

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ



Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Методические указания к лабораторной работе №10

«Использование фигур в WPF-приложениях»

по дисциплине «Технологии разработки пользовательских интерфейсов»

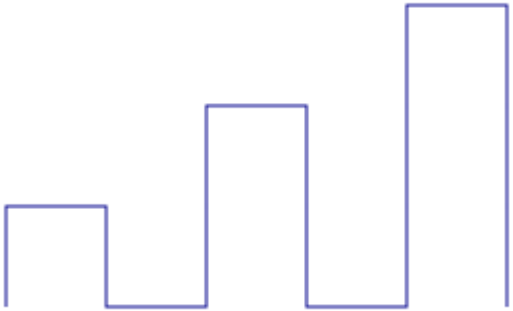
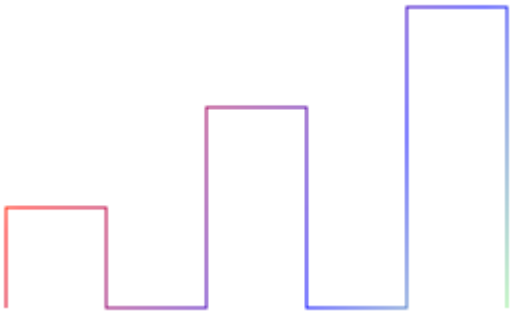
Вичугов В.Н., доцент каф. АиКС

Томск 2012


Фигуры в WPF-приложениях используются для рисования двумерных графических примитивов: линий (Line, Polyline), эллипсов (Ellipse), прямоугольников (Rectangle) и многоугольников (Polygon). Классы фигур являются наследниками абстрактного класса System.Windows.Shapes.Shape, который является наследником класса FrameworkElement. Таким образом, фигуры обрабатываются как обычные элементы пользовательского интерфейса, поддерживают те же основные события, что и другие элементы.

В классе Shape определены следующие базовые свойства фигур:

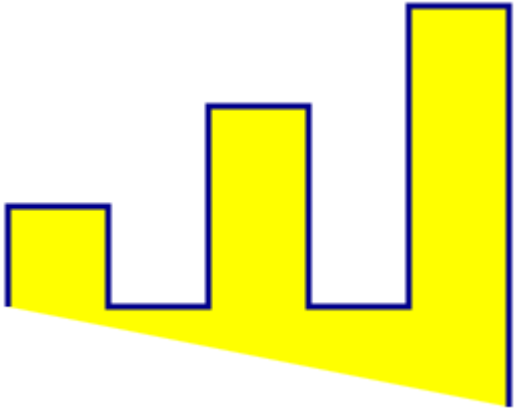
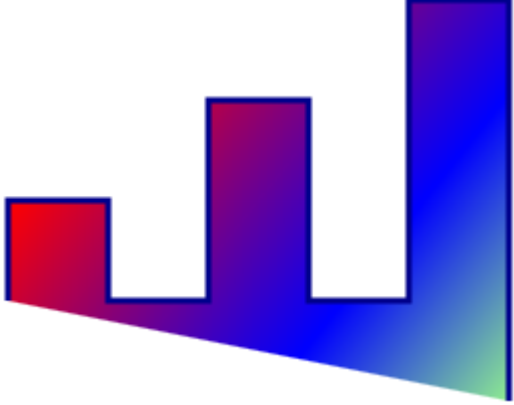
Stroke – кисть для рисования границы фигуры;

<code>Stroke="DarkBlue"</code>	
<pre><Polyline.Stroke> <LinearGradientBrush> <GradientStop Color="Yellow" Offset="0.0" /> <GradientStop Color="Red" Offset="0.2" /> <GradientStop Color="Blue" Offset="0.7" /> <GradientStop Color="LightGreen" Offset="1.0" /> </LinearGradientBrush> </Polyline.Stroke></pre>	

StrokeThickness – толщина границы фигуры;

<code>StrokeThickness="5"</code>	
----------------------------------	--

Fill – кисть для рисования поверхности фигуры;

<pre>Fill="Yellow"</pre>	
<pre><Polyline.Fill> <LinearGradientBrush> <GradientStop Color="Yellow" Offset="0.0" /> <GradientStop Color="Red" Offset="0.2" /> <GradientStop Color="Blue" Offset="0.7" /> <GradientStop Color="LightGreen" Offset="1.0" /> </LinearGradientBrush> </Polyline.Fill></pre>	

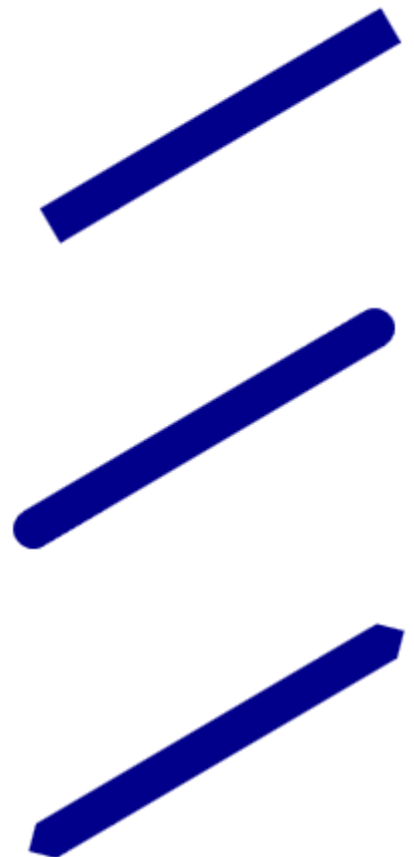
StrokeStartLineCap, StrokeEndLineCap – контуры краев в начале и в конце линии для незамкнутых фигур; возможные значения:

Flat (значение по умолчанию) – четкая граница;

Round – скругленная граница;

Triangle – треугольная граница;

Square – четкая граница (отличается от Flat тем, что линия увеличивается на половину толщины);





StrokeLineJoin – контур углов фигуры; возможные значения:
Miter (значение по умолчанию) – четкие грани углов;



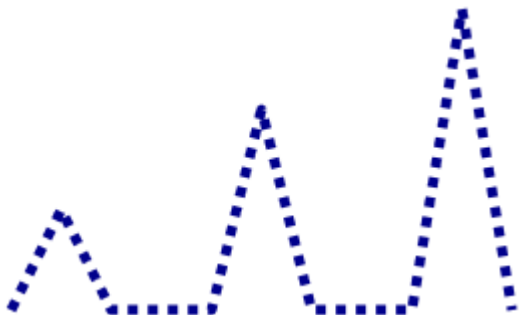



Bevel – угол обрезается в точке сопряжения;



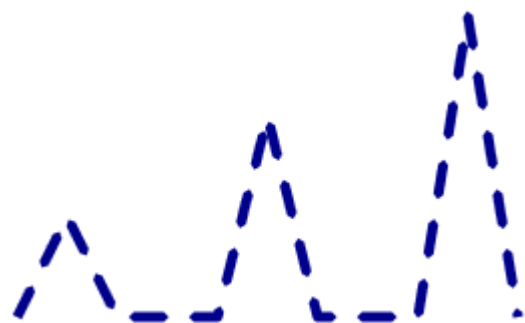
Round – скругление углов;



StrokeDashArray – шаблон для определения вида пунктирной линии границы фигуры; определяется в виде набора чисел, разделенных пробелом или запятой. Каждое значение, умноженное на ширину линии, определяет длину либо сплошного сегмента, либо промежутка между сегментами. Если количество чисел нечетное, то каждое значение используется попеременно то для сплошного сегмента, то для промежутка. Если количество чисел в шаблоне четное, то каждое число, стоящее на нечетной позиции, определяет длину сплошного сегмента, а каждое число, стоящее на четной позиции, определяет длину промежутка между сегментами.

StrokeDashArray="1"	
StrokeDashArray="3"	
StrokeDashArray="5,1"	
StrokeDashArray="1,4,2.5,1"	

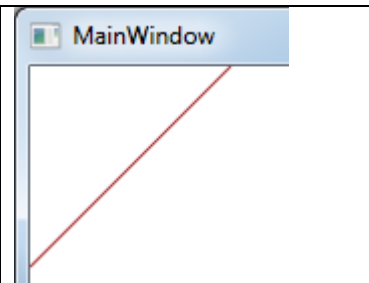
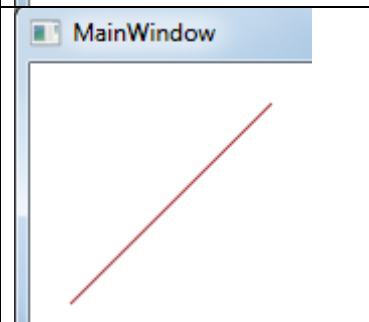
StrokeDashCap – контуры краев пунктирной линии границы фигуры; возможные значения: Flat, Round, Triangle, Square (такие же, как и у свойств StrokeStartLineCap и StrokeEndLineCap).



Линия Line

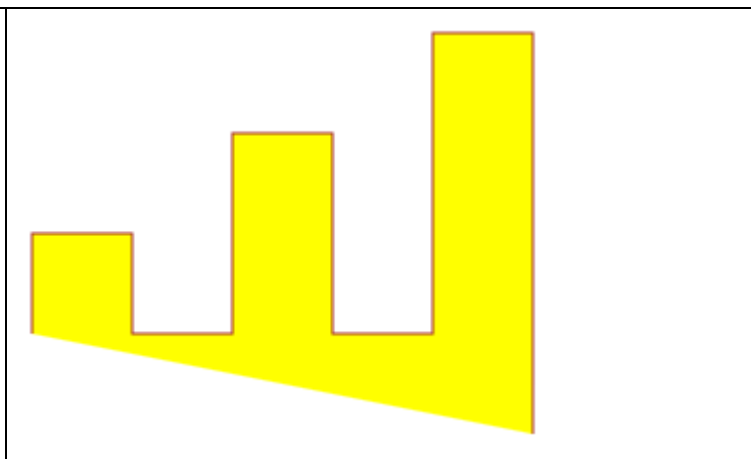
Фигура Line представляет отрезок прямой, соединяющий две точки. Начальная и конечная точки задаются свойствами X1 и Y1 (для первой точки) и X2 и Y2 (для второй точки). Координаты отсчитываются относительно верхнего левого угла элемента Line. Свойство Fill для линии не

используется. Ширина и высота элемента Line определяется автоматически в соответствии со значениями свойств X1, Y1, X2, Y2.

<pre><Canvas> <Line X1="0" Y1="100" X2="100" Y2="0" Stroke="Brown" /> </Canvas></pre>	
<pre><Canvas> <Line Canvas.Top="20" Canvas.Left="20" X1="0" Y1="100" X2="100" Y2="0" Stroke="Brown" /> </Canvas></pre>	

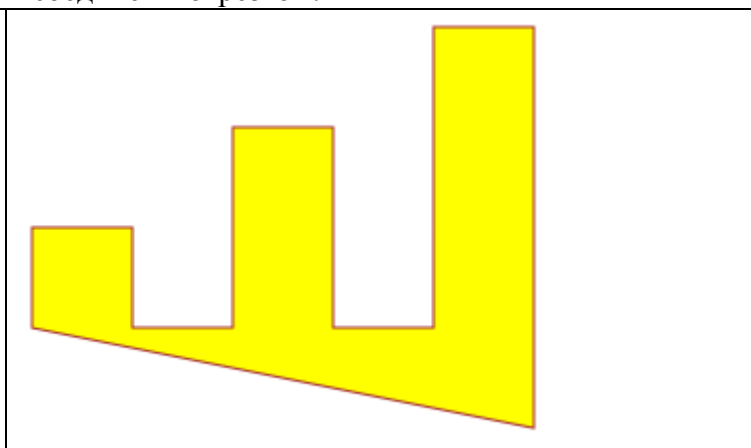
Ломаная линия *Polyline*

Фигура Polyline представляет последовательность связанных отрезков. В свойстве Points указываются координаты вершин, разделенные пробелом или запятой.

<pre><Polyline Stroke="Brown" Fill="Yellow" Points="10,200 10,150 60,150 60,200 110,200 110,100 160,100 160,200 210,200 210,50 260,50, 260,250" /></pre>	
--	--

Многоугольник *Polygon*

Фигура Polygon представляет многоугольник, который отличается от ломаной линии Polyline только тем, что начальная и конечная точки ломаной линии соединены отрезком.

<pre><Polygon Stroke="Brown" Fill="Yellow" Points="10,200 10,150 60,150 60,200 110,200 110,100 160,100 160,200 210,200 210,50 260,50, 260,250" /></pre>	
---	--

Элемент Viewbox

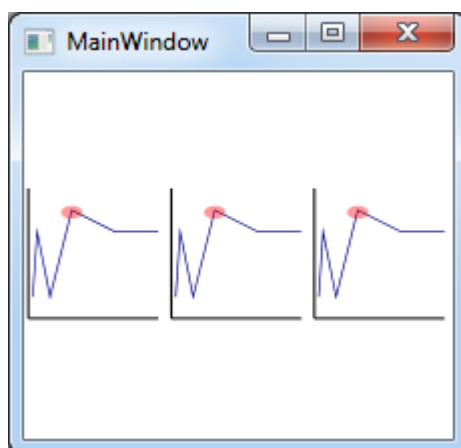
Наиболее распространенным диспетчером компоновки для фигур является Canvas, так как он позволяет совмещать и размещать их произвольным образом. В свою очередь, Canvas помещают в элемент Viewbox, который масштабирует дочерний элемент к своим размерам.

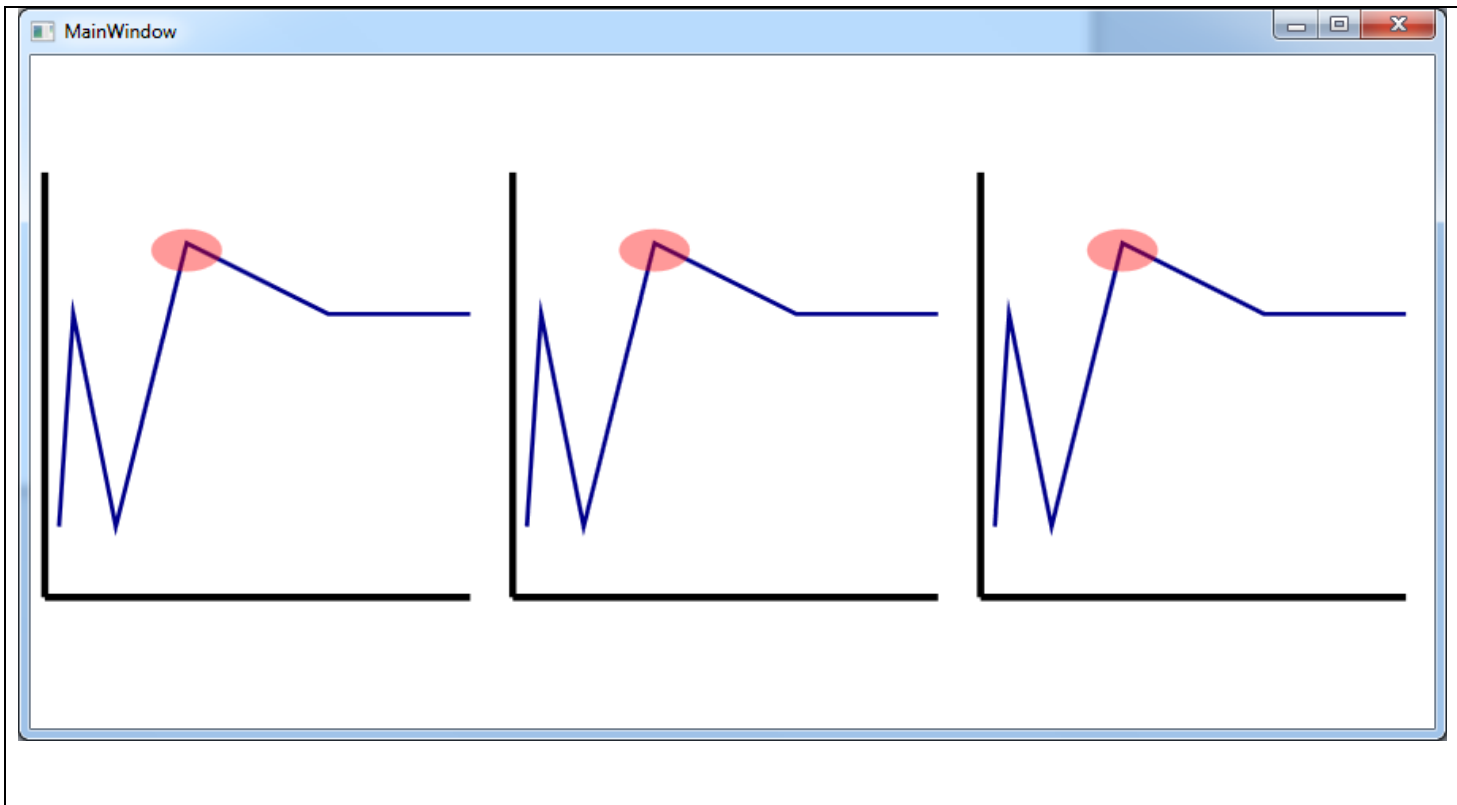
В приведенном ниже примере в таблицу из трех ячеек помещены элементы Viewbox, содержащие диспетчеры компоновки Canvas с фигурами. При изменении размеров окна фигуры масштабируются.

Код XAML

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition />
    <ColumnDefinition />
    <ColumnDefinition />
  </Grid.ColumnDefinitions>
  <Viewbox Grid.Row="0" Grid.Column="0">
    <Canvas Margin="10" Width="310" Height="310">
      <Line Stroke="Black" StrokeThickness="5" X1="0" Y1="0" X2="0" Y2="300" />
      <Line Stroke="Black" StrokeThickness="5" X1="0" Y1="300" X2="300" Y2="300" />
      <Polyline Stroke="DarkBlue" StrokeThickness="3" Points="10,250 20,100 50,250 100,50
200,100 300,100" />
      <Ellipse Width="50" Height="30" Fill="Red" Canvas.Top="40" Canvas.Left="75" Opacity="0.4"
/>
    </Canvas>
  </Viewbox>
  <Viewbox Grid.Row="0" Grid.Column="1">
    ...
  </Viewbox>
  <Viewbox Grid.Row="0" Grid.Column="2">
    ...
  </Viewbox>
</Grid>
```

Результат



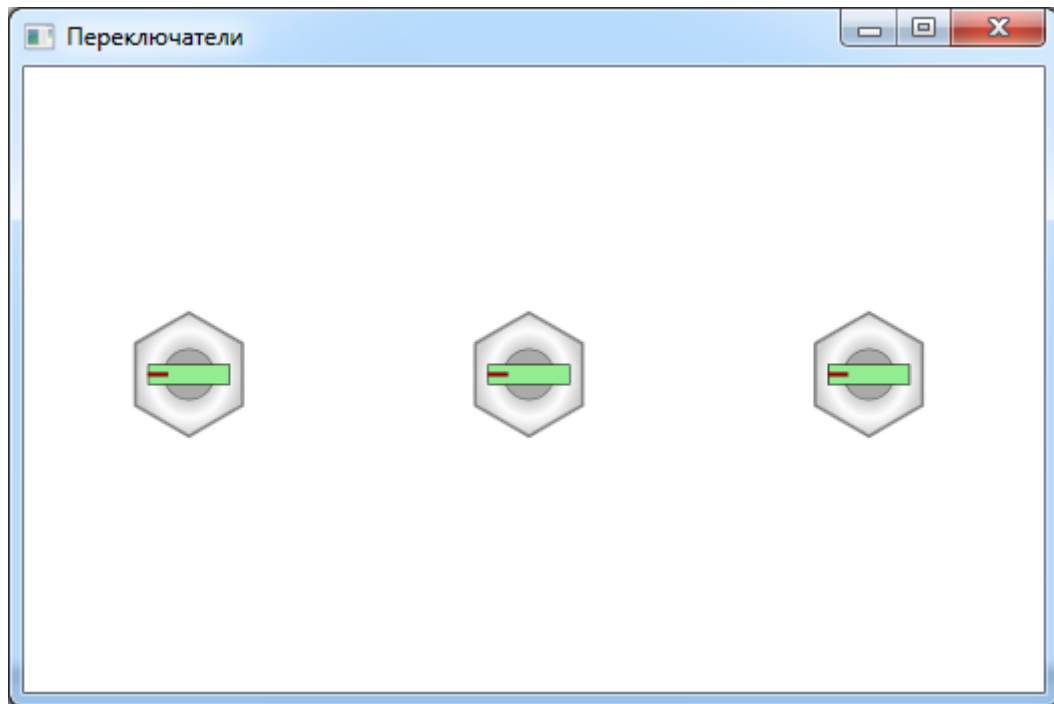


Задание 1

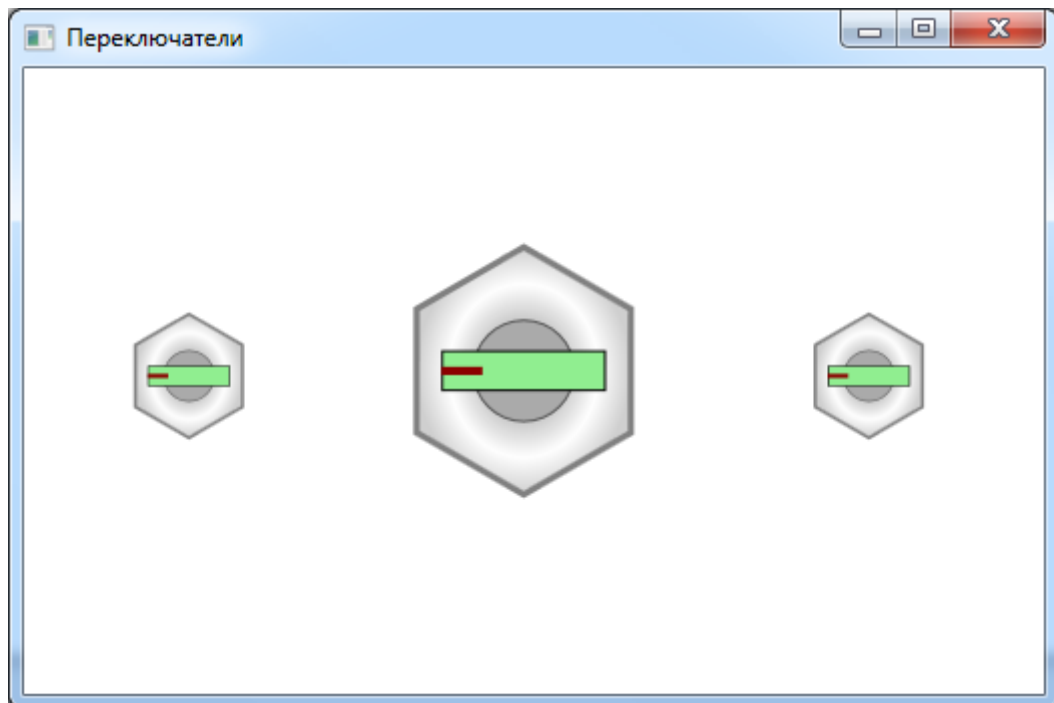
Определите, как изменится поведение программы, если из нее удалить элементы Viewbox и в ячейках таблицы вместо Viewbox разместить сразу Canvas.

Задание 2

Разработайте WPF-приложение «Переключатели». На экране изображены три одинаковых переключателя, которые при изменении размеров окна пропорционально увеличиваются или уменьшаются:



При наведении курсора на переключатель он увеличивается:



При каждом клике по переключателю он **плавно** поворачивается на 20 градусов по часовой стрелке:

