## ProgressBar и Slider

ProgressBar и Slider представляют элементы, основанные на диапазонах значений. То есть они хранят и отображают числовые данные на определенном диапазоне.

Все они являются наследниками класса **RangeBase**, поэтому наследуют такие его свойства, как:

* **Value** указывает на текущее значение элемента (представлено типом Double)
* **Maximum/Minimum** указывает на максимальное/минимальное значение элемента

### Slider

Представляет собой обычный ползунок. Он добавляет следующие свойства:

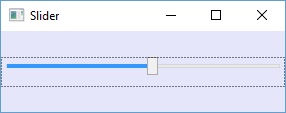
* **Orientation**: указывает ориентацию ползунка - горизонтальную(Horizontal) или вертикальную (Vertical)
* **Delay**: указывает время в миллисекундах, по истечении которого ползунок переместится на одну единицу после щелчка.
* **Interval**: указывает время в миллисекундах, по истечении которого ползунок может перемещаться
* **TickPlacement**: задает визуализацию шкалы ползунка. По умолчанию имеет значение None (отсутствие шкалы). Значение BottomRight создают шкалу в нижней части ползунка, TopLeft - в верхней, Both - по обоим сторонам.
* **TickFrequency**: указывает частоту появления отметок на шкале ползунка.
* **IsSelectionRangeEnabled**: задает затенение участка ползунка. Если оно установлено в True, то начальная и конечная отметка затенения задаются с помощью свойств **SelectionStart** и **SelectionEnd**.

Простейший слайдер:

|  |  |
| --- | --- |
| 1  2 | <Slider Value="0" Minimum="0" Maximum="100" SelectionStart="0" Height="30"          IsSelectionRangeEnabled="True" ValueChanged="Slider\_ValueChanged" /> |

В файле кода мы можем прописать обработчик Slider\_ValueChanged, который будет срабатывать при возникновении события ValueChanged - изменении значения слайдера. В данном случае обработчик этого события будет изменять выделение слайдера:

|  |  |
| --- | --- |
| 1  2  3  4 | private void Slider\_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)  {      ((Slider)sender).SelectionEnd=e.NewValue;  } |



### ProgressBar

ProgressBar представляет индикатор, отображающий выполнение определенного процесса. Также имеет свойство **Orientation**, которое устанавливает вертикальное или горизонтальное расположение индикатора. Для связи с реальными процессами вся логика изменения индикатора, его свойства Value должна отрабатываться в коде.

|  |  |
| --- | --- |
| 1 | <ProgressBar Minimum="0" Maximum="100" Value="50" Height="25" /> |

