**національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ЛАБОРАТОРНА РОБОТА №6**

Виконав:

студент групи ІПЗ-2006б

Купчик В. Г.

Київ – 2023

**Завдання до лабораторної роботи**

**Мета роботи:** В процесі виконання роботи необхідно розробити сторінкове / віконне WPF застосування

**Завдання до лабораторної роботи:**

1. Створити сторінкове / віконне WPF застосування.

2. Розробити графічний інтерфейс користувача с використанням

технології WPF для реалізації завдання, відповідно варіанту.

Варіант - 15



Код:

<Window x:Class="lab6.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:lab6"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Label Content="Введіть довжину першого катета:" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>

<TextBox Name="txtCatet1" HorizontalAlignment="Left" Margin="209,14,0,0" VerticalAlignment="Top" Width="120"/>

<Label Content="Введіть довжину другого катета:" HorizontalAlignment="Left" Margin="10,50,0,0" VerticalAlignment="Top"/>

<TextBox Name="txtCatet2" HorizontalAlignment="Left" Margin="203,54,0,0" VerticalAlignment="Top" Width="120"/>

<Button Content="Обчислити" Name="btnCalculate" HorizontalAlignment="Left" Margin="120,90,0,0" VerticalAlignment="Top" Width="100" Click="btnCalculate\_Click"/>

<Label Content="Довжина гіпотенузи:" HorizontalAlignment="Left" Margin="10,130,0,0" VerticalAlignment="Top"/>

<TextBlock Name="txtHypotenuse" HorizontalAlignment="Left" Margin="180,130,0,0" VerticalAlignment="Top"/>

<Label Content="Площа трикутника:" HorizontalAlignment="Left" Margin="10,170,0,0" VerticalAlignment="Top"/>

<TextBlock Name="txtArea" HorizontalAlignment="Left" Margin="180,170,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace lab6

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

if (double.TryParse(txtCatet1.Text, out double catet1) && double.TryParse(txtCatet2.Text, out double catet2))

{

double hypotenuse = Math.Sqrt(catet1 \* catet1 + catet2 \* catet2);

double area = 0.5 \* catet1 \* catet2;

txtHypotenuse.Text = hypotenuse.ToString("F2");

txtArea.Text = area.ToString("F2");

}

else

{

MessageBox.Show("Введіть коректні значення для катетів.");

}

}

}

}

Результати виконання :

Изображение выглядит как снимок экрана, компьютер, текст, программное обеспечение

Автоматически созданное описание

Висновок

В цій лабораторній роботі було розроблено сторінкове / віконне WPF застосування для обчислення довжини гіпотенузи і площі прямокутного трикутника за двома катетами. Ця лабораторна робота показала, як можна створити простий, але функціональний додаток для обчислення геометричних параметрів прямокутного трикутника. WPF дозволяє створювати багатофункціональні інтерфейси, які забезпечують зручний взаємозв'язок користувача з програмою.

Контрольні питання

1. Основні компоненти архітектури WPF включають в себе:
   * Елементи керування (Controls): Вони представляють графічні компоненти, такі як кнопки, текстові поля, списки і т. д.
   * Панелі компонування (Layout Panels): Вони визначають, як елементи керування розміщуються на вікні, наприклад, Grid, StackPanel.
   * Шаблони (Templates): Дозволяють визначити зовнішній вигляд елементів керування.
   * Джерела даних (Data Sources): Використовуються для зв'язування даних з елементами керування.
   * Стилі (Styles): Визначають зовнішній вигляд елементів керування і можуть бути застосовані до декількох елементів.
   * Тригери (Triggers): Дозволяють реагувати на події і змінювати вигляд елементів керування.
2. Ключові ієрархії фундаментальних класів WPF включають в себе:
   * UIElement: Базовий клас для всіх візуальних елементів.
   * FrameworkElement: Розширює UIElement та надає функціональність для компонування та зв'язування даних.
   * Control: Розширює FrameworkElement і представляє стандартні елементи керування, такі як Button або TextBox.
   * Panel: Базовий клас для контейнерів компонування, таких як Grid або StackPanel.
3. Компонування елементів управління в WPF відбувається за допомогою контейнерів компонування, таких як Grid, StackPanel, або DockPanel, які визначають, як елементи розміщуються та співвідносяться один з одним на вікні.
4. Основні контейнери компонування в WPF включають в себе:
   * Grid: Дозволяє розміщати елементи у вигляді сітки з рядками та стовпцями.
   * StackPanel: Розташовує елементи вгороджені один в одного в горизонтальному або вертикальному напрямку.
   * DockPanel: Дозволяє прикріплювати елементи до одного з країв вікна.
5. Клас Grid призначений для розміщення елементів у вигляді сітки з рядками та стовпцями. Він надає можливість точно контролювати розташування та розмір елементів на вікні.
6. StackPanel служить для розташування елементів вгороджені один в одного в горизонтальному або вертикальному напрямку. Він спрощує компонування елементів в одному напрямку.
7. Елементи керування вмістом (Content Controls) призначені для відображення та управління вмістом в інших елементах. Наприклад, Button може мати текст або інший вміст всередині себе. Інші приклади це Label або ContentPresenter.
8. Клас ContentControl служить для відображення та керування вмістом інших елементів керування, включаючи текст та інші елементи.
9. Текстові елементи управління в WPF включають в себе TextBlock (для відображення тексту), TextBox (для введення тексту), і Label (для позначення інших елементів).
10. Для списків в WPF використовуються елементи ListBox (для відображення списку елементів) та ComboBox (для вибору одного зі списку).
11. Спеціалізовані елементи управління в WPF призначені для конкретних завдань. Наприклад, DatePicker для вибору дати, Slider для вибору значення на діапазоні, або DataGrid для відображення таблиць даних.
12. Командна модель WPF дозволяє визначити та використовувати команди, які можна пов'язати з елементами керування. Наприклад, Button може виконувати команду, коли його натискають, що спрощує обробку подій та взаємодію зі шаблонами. Наприклад, команда може виконувати дії, такі як збереження або відкриття файлу.