**Методические указания к лабораторной работе №12**

# «Классы Path и Geometry»

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 PathFirgure, 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 | **М X,Y**  **m dX, dY**    (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 добавляется отрезок, соединяющий текущую точку с начальной. | **z**    (Close) Пример:  <Path Data="M10,20 ... **Z** M3.5,10..."  />    Первая фигура завершена, вторая – нет. | Отрезок, соединяющий текущую точку с начальной, добавляется в том случае, если свойство IsClosed=True    Пример:  <PathFigure  StartPoint="10,20" **IsClosed="True"**> ...  </PathFigure>  <PathFigure  StartPoint="3.5,10">  ...  </PathFigure>  Первая фигура завершена, вторая – нет. |
| Отрезок от текущей точки до указанной точки. Для всех сегментов новая точка становится текущей. | **L X,Y**  **l dX,dY**    (Line) Пример завершенной фигуры:  <Path Stroke="Brown" Fill="Yellow" Data="M10,110 **L10,10 L110,10** Z" />        Повторяющиеся буквы L можно не повторять. Пример незавершенной фигуры:  <Path Stroke="Brown" Fill="Yellow" Data="M10,110 L10,10 110,10" />        Относительные координаты:  <Path Stroke="Brown" Fill="Yellow" Data="M10,110 l0,-100 100,0" /> | **<LineSegment**  **Point="X,Y" />**      Пример:  <PathFigure  StartPoint="10,110"  IsClosed="True">  **<LineSegment Point="10,10"**  **/>**  **<LineSegment Point="110,10"**  **/>**  </PathFigure> |

|  |  |  |
| --- | --- | --- |
| Горизонтальный отрезок от текущей точки до указанной координаты. | **H X h dX**    (Horizontal line) Пример:  <Path Stroke="Brown" Fill="Yellow" Data="M10,110 **H100**" />    <Path Stroke="Brown" Fill="Yellow"  Data="M10,110 **h90**" /> |  |
| Вертикальный отрезок от текущей точки до указанной координаты. | **V Y v dY**    (Vertical line) Пример:  <Path Stroke="Brown" Fill="Yellow" Data="M10,110 V10" />    <Path Stroke="Brown" Fill="Yellow"  Data="M10,110 v-90" /> |  |
| Кубическая кривая Безье от текущей точки (MX, MY) до заданной (X,Y) с начальной контрольной точкой (X1, Y1) и конечной (X2, Y2) | **С X1,Y1 X2,Y2 X,Y**    **c dX1,dY1 dX2,dY2 dX,dY**    (Cubic Bezier Curve)Пример:  <Path Stroke="Brown" Fill="Yellow"  Data="M10,110 c0,50 100,-50 100,0" /> | **<BezierSegment**  **Point1="X1,Y1"**  **Point2="X2,Y2"**  **Point3="X,Y"/>**      Пример:  <PathFigure  StartPoint="10,110">    **<BezierSegment**  **Point1="10,160"**  **Point2="110,60"**  **Point3="110,110"/>**    </PathFigure> |
| Квадратичная кривая Безье от текущей точки (MX, MY) до заданной (X,Y) с контрольной точкой (X1, Y1) | **Q X1,Y1 X,Y**    **q dX1,dY1 dX,dY**    (Quadratic Bezier Curve)Пример:  <Path Stroke="Brown" Fill="Yellow"  Data="M10,110 q0,100 100,0" /> | **<QuadraticBezierSegment**  **Point1="X1,Y1"**  **Point2="X,Y"/>**        Пример:    <PathFigure  StartPoint="10,110">    **<QuadraticBezierSegment**  **Point1="10,210"**  **Point2="110, 110"/>**  </PathFigure> |
| 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. | **S X1,Y1 X,Y**    **s dX1,dY1 dX,dY**    (Smooth Cubic Bezier Curve)Пример:  <Path Stroke="Brown" Fill="Yellow"  Data="M10,110 q0,100 100,0 s200,-150  100,0" /> |  |
| Квадратичная кривая Безье от текущей точки (MX, MY) до заданной (X,Y). Контрольная точка представляет собой отражение контрольной точки предыдущей команды. | **T X,Y t dX,dY**    (Smooth Quadratic Bezier Curve)Пример:  <Path Stroke="Brown" Fill="Yellow"  Data="M10,110 q0,100 100,0 t100,0  100,0 100,0" />    Повторяющиеся буквы (в данном случае t) можно не указывать. |  |

***За***

***дание 1***

Разработайте

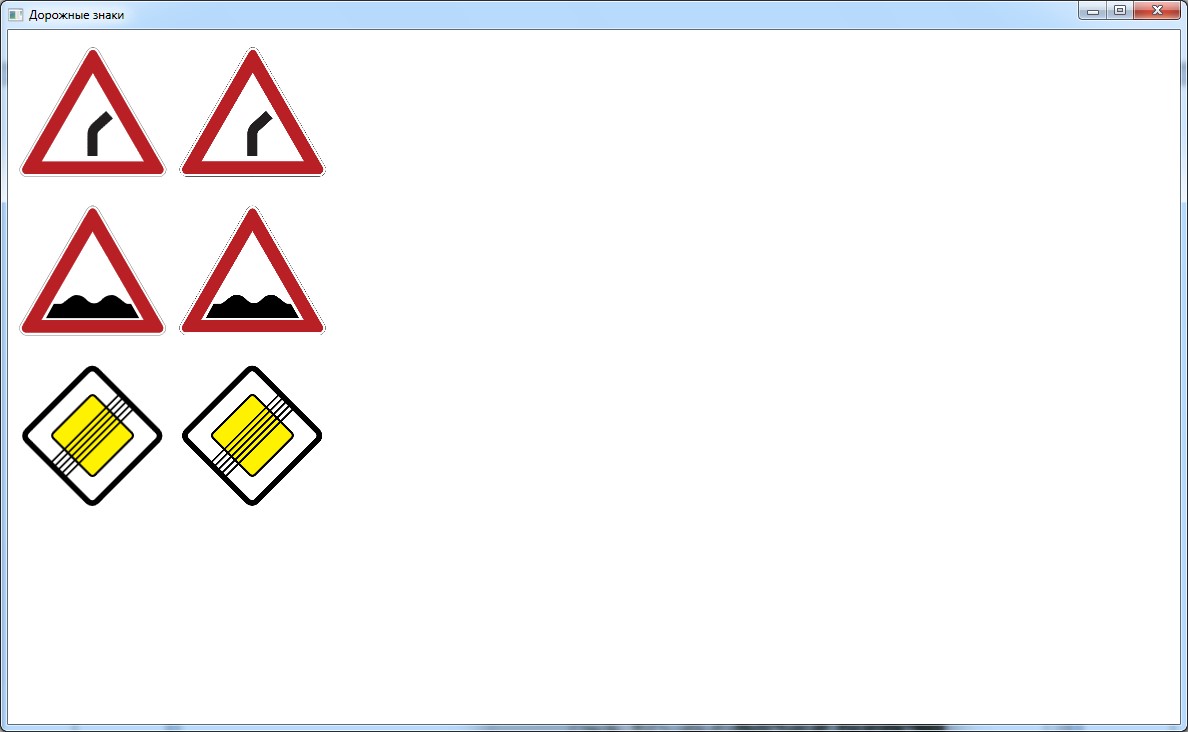
WPF

-

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 по

мощью объектов

Path

, во втором

–

с помощью картинок в

формате

PNG

.

Конвертацию

SVG

-

файла в

XAML

-

и

PNG

-

файлы можно произвести с помощью

программы

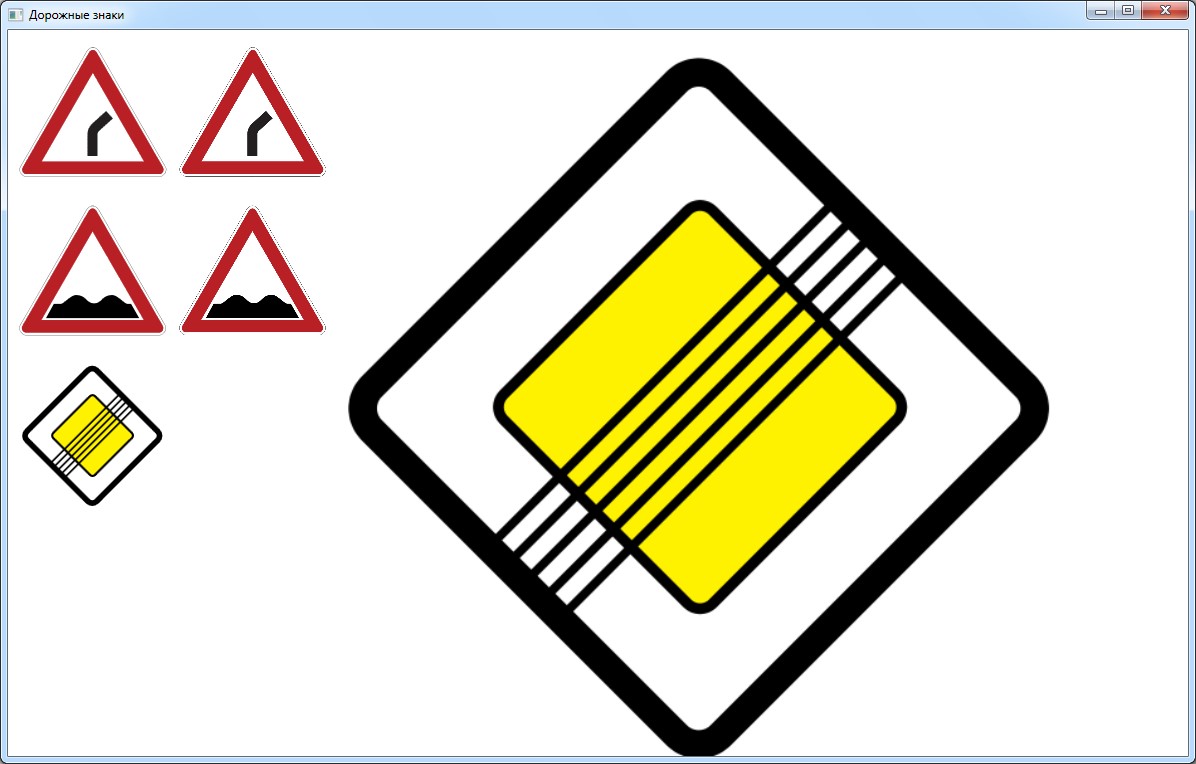
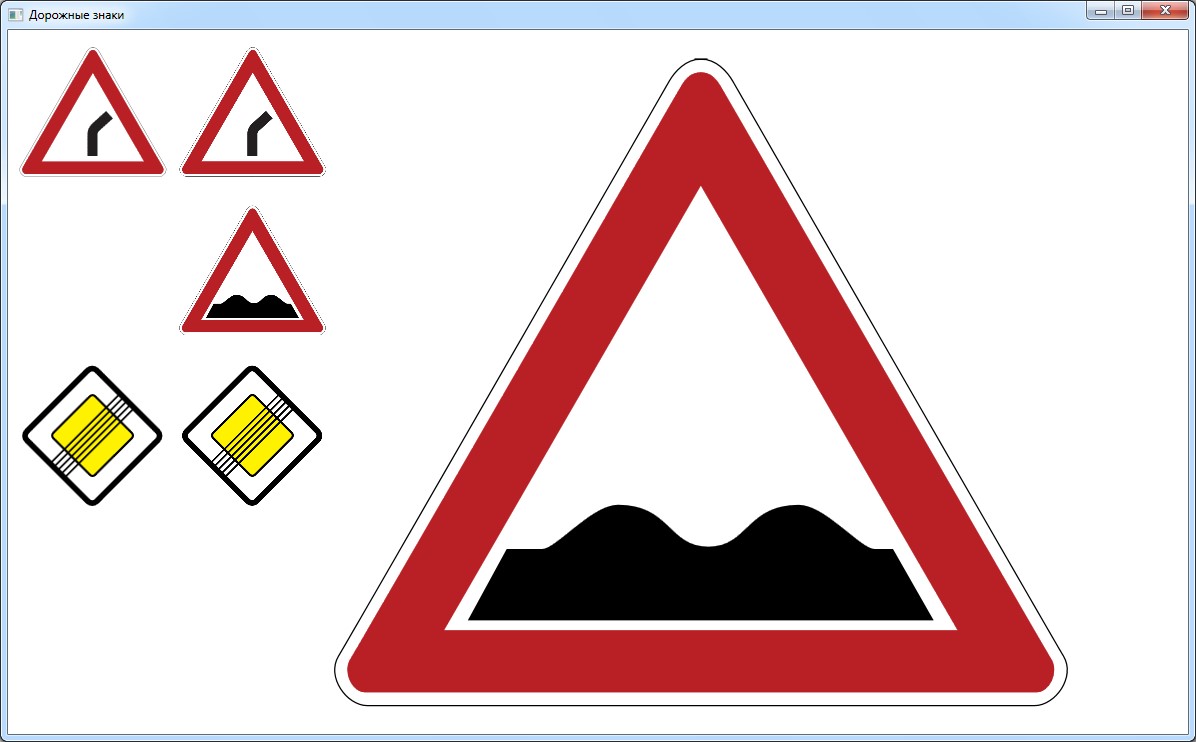
Inkscape

.

При нажатии на любой знак он плавно увеличивается и перемещается в правую часть окна. Через две

сек

унды он плавно возвращается в исходное состояние:



При наведении курсора мыши на какой

-

то

один

знак

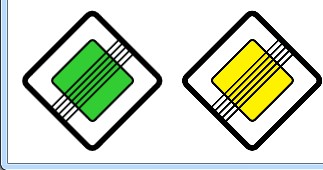
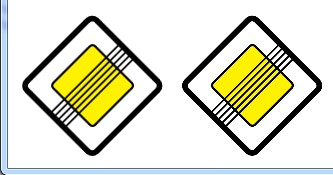
плавно

меняется цвет какой

-

либо части этого

знака. Пример:



**Подсказки:**

Для плавного изменения цвета

используйте объект

ColorAnimation

и свойство

Fill.(SolidColorBrush.Color)