

Класс Path («путь») является наследником от абстрактного класса Shape и позволяет объединять в себе различные геометрические фигуры (линии, прямоугольники, эллипсы, дуги, кривые линии).

Свойство Data класса Path содержит объект-наследник класса Geometry, который определяет геометрическую фигуру. Объект Geometry определяет фигуру, а объект Path рисует её с использованием заданной кисти Stroke и фона Fill.

В свойство Data можно записать объект одного из следующих классов-наследников абстрактного класса Geometry:

**LineGeometry** – прямая линия (объект аналогичен фигуре Line);

**RectangleGeometry** – прямоугольник (объект аналогичен фигуре Rectangle);

**EllipseGeometry** – эллипс (объект аналогичен фигуре Ellipse);

**GeometryGroup** – набор объектов класса Geometry;

**CombinedGeometry** – объединение из двух объектов класса Geometry;

**PathGeometry** – сложные фигуру, состоящие из отрезков прямой, эллиптических дуг, кривых Безье;

**StreamGeometry** – упрощенный вариант задания сложных фигур с помощью мини-языка описания геометрии; объект StreamGeometry не может быть изменен в процессе выполнения программы.

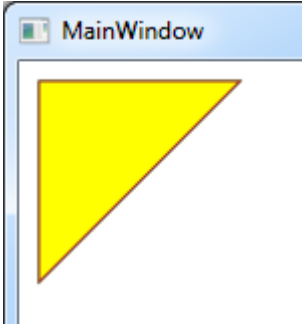
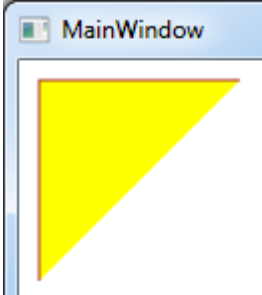
Если в свойстве Data записан текст (например, Data="M10,10 L180,10 180,50 10,80 z"), то на основе этого текста создается объект StreamGeometry, который записывается в свойство Data.

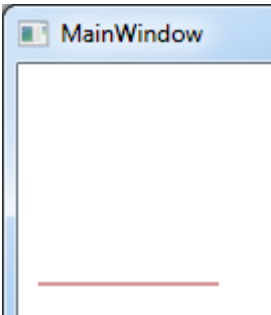
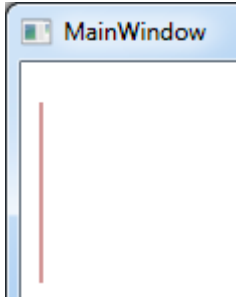


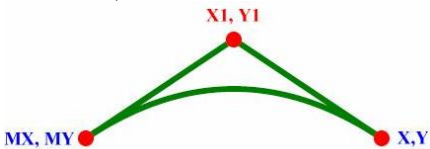
## Классы PathGeometry и StreamGeometry

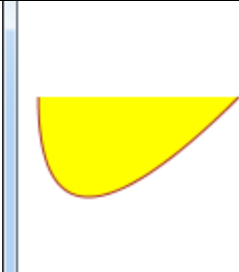
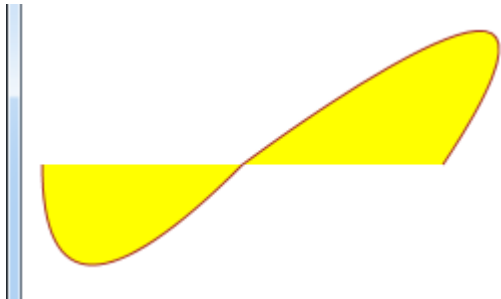

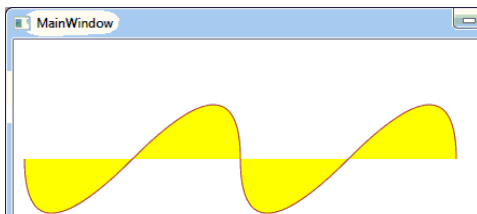
Объект класса PathGeometry может содержать один или несколько объектов класса PathFigure, каждый из которых может содержать один или несколько сегментов. Также данные сегменты могут быть описаны на мини-языке описания геометрии в объекте StreamGeometry.

Виды сегментов:

Описание	Мини-язык	PathGeometry
Начальная точка новой фигуры	<b>M X, Y</b> <b>m dX, dY</b>  (Move) X, Y – абсолютные координаты, dX, dY – смещение относительно последней точки;  Пример: <Path Data="M10,20 ... M3.5,10 ..." />	Свойство StartPoint класса PathGeometry в формате "X,Y" (только абсолютные координаты)  Пример: <Path> <Path.Data> <PathGeometry> <PathFigure StartPoint="10,20"> ... </PathFigure> ... <PathFigure StartPoint="3.5,10"> ... </PathFigure> </PathGeometry> </Path.Data> </Path>
Завершение фигуры. К фигуре	<b>Z</b>	Отрезок, соединяющий

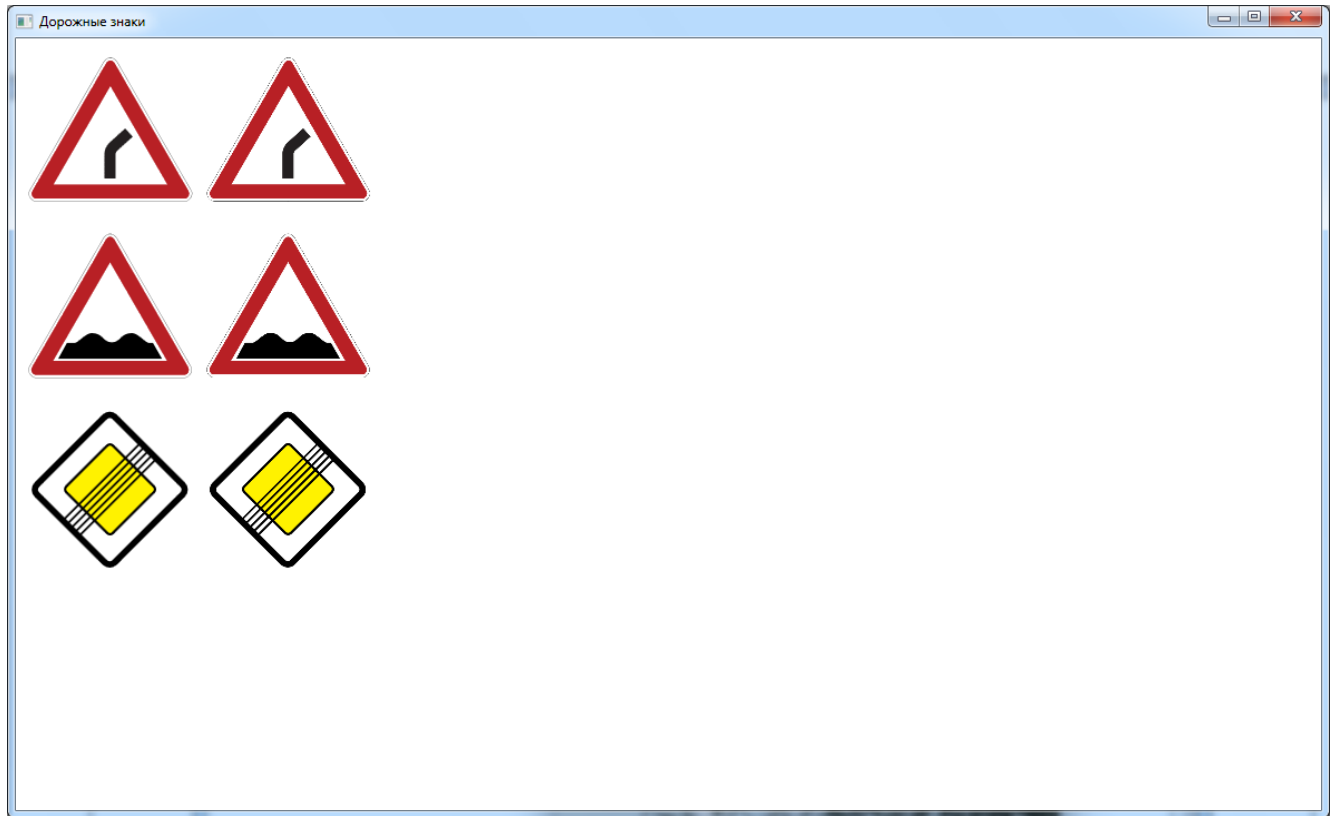
<p>добавляется отрезок, соединяющий текущую точку с начальной.</p>	<p><b>z</b></p> <p>(Close)</p> <p>Пример:</p> <pre>&lt;Path Data="M10,20 ... Z M3.5,10 ..." /&gt;</pre> <p>Первая фигура завершена, вторая – нет.</p>	<p>текущую точку с начальной, добавляется в том случае, если свойство IsClosed=True</p> <p>Пример:</p> <pre>&lt;PathFigure   StartPoint="10,20"   IsClosed="True"&gt;   ... &lt;/PathFigure&gt; &lt;PathFigure   StartPoint="3.5,10"&gt;   ... &lt;/PathFigure&gt;</pre> <p>Первая фигура завершена, вторая – нет.</p>
<p>Отрезок от текущей точки до указанной точки. Для всех сегментов новая точка становится текущей.</p>	<p><b>L X,Y</b> <b>l dx,dy</b></p> <p>(Line)</p> <p>Пример завершенной фигуры:</p> <pre>&lt;Path Stroke="Brown" Fill="Yellow"   Data="M10,110 L10,10 L110,10 Z" /&gt;</pre>  <p>Повторяющиеся буквы L можно не повторять. Пример незавершенной фигуры:</p> <pre>&lt;Path Stroke="Brown" Fill="Yellow"   Data="M10,110 L10,10 110,10" /&gt;</pre>  <p>Относительные координаты:</p> <pre>&lt;Path Stroke="Brown" Fill="Yellow"   Data="M10,110 10,-100 100,0" /&gt;</pre>	<pre>&lt;LineSegment   Point="X,Y" /&gt;</pre> <p>Пример:</p> <pre>&lt;PathFigure   StartPoint="10,110"   IsClosed="True"&gt;   &lt;LineSegment Point="10,10" /&gt;   &lt;LineSegment Point="110,10" /&gt; &lt;/PathFigure&gt;</pre>
<p>Горизонтальный отрезок от текущей точки до указанной координаты.</p>	<p><b>H X</b> <b>h dx</b></p> <p>(Horizontal line)</p> <p>Пример:</p> <pre>&lt;Path Stroke="Brown" Fill="Yellow"</pre>	

	<pre>Data="M10,110 H100" /&gt;</pre>  <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 h90" /&gt;</pre>	
<p>Вертикальный отрезок от текущей точки до указанной координаты.</p>	<p><b>V Y</b> <b>v dY</b></p> <p>(Vertical line) Пример:  <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 V10" /&gt;</pre> </p>  <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 v-90" /&gt;</pre>	
<p>Кубическая кривая Безье от текущей точки (MX, MY) до заданной (X,Y) с начальной контрольной точкой (X1, Y1) и конечной (X2, Y2)</p> 	<p><b>C X1,Y1 X2,Y2 X,Y</b> <b>c dx1,dY1 dx2,dY2 dX,dY</b></p> <p>(Cubic Bezier Curve) Пример:  <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 c0,50 100,-50 100,0" /&gt;</pre> </p> 	<pre>&lt;BezierSegment Point1="X1,Y1" Point2="X2,Y2" Point3="X,Y"/&gt;</pre> <p>Пример:  <pre>&lt;PathFigure StartPoint="10,110"&gt;</pre> <pre>&lt;BezierSegment Point1="10,160" Point2="110,60" Point3="110,110"/&gt;</pre> <pre>&lt;/PathFigure&gt;</pre> </p>
<p>Квадратичная кривая Безье от текущей точки (MX, MY) до заданной (X,Y) с контрольной точкой (X1, Y1)</p> 	<p><b>Q X1,Y1 X,Y</b> <b>q dx1,dY1 dX,dY</b></p> <p>(Quadratic Bezier Curve) Пример:  <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 q0,100 100,0" /&gt;</pre> </p>	<pre>&lt;QuadraticBezierSegment Point1="X1,Y1" Point2="X,Y"/&gt;</pre> <p>Пример:  <pre>&lt;PathFigure StartPoint="10,110"&gt;</pre> <pre>&lt;QuadraticBezierSegment Point1="10,210" Point2="110, 110"/&gt;</pre> </p>

		<code>&lt;/PathFigure&gt;</code>
<p>Кубическая кривая Безье от текущей точки (MX, MY) до заданной (X,Y) с конечной контрольной точкой (X1, Y1). Начальная контрольная точка представляет собой отражение контрольной точки предыдущей команды.</p>	<p><b>S</b> <b>x1,y1 x,y</b></p> <p><b>s</b> <b>dx1,dy1 dx,dy</b></p> <p>(Smooth Cubic Bezier Curve)</p> <p>Пример:</p> <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 q0,100 100,0 s200,-150 100,0" /&gt;</pre> 	
<p>Квадратичная кривая Безье от текущей точки (MX, MY) до заданной (X,Y). Контрольная точка представляет собой отражение контрольной точки предыдущей команды.</p> 	<p><b>T</b> <b>x,y</b></p> <p><b>t</b> <b>dx,dy</b></p> <p>(Smooth Quadratic Bezier Curve)</p> <p>Пример:</p> <pre>&lt;Path Stroke="Brown" Fill="Yellow" Data="M10,110 q0,100 100,0 t100,0 100,0 100,0" /&gt;</pre> <p>Повторяющиеся буквы (в данном случае t) можно не указывать.</p> 	

## Задание 1

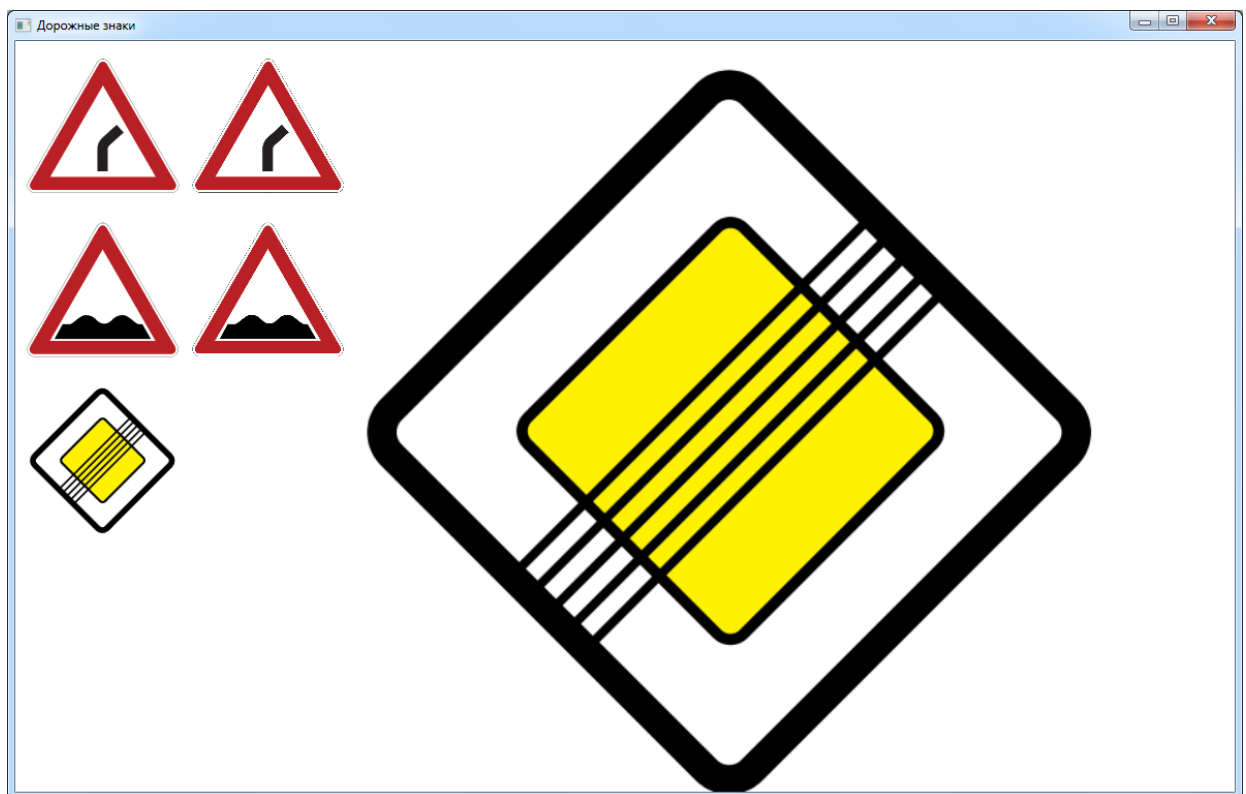
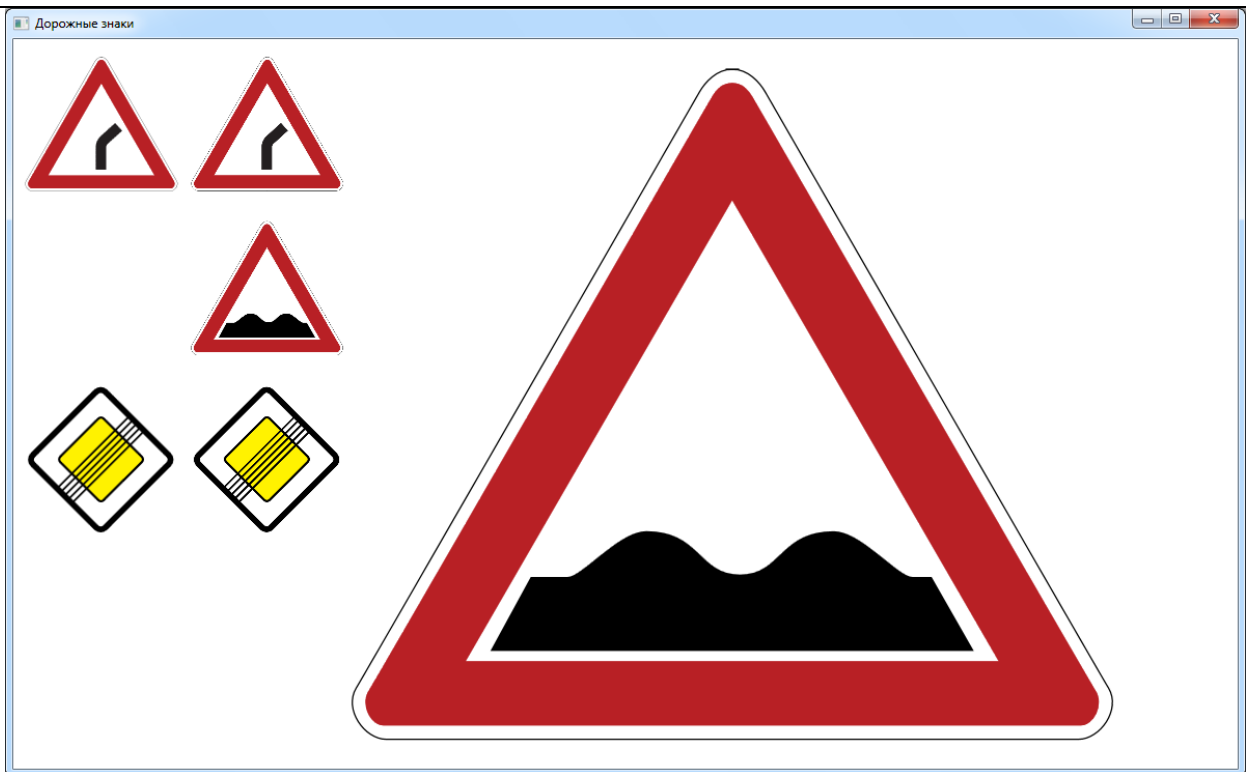
Разработайте WPF-приложение «Дорожные знаки» со следующим интерфейсом:



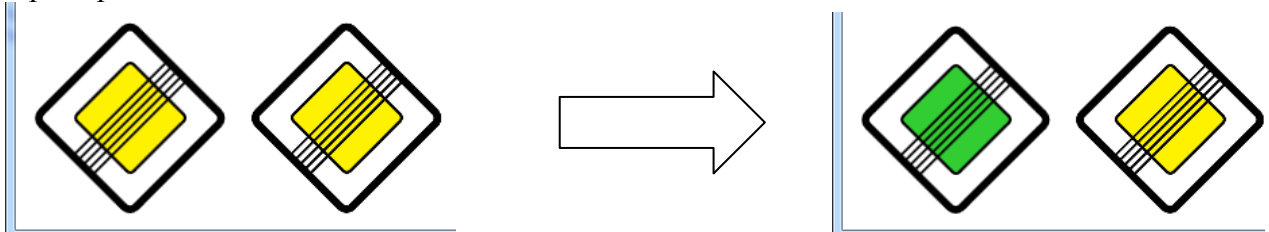
По запросу «Дорожные знаки Российской Федерации» найдите в Википедии соответствующую страницу и выберите на ней три **сложных** знака, **отличных от изображенных в примере**. Обязательно, чтобы при нажатии на знак открывался файл SVG.

В первом столбце знаки изображаются с помощью объектов Path, во втором – с помощью картинок в формате PNG. Конвертацию SVG-файла в XAML- и PNG-файлы можно произвести с помощью программы Inkscape.

При нажатии на любой знак он плавно увеличивается и перемещается в правую часть окна. Через две секунды он плавно возвращается в исходное состояние:



При наведении курсора мыши на какой-то один знак плавно меняется цвет какой-либо части этого знака. Пример:



#### Подсказки:

Для плавного изменения цвета используйте объект `ColorAnimation` и свойство `Fill.(SolidColorBrush.Color)`