

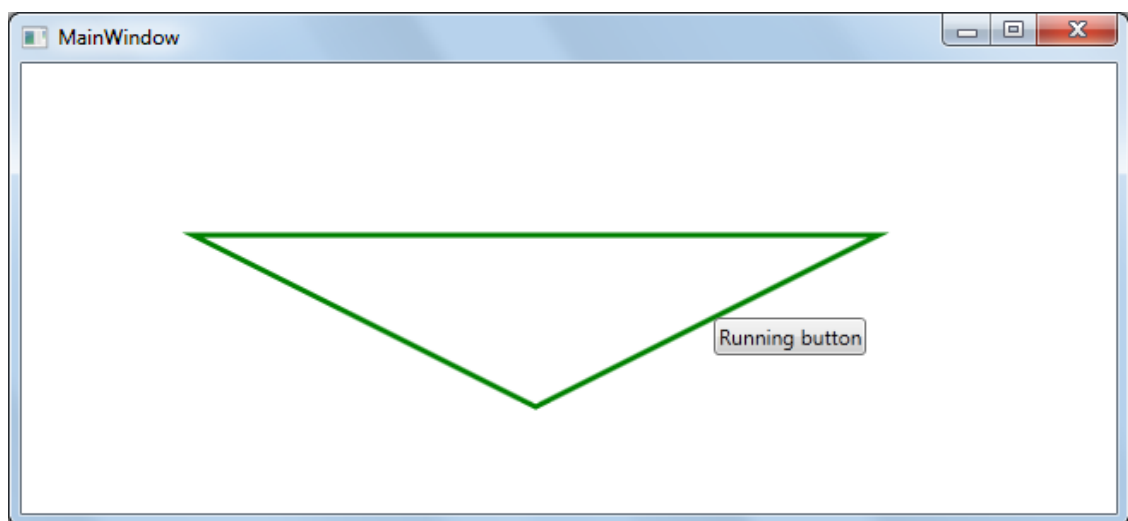
Объект класса PathGeometry может быть использован для анимации свойств по описанному геометрическому пути. Анимацию по геометрическому пути обеспечивают следующие классы:

- DoubleAnimationUsingPath
- PointAnimationUsingPath
- MatrixAnimationUsingPath

В большинстве случаев анимации подвергаются свойства, отвечающие за положение элементов в окне (например, Canvas.Top и Canvas.Left).

Пример приложения с кнопкой, «бегающей» по треугольнику:

```
<Window.Resources>
    <PathGeometry x:Key="PathAnimation">
        <PathFigure StartPoint="100,100" IsClosed="True">
            <LineSegment Point="500,100" />
            <LineSegment Point="300,200" />
        </PathFigure>
    </PathGeometry>
</Window.Resources>
<Canvas>
    <Path Stroke="Green" StrokeThickness="3" Data="{StaticResource ResourceKey=PathAnimation}" />
    <Button Content="Running button">
        <Button.Triggers>
            <EventTrigger RoutedEvent="Loaded">
                <BeginStoryboard>
                    <Storyboard RepeatBehavior="Forever">
                        <DoubleAnimationUsingPath Storyboard.TargetProperty="(Canvas.Left)"
Duration="0:0:3"
PathGeometry="{StaticResource ResourceKey=PathAnimation}"
Source="X" />
                        <DoubleAnimationUsingPath Storyboard.TargetProperty="(Canvas.Top)"
Duration="0:0:3"
PathGeometry="{StaticResource ResourceKey=PathAnimation}"
Source="Y" />
                    </Storyboard>
                </BeginStoryboard>
            </EventTrigger>
        </Button.Triggers>
    </Button>
</Canvas>
```



Геометрический путь описан как ресурс окна и используется три раза:

- 1) для рисования пути Path;
- 2) для анимации свойства Canvas.Left – в качестве источника (Source) используется горизонтальная координата X;

- 3) для анимации свойства Canvas.Top – в качестве источника (Source) используется вертикальная координата Y.

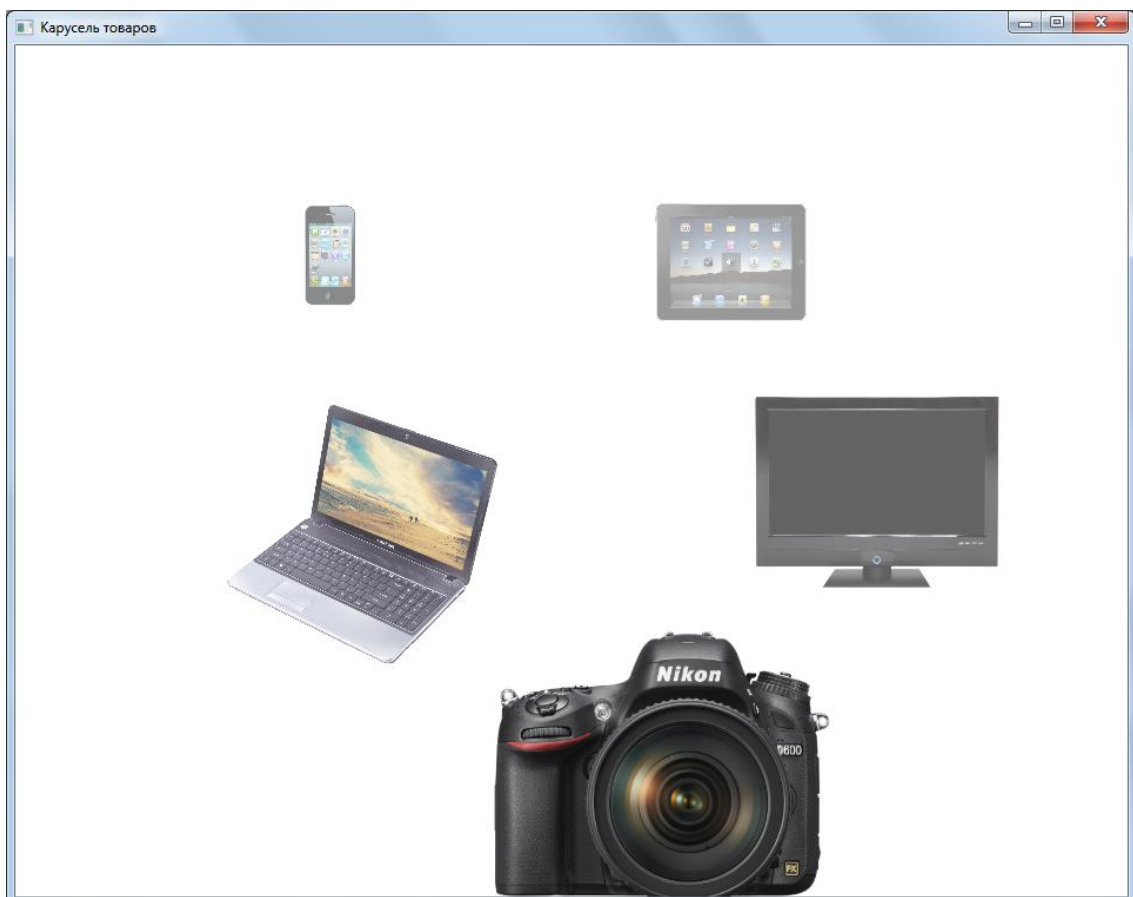
В объекте **DoubleAnimationUsingPath** свойство **PathGeometry** определяет геометрический путь, по которому будет осуществляться анимация, а свойство **Source** содержит либо значение X, если из геометрического пути необходимо выбирать горизонтальные координаты, либо Y, если необходимо выбирать вертикальные координаты.

Задание 1

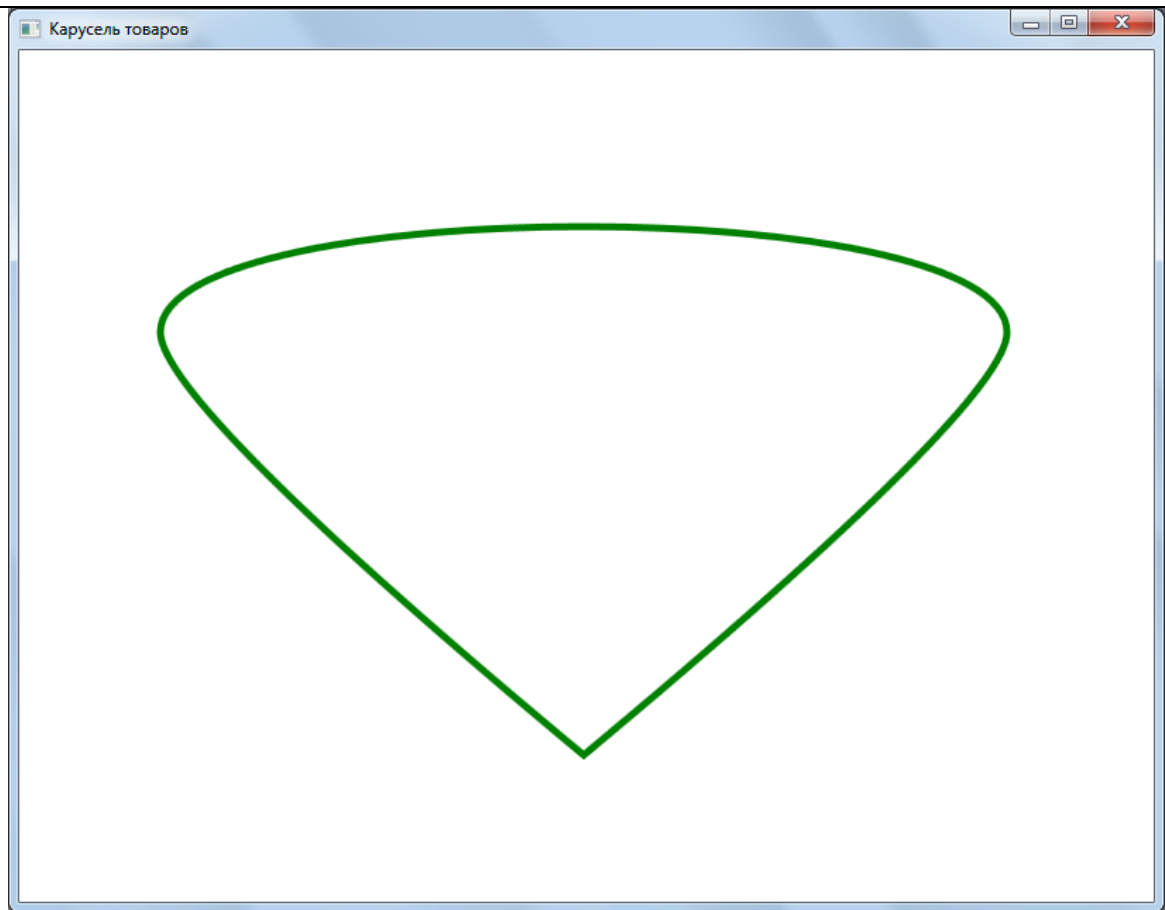
Проверьте, как будет происходить анимация кнопки, если геометрический путь будет содержать несколько фигур, не связанных между собою.

Задание 2

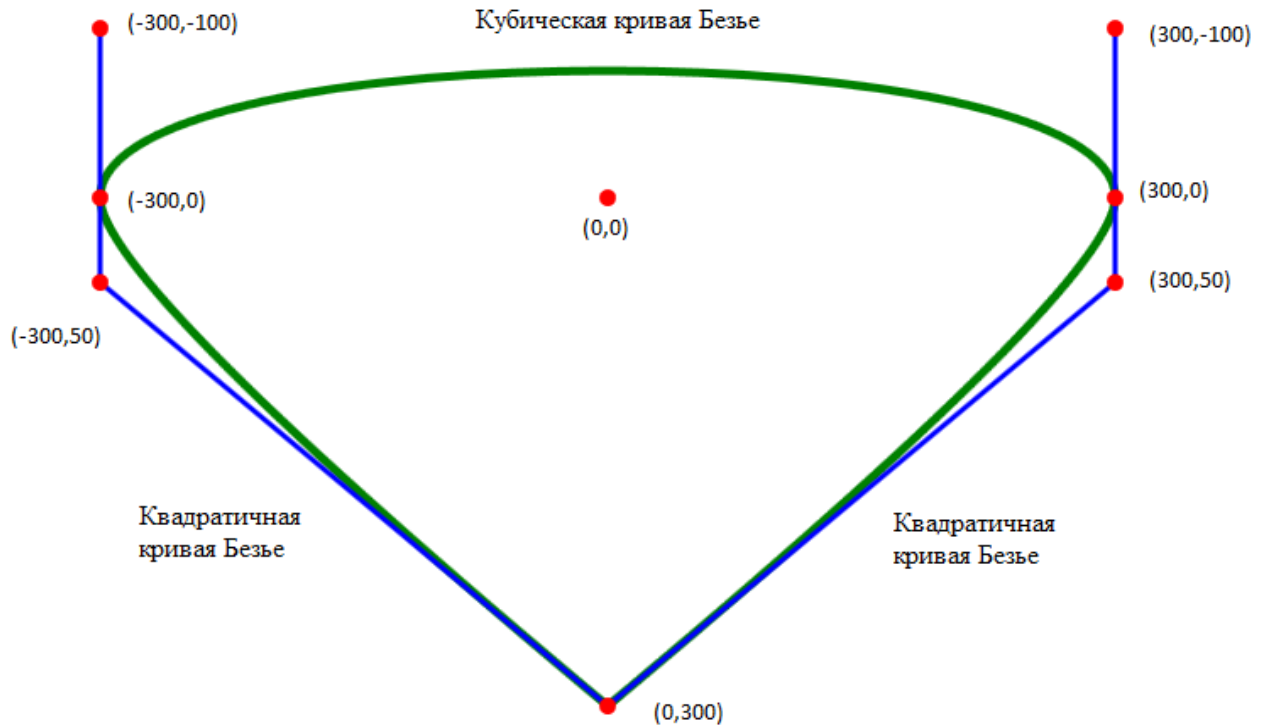
Разработайте WPF-приложение «Карусель товаров» со следующим интерфейсом:



В окне перемещаются PNG-изображения пяти товаров (в Вашем приложении могут быть любые товары) по часовой стрелке по следующему геометрическому пути:



Этот же геометрический путь с контрольными точками и отрезками:



При перемещении изменяется масштаб и прозрачность изображений. В точке $(0, 300)$ изображения становятся полностью непрозрачными с максимальным размером.

Исходный XAML-код должен выглядеть **примерно** следующим образом:

```
<Window ...>
  <Window.Resources>
    ...
  </Window.Resources>
  <Canvas>
```

```
<Viewbox Canvas.Top="-1000" Width="100" Style="{StaticResource ResourceKey=s1}">
    <Image Source="./1.png" />
</Viewbox>

<Viewbox Canvas.Top="-1000" Width="100" Style="{StaticResource ResourceKey=s2}">
    <Image Source="./2.png" />
</Viewbox>

<Viewbox Canvas.Top="-1000" Width="100" Style="{StaticResource ResourceKey=s3}">
    <Image Source="./3.png" />
</Viewbox>

<Viewbox Canvas.Top="-1000" Width="100" Style="{StaticResource ResourceKey=s4}">
    <Image Source="./4.png" />
</Viewbox>

<Viewbox Canvas.Top="-1000" Width="100" Style="{StaticResource ResourceKey=s5}">
    <Image Source="./5.png" />
</Viewbox>
</Canvas>
</Window>
```