Tristan Izlar

COP2362

TUTORIAL 3-3: Chapter 13 - Creating Interfaces and Defining Abstract Classes

I worked alone.

Chart

Description automatically generated with low confidence

DrawingShape.cs

using Windows.UI;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Shapes;

using Windows.UI.Xaml.Controls;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Drawing

{

abstract class DrawingShape

{

protected int size;

protected int locX = 0, locY = 0;

protected Shape shape = null;

public DrawingShape(int size)

{

this.size = size;

}

public void SetLocation(int xCoord, int yCoord)

{

this.locX = xCoord;

this.locY = yCoord;

}

public void SetColor(Color color)

{

if(this.shape!=null)

{

SolidColorBrush brush = new SolidColorBrush(color);

this.shape.Fill = brush;

}

}

public virtual void Draw(Canvas canvas)

{

if(this.shape == null)

{

throw new InvalidOperationException("Shape is null");

}

this.shape.Height = this.size;

this.shape.Width = this.size;

Canvas.SetTop(this.shape, this.locY);

Canvas.SetLeft(this.shape, this.locX);

canvas.Children.Add(this.shape);

}

}

}

Square.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Windows.UI;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Shapes;

using Windows.UI.Xaml.Controls;

namespace Drawing

{

class Square : DrawingShape, IDraw, IColor

{

public Square(int sideLength):base(sideLength)

{

}

public override void Draw(Canvas canvas)

{

if(this.shape!=null)

{

canvas.Children.Remove(this.shape);

}

else

{

this.shape = new Rectangle();

}

base.Draw(canvas);

}

}

}

Circle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Windows.UI;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Shapes;

using Windows.UI.Xaml.Controls;

namespace Drawing

{

class Circle : DrawingShape, IDraw, IColor

{

public Circle(int diameter):base(diameter)

{

}

public override void Draw(Canvas canvas)

{

if(this.shape!=null)

{

canvas.Children.Remove(this.shape);

}

else

{

this.shape = new Ellipse();

}

base.Draw(canvas);

}

}

}