**Событийно управляемая тригонометрия**

event

Виды событий WPF:

1. Мышь
2. Клавиатура
3. Стилус
4. Сенсорного экрана
5. Жизненного цикла

Пример события:

* Closing

Свойства: e.Cancel = true; //позволяет отменить закрытие

* Initialized срабатывает при установке всех свойств.
* *Closed* – окно закрыто.
* *Loaded* – загрузка. После инициализации, применение стилей и привязки данных.
* Unloaded работает после закрытия и выгрузки ресурсов.

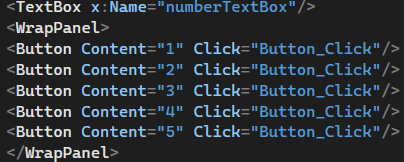
Для перехода к обработчику из разметки нажимаем на обработчик + F12

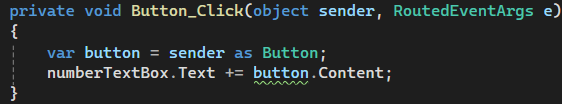
или

ПКМ => Перейти к определению

Можно привязать разные элементы к одному обработчику.

Пример:



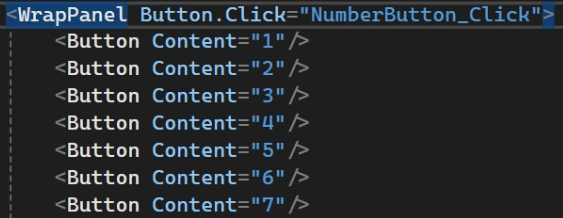


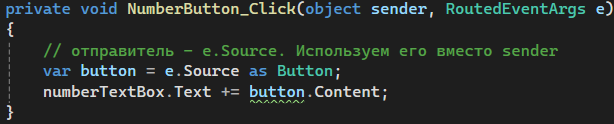
Можно указать обработчик в элементе контейнера для упрощения.

Общая форма:

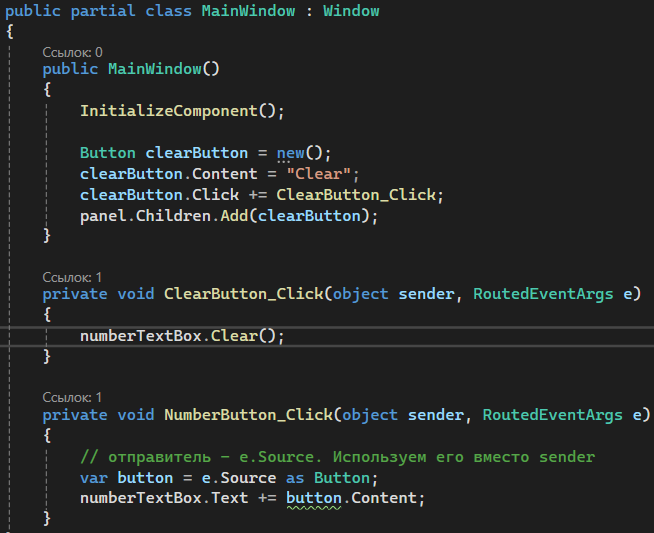


Изменённый пример:





Программное создание и подписка на событие:



**Маршрутизация событий**

События могут обрабатываться по дереву событий.



Виды маршрутизированных событий:

1. Прямые. Возникают и обрабатываются на одном элементе.
2. Опускающиеся/Туннельные. Обработка начинается в главном элементе и идет по вложенному, пока не достигнет вызвавшего его элемента.
3. Поднимающиеся/Пузырьковые. Сначала обрабатываются в вызвавшем элементе и передаются дальше к контейнеру и родительскому для него.

Особенности маршрутизированных событий:

* Принимают параметры RoutedEventAtgs.
* Опускающиеся и поднимающиеся события часто используются в паре.

Как описываются:

*ИмяСобытия* – поднимающиеся

*PreviewИмяСобытия* – опускающиеся

Переменые типа RoutedEventArgs содержат свойства:

1. *Source* – это элемент источника событий
2. *OriginalSource* – это обычно то же что и Source
3. *RoutedEvent* – это имя текущего события
4. *Handled*. При установке true событие будет обрабатываться в источнике без подъема и опускания

События клавиатуры:

1. KeyDown/PreviewKeyDown (PKD)

Возникает при нажатии.

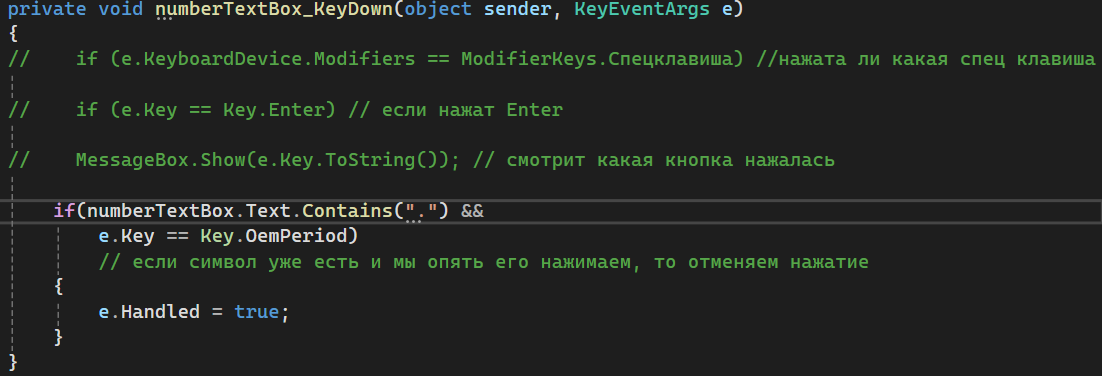
1. KeyUp/PreviewKeyUp (PKU)

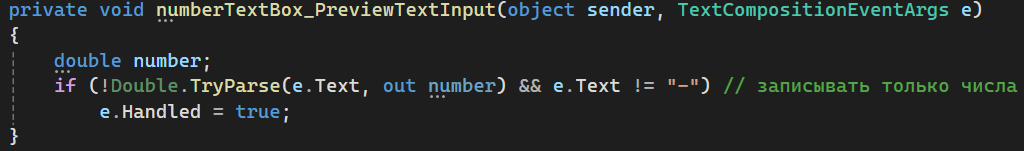
Отжали клавишу.

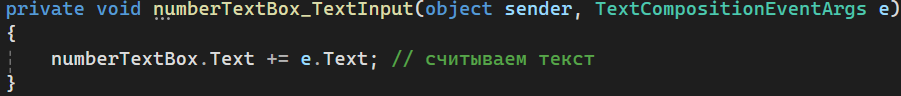
1. TextInput/PreviewTextInput (PTI)

Возникает при получении элемента текстового ввода.

KeyDown и KeyUp принимают KeyEventArgs









События мыши и фокуса:

1. MouseLeftButtonDown/ MouseRigthButtonDown
2. MouseDown/MouseUp
3. MouseMove (когда двигаем мышкой)
4. MouseWheel (колесико)

Не парные события:

1. MouseEnter (зашли на зону элемента)
2. MouseLeave (курсор вышел за пределы элемента)
3. GotMouseCapture/ LostMouseCapture (при получении и потери фокуса)

У элементов можно указать IsHitTestVisible, чтобы элемент не генерировал событие.

Свойства и методы доступные при обработке событий:

**** - нажата или отпущена кнопка

 возвращает нажатую кнопку

 - количество нажатий

 - это положение мыши