Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

По дисциплине: «Современные платформы программирования»

Выполнил:

Студент 3 курса

Группы ПО-4

Иваненко И. Л.

Проверил:

Монтик Н. С.

Брест 2022

Вариант 10

Цель работы: Приобрести практические навыки в области разработки десктопных приложений.

# Общее задание

Разработать оконное приложение с использованием Java API, использующее один вспомогательный поток, вычисляющий заданную сумму и выполняющий вывод результата вычисления в любой визуальный компонент. Все исходные данные вводятсяв соответствующие визуальные компоненты. Обработать исключения.

# Код программы:

## MainWindow.xaml:

<Window x:Class="Labs.MPP.Lab2.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"  
 mc:Ignorable="d"  
 Title="MainWindow" Height="90" Width="120">  
 <StackPanel>  
 <TextBox Name="UpperLimitTextBox" MinHeight="30">  
 </TextBox>  
 <Button Margin="6" Padding="6" Name="PerformCalculationButton" MinHeight="30" Click="PerformCalculationButton\_OnClick">  
 Calculate  
 </Button>  
 <TextBlock Name="CalculationResult" MinHeight="30">  
   
 </TextBlock>  
 </StackPanel>  
</Window>

## MainWindow.xaml.cs:

using System;  
using System.Globalization;  
using System.Threading;  
using System.Windows;  
  
namespace Labs.MPP.Lab2  
{  
 */// <summary>  
 /// Interaction logic for MainWindow.xaml  
 /// </summary>* public partial class MainWindow  
 {  
 private const int **LowerLimit** = 1;  
 private const int **StepCalculationTimeout** = 200;  
   
 public MainWindow()  
 {  
 InitializeComponent();  
 }  
  
 */// <summary>  
 /// Method that is invoked when button PerformCalculationButton is clicked.  
 /// </summary>  
 /// <param name="sender">Object that invoked this method.</param>  
 /// <param name="e">Object that contains addition information about event.</param>* private void PerformCalculationButton\_OnClick(object sender, RoutedEventArgs e)  
 {  
 if (TryParseUpperLimit(UpperLimitTextBox.Text, out var upperLimit))  
 {  
 CalculationResult.Text = PerformCalculation(upperLimit).ToString(CultureInfo.InvariantCulture);  
 }  
 }  
  
 */// <summary>  
 /// Checks whether string is a valid upper limit series and parses string to int.  
 /// </summary>  
 /// <param name="upperLimitString">Input string.</param>  
 /// <param name="upperLimit">Out int upper limit.</param>  
 /// <returns>  
 /// Returns true of input string is a valid upper limit series.  
 /// Otherwise returns false.  
 /// </returns>* private static bool TryParseUpperLimit(string upperLimitString, out int upperLimit)  
 {  
 upperLimit = 0;  
 if (string.IsNullOrWhiteSpace(upperLimitString) || !int.TryParse(upperLimitString, out upperLimit))  
 {  
 return false;  
 }  
  
 return upperLimit > 0;  
 }  
  
 */// <summary>  
 /// Calculates series.  
 /// </summary>  
 /// <param name="upperLimit">Upper limit of the series.</param>  
 /// <returns>Result sum of the series.</returns>* private static double PerformCalculation(int upperLimit)  
 {  
 var resultSum = 0d;  
 for (var stepNumber = **LowerLimit**; stepNumber <= upperLimit; stepNumber++)  
 {  
 var currentStep = CalculateCurrentStep(stepNumber);  
 resultSum += currentStep;  
 Thread.Sleep(**StepCalculationTimeout**);  
 }  
  
 return resultSum;  
 }  
  
 */// <summary>  
 /// Calculates current step of the series.  
 /// </summary>  
 /// <param name="stepNumber">Number of the step of the series.</param>  
 /// <returns>Result sum of the series.</returns>* private static double CalculateCurrentStep(int stepNumber)  
 {  
 return (2 \* stepNumber - 1) / Math.Pow(2, stepNumber);  
 }  
 }  
}

# Результат выполнения:

Graphical user interface, text, application, email

Description automatically generated

# Вывод

В ходе выполнения данной лабораторной работы были приобретены практические навыки разработки десктопных приложений на платформе Windows с использованием платформы .NET Core и ЯП C# 9.0.