = xn / 6;

sum = f;

int n = 1;

while (n < k)

{

T = ((n + 1) \* xn / ((4 \* n + 6) \* Pow(n, 2)));

f \*= T;

sum += f;

n++;

}

sxList.Add(sum);

xn += h;

xn = Round(xn, 6);

}

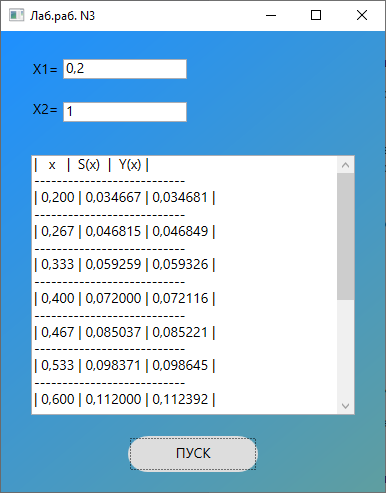
return sxList;

}

}

}

***Скриншоты результатов:***



**Рисунок 1. Результат работы программы**