**28 Работа с графикой. Трехмерная графика**

Задание 1. Нарисуйте фигуру.

Листинг программы:

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="7\*"/>

<ColumnDefinition Width="93\*"/>

</Grid.ColumnDefinitions>

<Canvas Width="200" Height="200" Grid.Column="1" Margin="244,117,300,117">

<Ellipse Stroke="Black" StrokeThickness="2" Width="200" Height="200" />

<Ellipse Stroke="Black" Width="150" Height="150" Canvas.Left="25" Canvas.Top="25" />

<Ellipse Stroke="Black" Width="90" Height="90" Canvas.Left="55" Canvas.Top="55" RenderTransformOrigin="0.304,0.319" />

</Canvas> </Grid>

Анализ результатов:

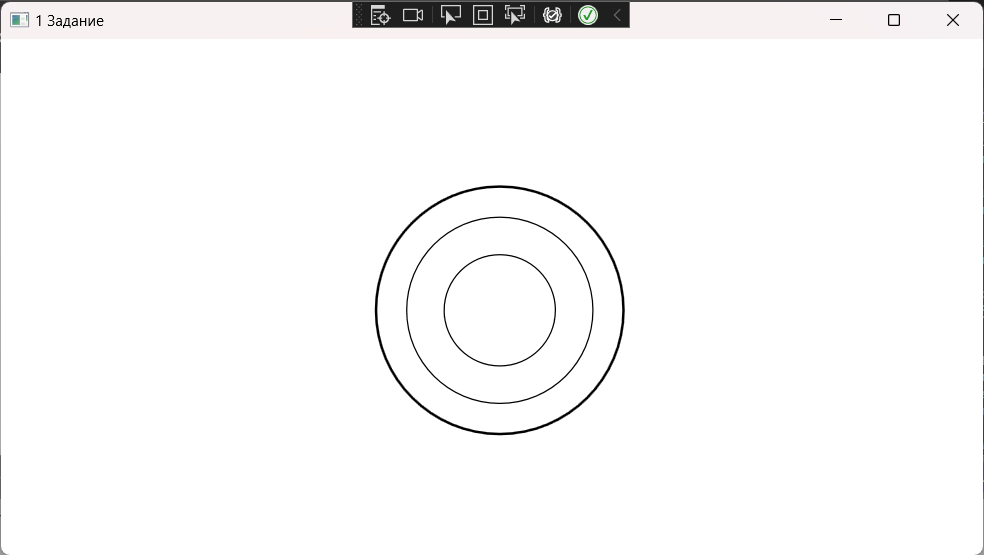


Рисунок 28.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Выполнить задание в приложении WPF используя объект DrawingContext.

Листинг программы:

public partial class MainWindow : Window

{ public MainWindow() { InitializeComponent();

this.Loaded += MainWindow\_Loaded; }

private void MainWindow\_Loaded(object sender, RoutedEventArgs e)

{ DrawingVisual visual = new DrawingVisual();

using (DrawingContext dc = visual.RenderOpen()) {

Point p1 = new Point(10, 10); Point p2 = new Point(50, 50);

Point p3 = new Point(10, 50); PathFigure pathFigure = new PathFigure();

pathFigure.StartPoint = p1; pathFigure.Segments.Add(new LineSegment(p2, true));

pathFigure.Segments.Add(new LineSegment(p3, true));

PathGeometry pathGeometry = new PathGeometry();

pathGeometry.Figures.Add(pathFigure);

dc.DrawGeometry(Brushes.Black, new Pen(Brushes.Black, 2), pathGeometry);

Rect rect = new Rect(70, 10, 40, 60);

dc.DrawEllipse(Brushes.Red, new Pen(Brushes.Blue, 2), new Point(80, 90), 40, 30);

dc.DrawEllipse(Brushes.Green, new Pen(Brushes.Yellow, 2), new Point(140, 40), 30, 30);

dc.DrawRectangle(Brushes.Purple, new Pen(Brushes.Black, 2), new Rect(180, 10, 60, 40));

Point center = new Point(280, 40); double radius = 30;

double startAngle = 30; double endAngle = 330;

bool isLargeArc = endAngle - startAngle > 180;

Size size = new Size(radius, radius);

Point startPoint = new Point(center.X + radius \* Math.Cos(startAngle \* Math.PI / 180),

center.Y + radius \* Math.Sin(startAngle \* Math.PI / 180));

Point endPoint = new Point(center.X + radius \* Math.Cos(endAngle \* Math.PI / 180),

center.Y + radius \* Math.Sin(endAngle \* Math.PI / 180));

PathFigure sectorFigure = new PathFigure();

sectorFigure.StartPoint = center;

sectorFigure.Segments.Add(new LineSegment(startPoint, true));

sectorFigure.Segments.Add(new ArcSegment(endPoint, size, 0, isLargeArc, SweepDirection.Clockwise, true));

sectorFigure.Segments.Add(new LineSegment(center, true));

PathGeometry sectorGeometry = new PathGeometry();

sectorGeometry.Figures.Add(sectorFigure);

dc.DrawGeometry(Brushes.Orange, new Pen(Brushes.Black, 2), sectorGeometry);} DrawingVisualHost host = new DrawingVisualHost(visual);

this.Content = host; } }

public class DrawingVisualHost : FrameworkElement

{ private DrawingVisual \_visual;

public DrawingVisualHost(DrawingVisual visual)

{ \_visual = visual; }

protected override void OnRender(DrawingContext drawingContext)

{ base.OnRender(drawingContext);

drawingContext.DrawDrawing(\_visual.Drawing); } }

Анализ результатов:

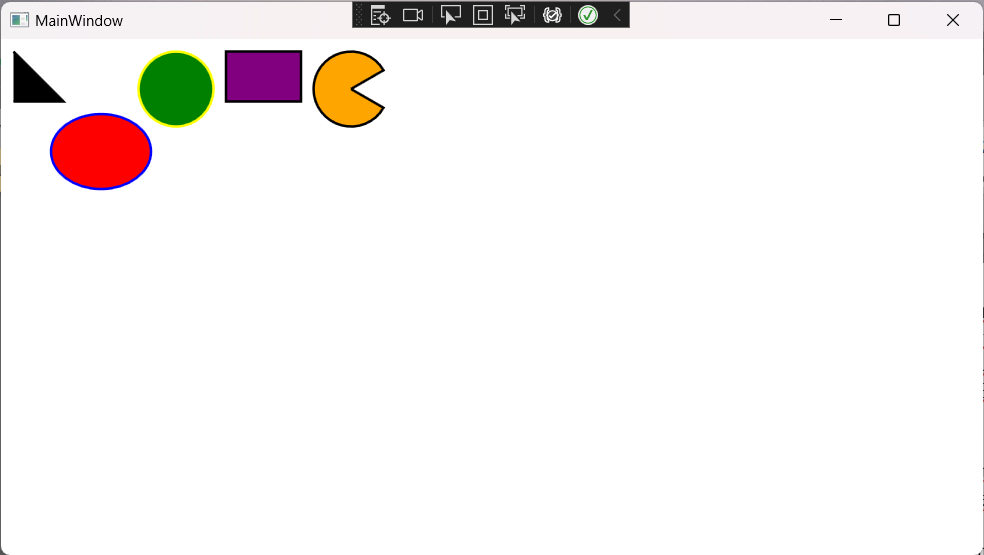


Рисунок 28.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Нарисовать 3 кнопки: Треугольную, круглую, в форме пирамиды

Листинг программы:

<Grid> <Button Width="100" Height="100" Margin="171,175,529,159"

Content="▲" FontSize="50" FontFamily="Segoe UI Symbol"

FontWeight="Bold" BorderThickness="0"

BorderBrush="Black"

Background="White" RenderTransformOrigin="0.622,0.748"/>

<Button Width="100" Height="100" Margin="544,175,0,0"

Content="●" FontSize="50"

FontFamily="Segoe UI Symbol" FontWeight="Bold"

BorderThickness="0" BorderBrush="Black"

Background="White"

Style="{StaticResource {x:Static ToolBar.ButtonStyleKey}}" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<Button Width="100" Height="100" Margin="365,167,335,167"

Content="◢" FontSize="50" FontFamily="Segoe UI Symbol"

FontWeight="Bold" BorderThickness="0"

BorderBrush="Black" Background="White">

<Button.Template>

<ControlTemplate TargetType="{x:Type Button}">

<Grid>

<Polygon Points="0,0 100,0 100,100 0,100" Fill="{TemplateBinding Background}"/>

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/> </Grid>

</ControlTemplate> </Button.Template> </Button> </Grid>

Анализ результатов:

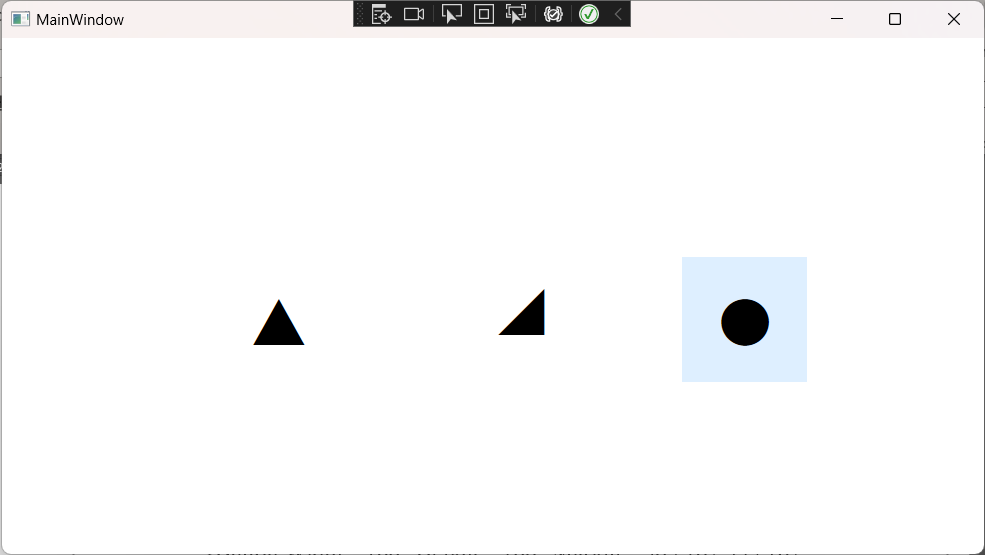


Рисунок 28.3 – Результат работы программы

Источник: собственная разработка