# СВПП. Лабораторная работа 1. Калькулятор

## Задание.

Разработайте калькулятор согласно макету.

| 0,00 | | | |
| --- | --- | --- | --- |
| С | <- | \* | / |
| 7 | 8 | 9 | - |
| 4 | 5 | 6 | + |
| 1 | 2 | 3 | = |
| 0 | | , |

## Описание решения.

### Создаем проект WPF Calculator

### Делаем разметку

Создадим сетку 4х6

<Grid>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

</Grid>

### Добавим кнопки

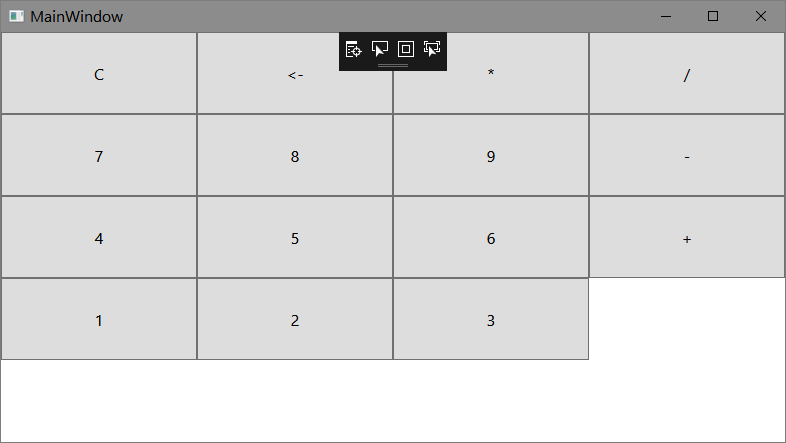
<Button Grid.Column="0" Grid.Row="1" Content="C" />

<Button Grid.Column="1" Grid.Row="1" Content="&lt;-" />

<Button Grid.Column="2" Grid.Row="1" Content="\*" />

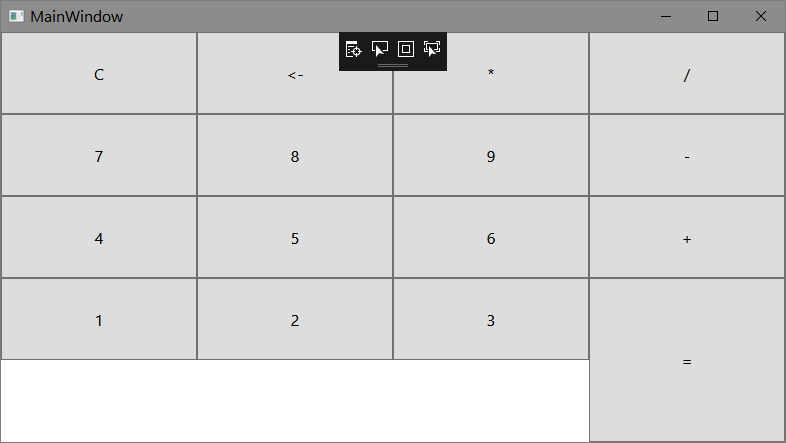
<Button Grid.Column="3" Grid.Row="1" Content="/" />

Обратите внимание, как экранируется символ "<"



Знак "=" занимает 2 строки

<Button Grid.Column="3" Grid.Row="3" Content="=" Grid.RowSpan="2" />

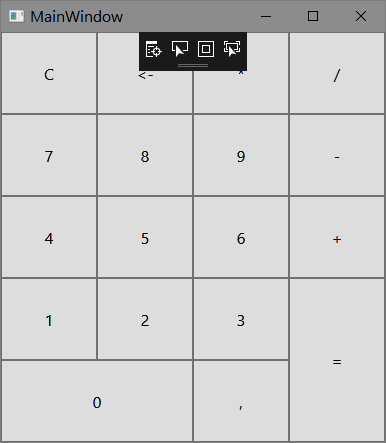


Символ "0" занимает 2 столбца

<Button Grid.Column="0" Grid.Row="5" Content="0" Grid.ColumnSpan="2" />

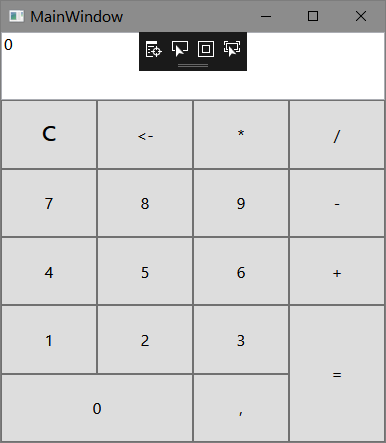


Поправим ширину формы (найдите где)



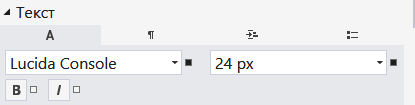
Добавим текстовое поле

<TextBox Grid.Row="0" Grid.ColumnSpan="4" />



### Отформатируем изображение

Поправим размер шрифта в окне свойств для текстового поля:



Теперь видно, как это можно записать в xaml:

<Button Grid.Column="0" Grid.Row="1" Content="C" FontSize="24" FontFamily="Lucida Console"/>

Скопируйте этот формат на все элементы:

FontSize="24" FontFamily="Lucida Console"

Текст в текстовом поле расположен неправильно – по левому краю.

Разместим его справа:

HorizontalContentAlignment ="Right"

И добавим вертикальное выравнивание по центру:

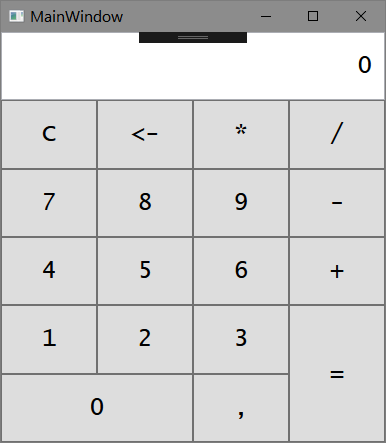
VerticalContentAlignment ="Center"

А также отступы слева и справа:

Padding="10 0 10 0"

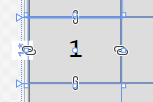
Вот как теперь это выглядит:

<TextBox Grid.ColumnSpan="4" Text="0" FontSize="24" FontFamily="Lucida Console" HorizontalContentAlignment ="Right" VerticalContentAlignment ="Center" Padding="10 0 10 0" />



### Перейдем к обработчикам событий. Цифровые кнопки

Дважды щелкнем в конструкторе по кнопке "1".



Обработчик будет создан автоматически

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

{

}

Поскольку для всех кнопок с цифрами будет выполняться одно и то же действие – цифра будет просто добавляться к текстовому полю, им всем подойдет один обработчик.

Переименуем его (не игнорируйте желтую лампочку):

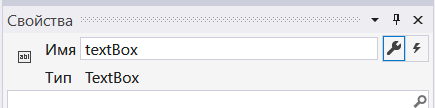
private void ButtonNumber\_Click(object sender, RoutedEventArgs e)

{

}

Теперь нужно извлечь цифру из контента кнопки и записать ее в текстовое поле.

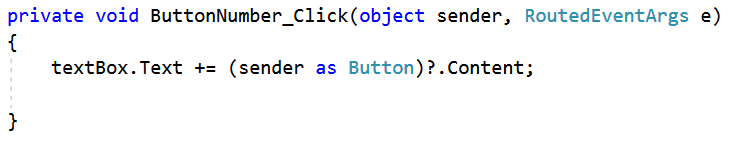
Для этого сначала добавим имя текстовому полю в окне свойств:



Или так

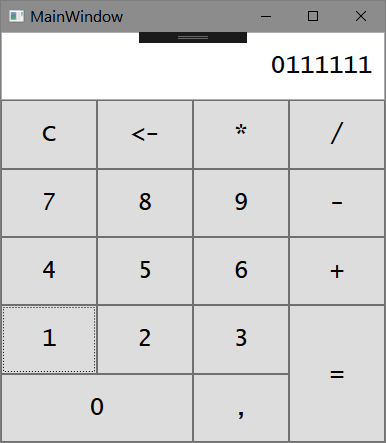
<TextBox x:Name="textBox"

Теперь добавим код к обработчику:

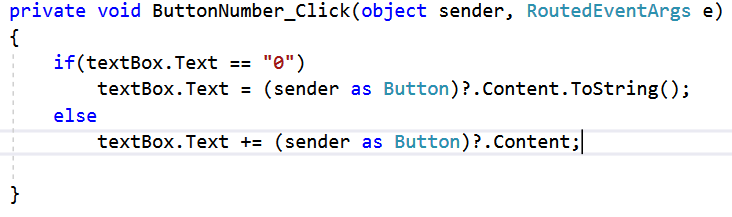


Что означает знак вопроса в коде?

Проверим работу кнопки "1" (пока мы назначили обработчик только ей):

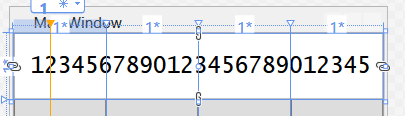


Очевидно, что "0" не нужно оставлять в начале строки. Если в строке содержится только "0", заменим строку, иначе – добавим:

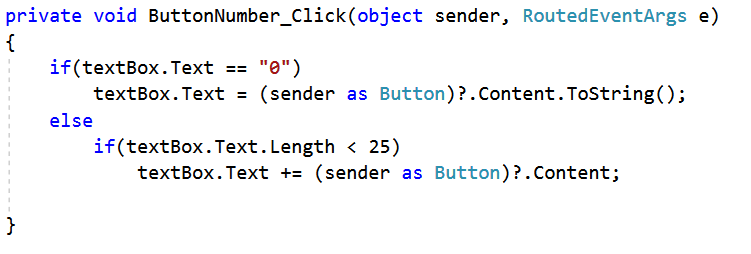


И ограничим длину строки разумными пределами.

Проверим длину текстового поля в конструкторе – у меня помещается 25 символов.

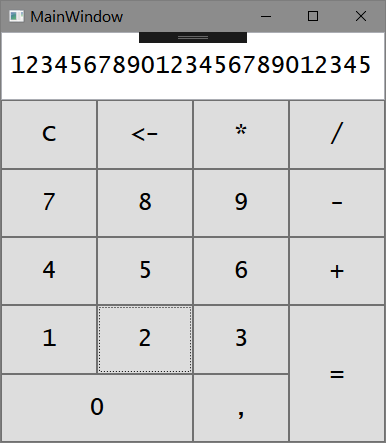


Сделаем так, чтобы новые символы добавлялись, только если в строке есть место:



Теперь назначим этот обработчик всем цифровым кнопкам:

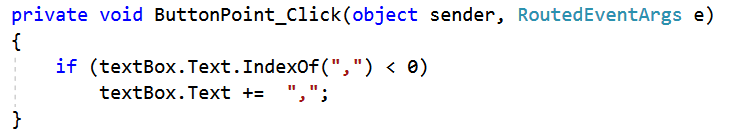
Click="ButtonNumber\_Click"



Убедимся, что лишние символы не добавляются.

### Сделаем отдельный обработчик для запятой.

Запятая может быть в строке только одна. Поэтому запятую добавляем, только если в строке она не найдена.

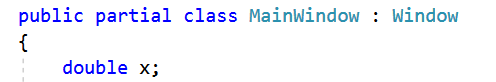


Что возвращает IndexOf?

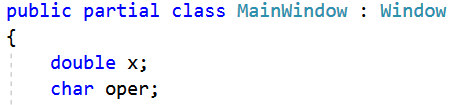
### Теперь научим наш калькулятор считать. Обработчики для кнопок с операциями

Мы реализуем простейший алгоритм калькулятора. Пользователь вводит число, потом знак операции (число при этом исчезает из текстового поля), а затем второе число. При нажатии на "=" получаем результат, который можно использовать в дальнейших вычислениях.

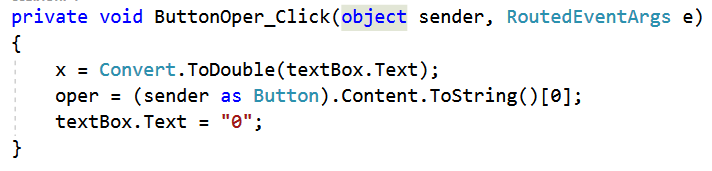
При нажатии на клавишу с операцией число из текстового поля должно где-то сохраняться. Заведем для этого переменную в виде поля класса



Так же должен сохраняться знак операции



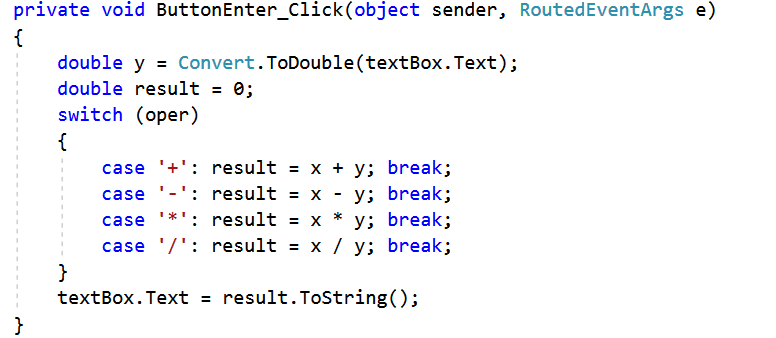
И само текстовое поле должно очищаться



Назначим этот обработчик всем кнопкам с операциями

### Вычисления выполняются при нажатии на кнопку "=".

При этом извлекаются сохраненное число и знак операции.



Проверьте работу калькулятора.

Протестируйте деление на 0, попытку вычисления без знака операции, попытайтесь перемножить очень большие числа.

### Самостоятельная часть

Самостоятельно запрограммируйте кнопку "С" – полная очистка памяти и поля калькулятора.

Кнопка "<-" удаляет последний введенный символ.