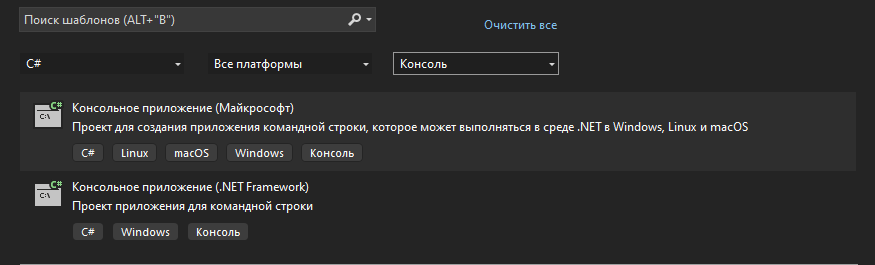
**Разработка интерфейса пользователя**



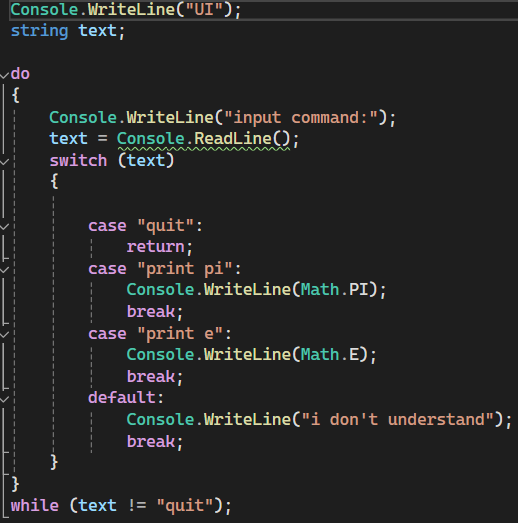
Интерфейс пользователя обеспечивает передачу информации между пользователем и приложением.

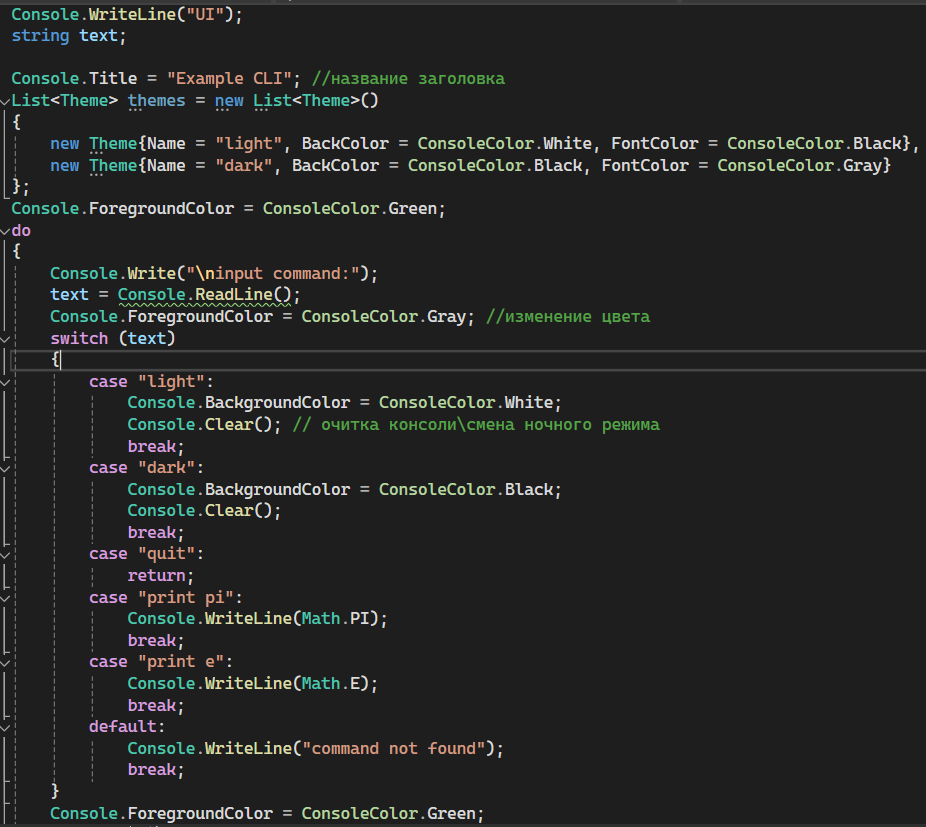
Виды интерфейсов пользователя:

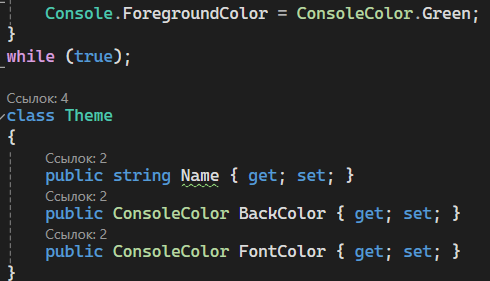
* Интерфейс команды строки (**CLI** – Command Line Interface)

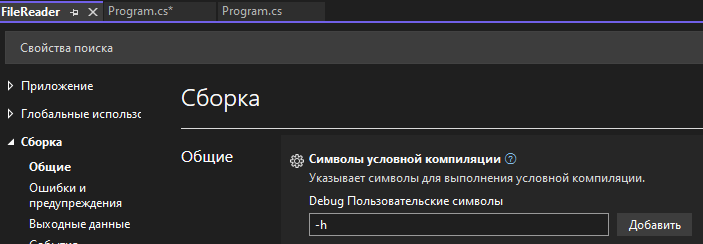
Для разового действия используется команда.

Для эмуляции меню можно использовать бесконечный цикл.

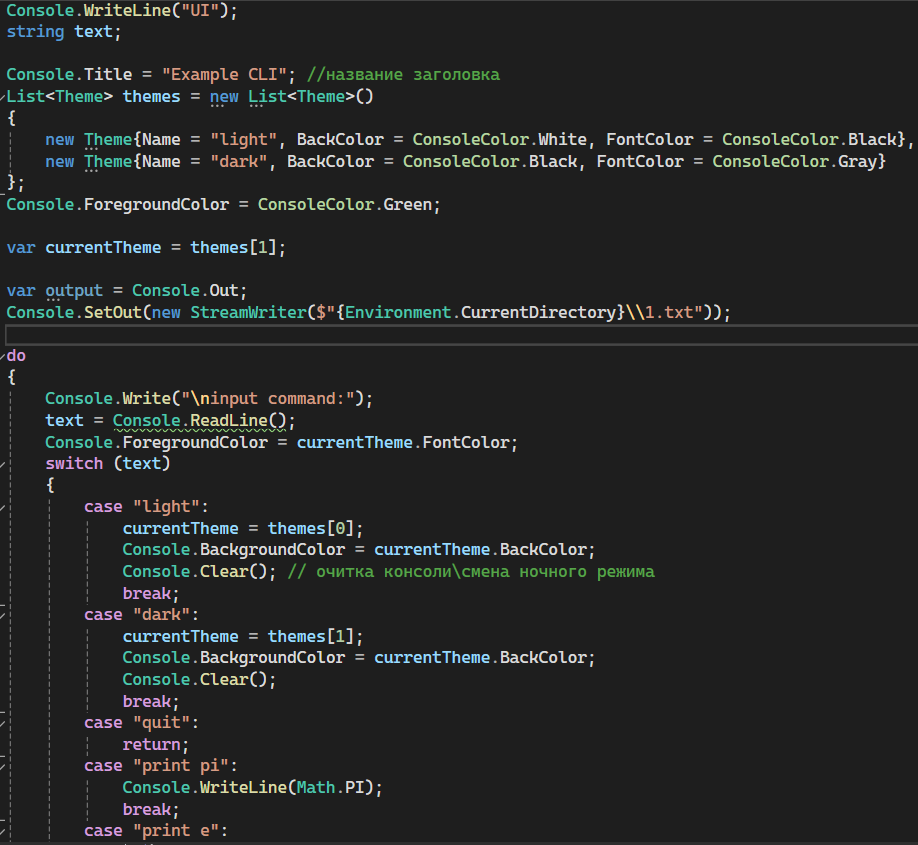


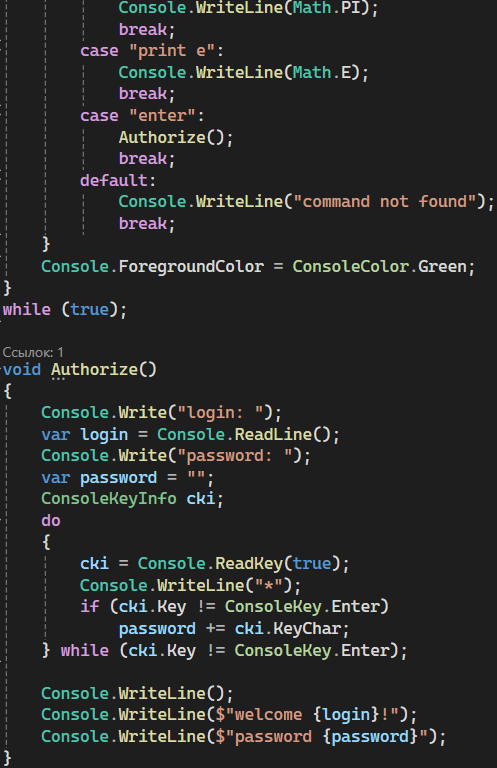


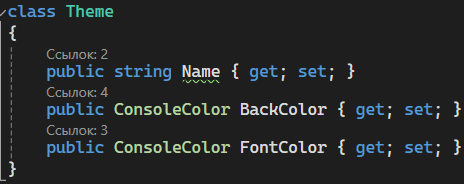




Конечный результат:







* Графический интерфейс пользователя (GUI – Graphic User Interface)

Виды:

1. *Диалоговое окно* – это модальный интерфейс. Требует ответа от пользователя, при открытии блокирует доступ к нижележащим объектам

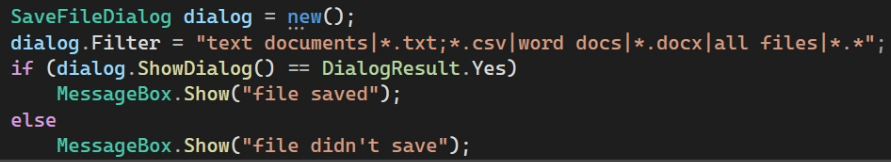
Примеры диалоговых окон:

* MessageBox.Swow(…);

Диалоговое окно с закрытием:



Диалоговое окно с сохранением:



Проверка что пользователь нажал:



1. SDI – Single Document Interface

Примеры: paint, блокнот

1. MDI – Multi Document Interfaсe

Примеры: фотошоп, VS

Создание:

var form = new();

form.**MdiParent** = this;

form.Show(); //элементы открываются внутри родительского окна или в **MdiParent**

**IsMdiContainer** - озволяет открывать дочерние окна в текущем окне.

1. **TDI** – Tabbed Document Interfaсe (многостраничный интерфейс)

Разработка … на С#

**WPF** (Windows Presentation Foundation) позволяет создавать интерфейсы, используя возможности видеокарт. Отделяет верткул от логики

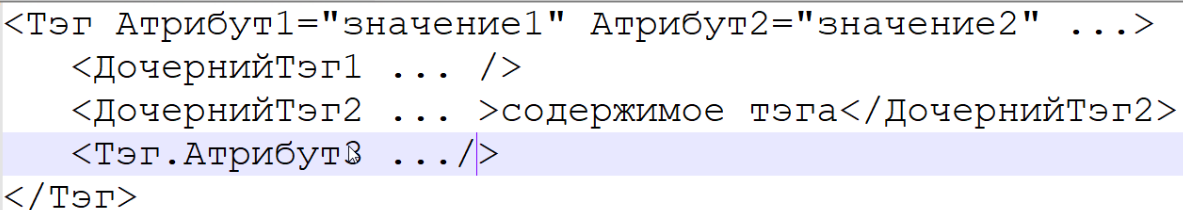
**XAML** (XML)

Или

*Avalonia* – аналог вэкпэк

MAUI позволяет создавать мобильные приложения

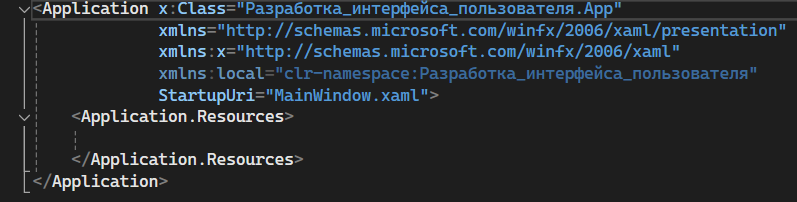
Общая форма разметки XAML:



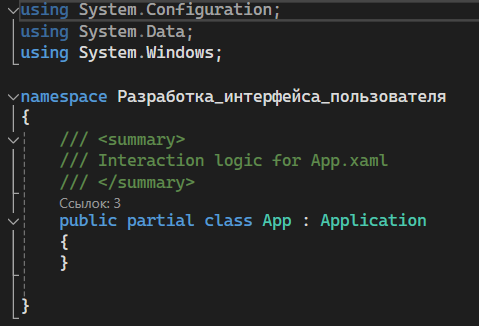
Открываем:



App.xaml (это настройка всего приложения):



App.xaml.cs\*:

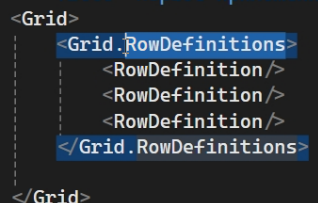


Настройка названия, размеров и значений:



**Компоновка элементов управления**

1. *Grid* – это сетка

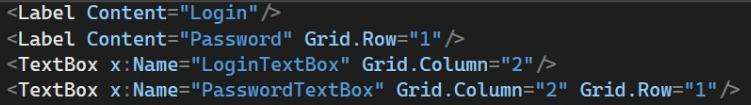


*RowDefinitions* – описание строк.



В **Content** может быть что угодно.

Строчки нумеруются с нуля.



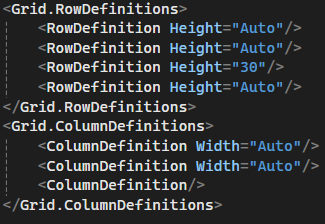
*Grid.Row* – перемещение по строкам.

*Grid.Column* – перемещение по столбцам.



*Grid.ColumnSpan* – Объединение столбцов.

Настройка размеров строк и столбцов:

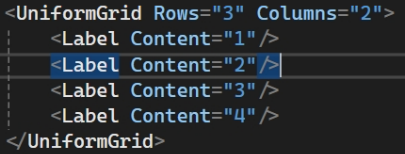


Для динамического изменение размера есть элемент 

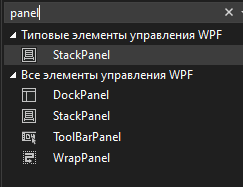
Пример вертикального разделителя:



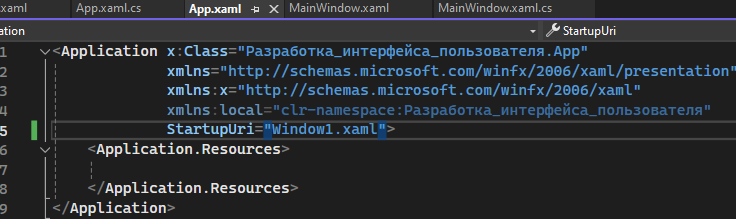
1. *UniformGrid* – это сетка, у которой одинаковое количество строк и столбцов.



Панели:



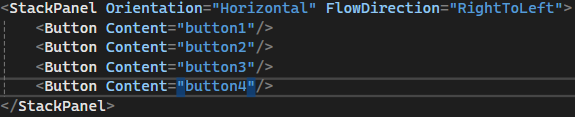
Добавляем новое окно и меняем:



1. *StackPanel* – размещение элементов друг за другом (по умолчанию: по вертикали).

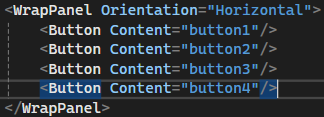
Изменение ориентации:

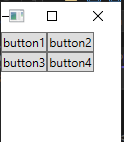




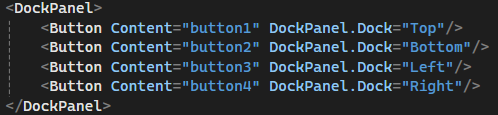


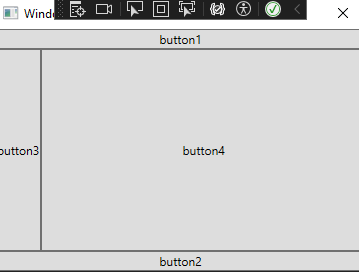
1. WrapPanel похоже на StackPanel, но если элементы не влезают, то они сдвигаются

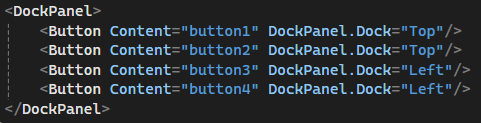


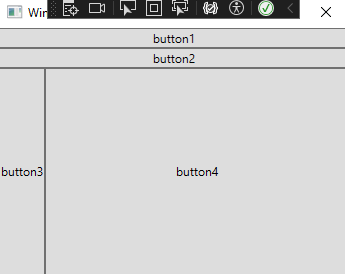


1. DockPanel позволяет прикрепить содержимое к требуемую краю (по умолчанию: слева на право).

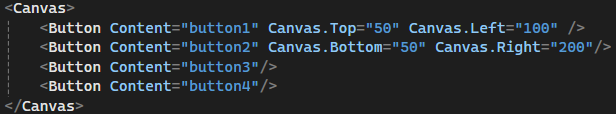


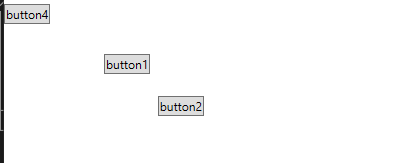


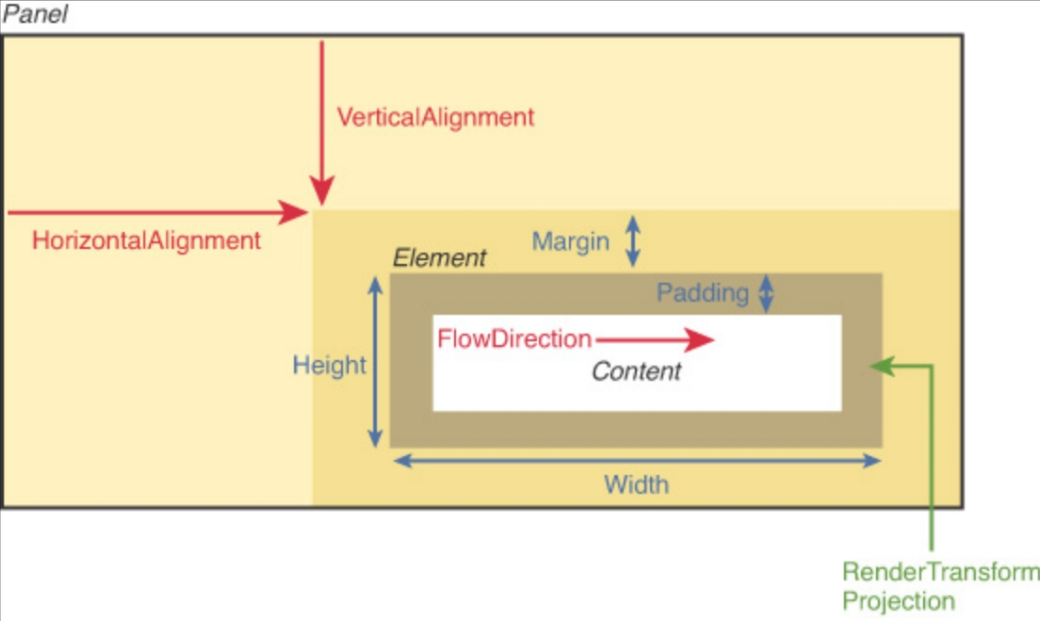


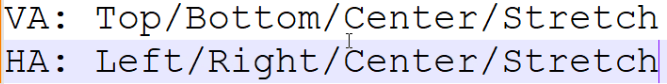


1. *Canvas* – это хост. Для размещения элементов указываются координаты (Координаты по умолчанию 0;0).



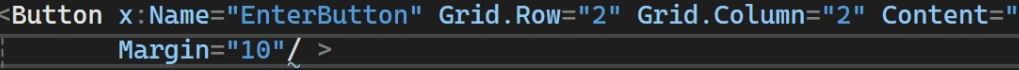


🡨 LayoutTransform

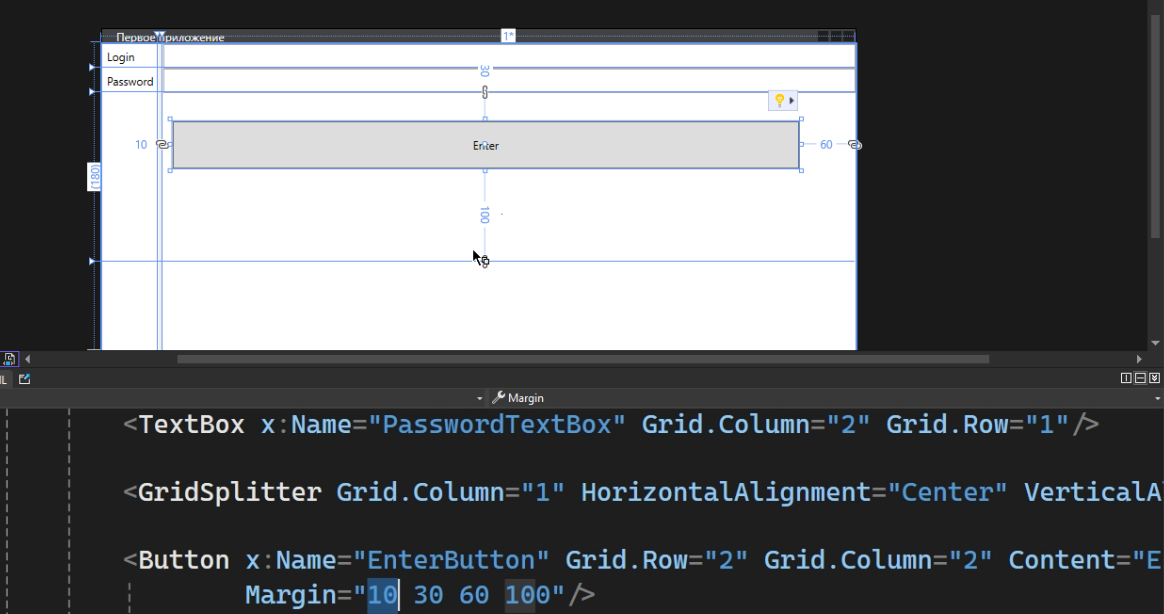


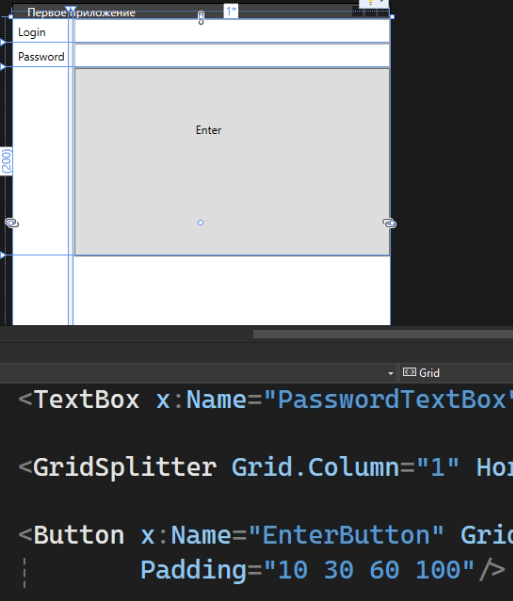
*Margin* – это внешний отступ. В ширину и высоту не включается.

*Padding* – это внутренний отступ.



Первый отступ справа\слева, второй отступ сверху\снизу.



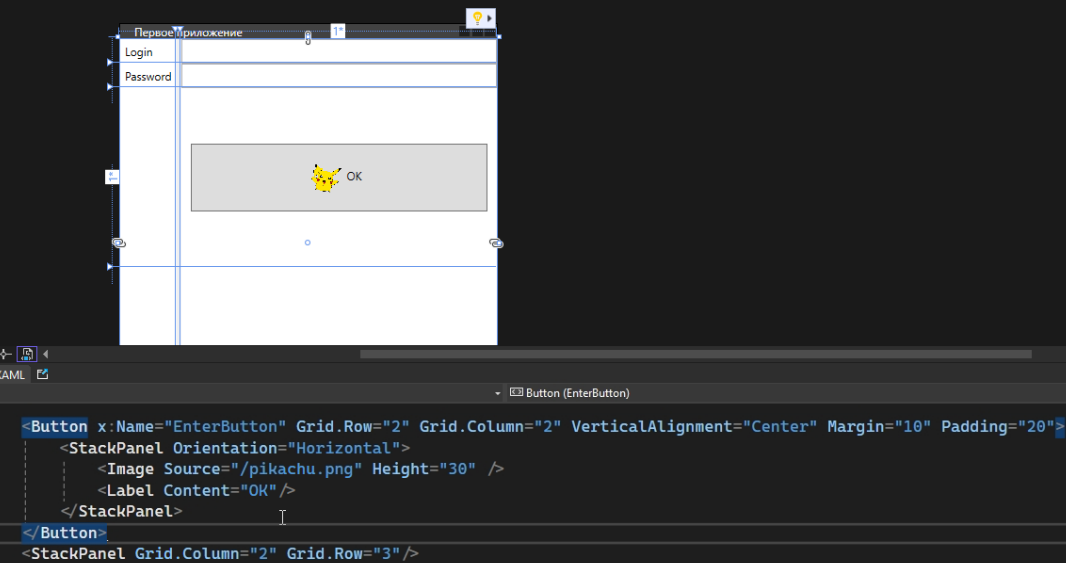


Минимум **Height** и **Width**:



**Content** поддерживается у элементов, внутри которых может быть один дочерний.

*Image* - картинка



*FlowDirection* – направление визуализации (по умолчанию: слева на право).

RenderTransform LayoutTransform и позволяют настроить повороты, искривления, масштабирование и т.д.

LayoutTransform учувствует в компоновке, а RenderTransform не участвует.

*Visibility* – это видимость.

*Collapsed* – элемент скрыт и не участвует в компоновке.

*Hidden* – элемент не отображается и участвует в компоновке

*Visibility* – элемент отображается и участвует в компоновке