# Руководство разработчика

1. **Описание системы**
   1. Технические требования

Минимальные аппаратные требования для программы «Логическое выражение»:

* Компьютер с 64-разрядный процессором 1,8 ГГц или более мощный. Рекомендуется четырехъядерный или с большим количеством ядер. Процессоры ARM не поддерживаются;
* Не менее 4 ГБ ОЗУ. На используемые ресурсы влияет множество факторов; для стандартных профессиональных решений рекомендуется 16 ГБ ОЗУ.
  1. Программные требования
* Visual Studio 2022, библиотека .NET;
* 64-разрядные операционные системы: Windows 11, Windows 10.

1. **Архитектура системы**
   1. Структура графического приложения

Графическое приложение на базе библиотек .NET Framework может содержать неограниченное количество элементов управления (Button, CheckBox…), которые располагаются на поле Grid.

* 1. Данные

Данные представляют собой файлы формата .txt. Работа с данными происходит в интерфейсе «Выбрать файл» и «Создать файл».

1. **Дизайн приложения**

Каждое приложение создается в файле MainWindow.xaml путем переноса с помощью мыши необходимых элементов из панели элементов на поле Grid. После переноса элемента автоматически создаётся код, характеризующий наш элемент, его расположение и размер на поле Grid.

Ниже приведен образец xaml – файла, готового к созданию графического приложения на языке С# на базе библиотеки .NET.

<Window x:Class="Логическое\_выражение.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:Логическое\_выражение"

mc:Ignorable="d"

Title="Конвертация в логическое выражение" Height="450" Width="800" WindowStartupLocation="CenterScreen" Topmost="True">

<Grid>

<Button Content="Выбрать файл" HorizontalAlignment="Left" Margin="33,33,0,0" VerticalAlignment="Top" Height="21" Width="88" Click="cmOpen"/>

<TextBox x:Name="tbText1" HorizontalAlignment="Left" Margin="273,32,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120" Height="21"/>

<Label Content="Путь к искомому файлу:" HorizontalAlignment="Left" Margin="131,28,0,0" VerticalAlignment="Top"/>

<Button Content="Создать логическое выражение!" HorizontalAlignment="Left" Height="25" Margin="33,144,0,0" VerticalAlignment="Top" Width="244" Click="cmMake"/>

<Label Content="Введите количество переменных функции:" HorizontalAlignment="Left" Margin="33,80,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbCount" HorizontalAlignment="Left" Margin="281,88,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="48"/>

<Button Content="Ввести" HorizontalAlignment="Left" Margin="341,87,0,0" VerticalAlignment="Top" Click="cmCountOfVar"/>

<Label Content="Полученное выражение:" HorizontalAlignment="Left" Margin="33,192,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbText3" HorizontalAlignment="Left" Margin="190,196,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Button Content="Создать файл" HorizontalAlignment="Left" Margin="33,245,0,0" VerticalAlignment="Top" Height="23" Width="102" Click="cmMakeFile"/>

<Label Content="Путь файла с логическим выражением:" HorizontalAlignment="Left" Margin="158,245,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.595,-0.353"/>

<TextBox x:Name="tbName" HorizontalAlignment="Left" Margin="398,249,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Выберите необходимый базис:" HorizontalAlignment="Left" Height="30" Margin="398,28,0,0" VerticalAlignment="Top" Width="185"/>

<CheckBox Content="И" HorizontalAlignment="Left" Margin="588,33,0,0" VerticalAlignment="Top" Height="19" Width="91"/>

<CheckBox Content="ИЛИ" HorizontalAlignment="Left" Margin="588,52,0,0" VerticalAlignment="Top"/>

<CheckBox Content="НЕ" HorizontalAlignment="Left" Margin="588,72,0,0" VerticalAlignment="Top"/>

<CheckBox Content="И-НЕ" HorizontalAlignment="Left" Margin="588,92,0,0" VerticalAlignment="Top"/>

<CheckBox Content="ИЛИ-НЕ" HorizontalAlignment="Left" Margin="588,112,0,0" VerticalAlignment="Top"/>

<CheckBox Content="ИСКЛЮЧАЮЩЕЕ ИЛИ" HorizontalAlignment="Left" Margin="588,132,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

Отдельно рассмотрим создание используемых элементов программы:

* 1. Создание кнопок

Для создания необходимо перенести элемент Button на панель Grid (рисунок 1).

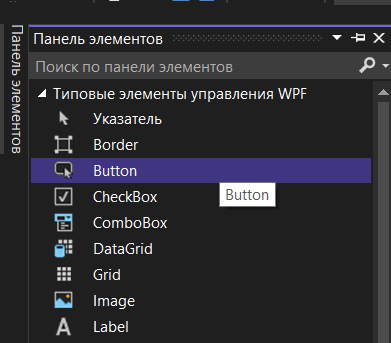


Рисунок 1 –расположение Button

Далее для происхождения какого-либо действия при нажатии кнопки необходимо выделить нужную кнопку и нажать на знак молнии (рисунок 2).

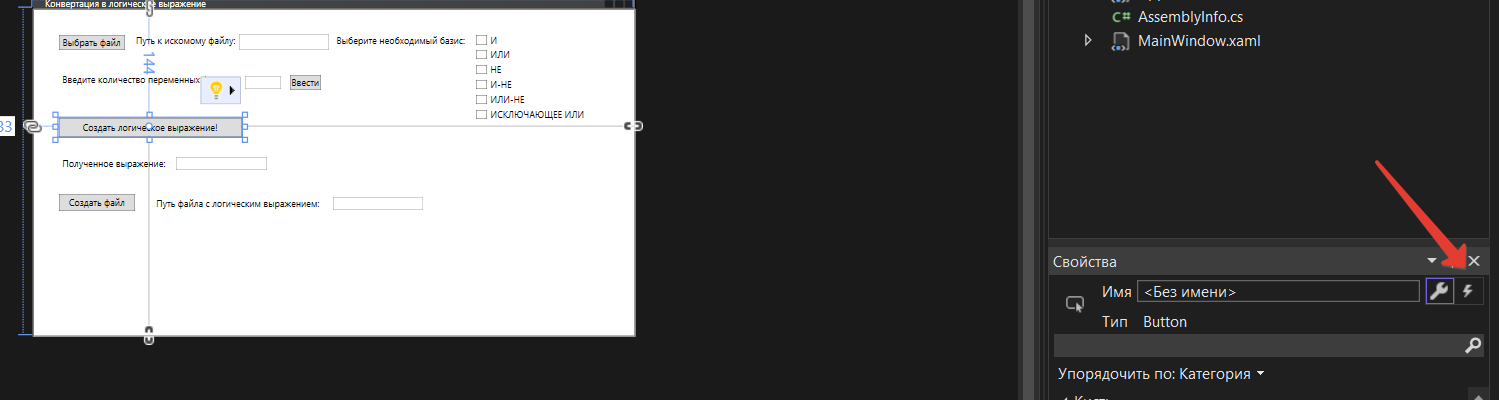


Рисунок 2 – изменение свойств кнопки

В поле «Click» пишем имя нашей функции, которая будет выполняться при нажатии данной кнопки (рисунок 3).

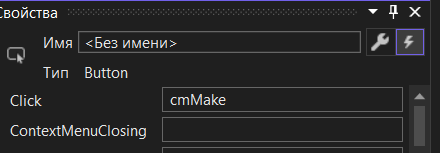


Рисунок 3 – необходимое поле Click

Данная команда автоматически создастся в файле MainWindow.xaml.cs (рисунок 4).

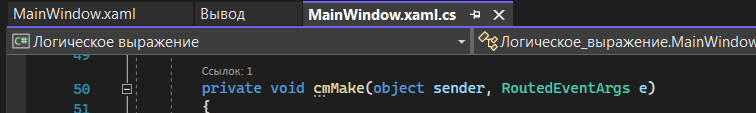


Рисунок 4 – пример созданной функции

Ниже приведен образец функции cmOpen, осуществляющей открытие файла при нажатии на кнопку «Выбрать файл».

private void cmOpen(object sender, RoutedEventArgs e)

{

string Name1;

OpenFileDialog dialog1 = new OpenFileDialog();

dialog1.Filter = "Text files (\*.txt)|\*.txt|All files|\*.\*";

if (dialog1.ShowDialog() == true)

{

Name1 = dialog1.FileName;

tbText1.Text = Name1;

MessageBox.Show("Файл " + Name1 + " успешно открыт!");

}

else

{

MessageBox.Show("Ошибка при работе с файлам!");

return;

}

}

* 1. Создание TextBox

Для создания необходимо перенести элемент TextBox на панель Grid (рисунок 5).

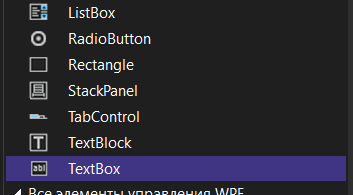


Рисунок 5 – расположение TextBox

Далее для последующего взаимодействия с информацией, введенной в TextBox, необходимо дать ему имя (рисунок 6).

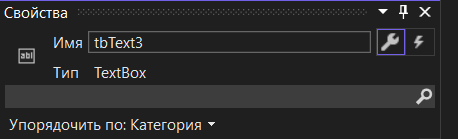


Рисунок 6 – создание имени для TextBox

Далее вся информация, введенная в данное поле, будет храниться в переменной tbText3.Text.

Ниже приведен пример обращения к переменной такого типа.

logic = makeLogic(st, k);

tbText3.Text = logic;

MessageBox.Show(logic);

* 1. Создание CheckBox

Для создания необходимо перенести элемент CheckBox на панель Grid

(рисунок 7).

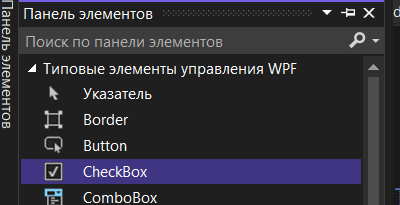


Рисунок 7 – расположение CheckBox

Далее выбрать пункт молния (рисунок 8).

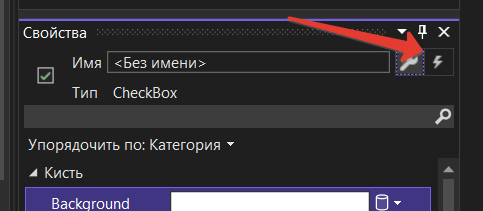


Рисунок 8 – изменение свойств CheckBox

В поле «Check» пишем имя нашей функции, которая будет выполняться при отметки данного поля в CheckBox (рисунок 9).

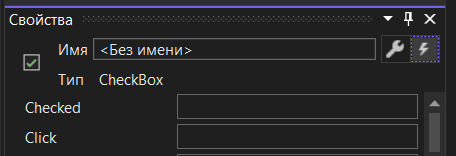


Рисунок 9 – необходимое поле Checked

Данная команда автоматически создастся в файле MainWindow.xaml.cs (рисунок 10).

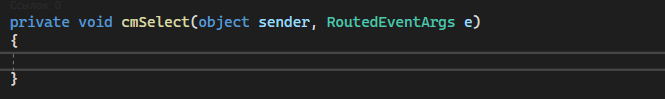


Рисунок 10 – пример созданной функции

1. **Структура программы**

В файле MainWindow.xaml содержится конструктор интерфейса графического приложения и код, соответствующий данному интерфейсу.

В файле MainWindow.xaml.cs содержится класс MainWindow, в котором находятся все функции, реализуемые данным интерфейсом.