

Министерство образования Республики Беларусь
Учреждение образования
«Брестский Государственный технический университет»
Кафедра ИИТ

Лабораторная работа №7

По дисциплине «Современные платформы программирования»

Выполнил:

Студент 3 курса

Группы ПО-8

Лобарев А.М.

Проверил:

Крощенко А.А.

Брест 2024

Цель работы: освоить возможности языка программирования C# в построении графических приложений.

Вариант 16

Задание 1

Определить класс Rectangle и класс Point. Объявить массив из n объектов класса Point. Написать функцию, определяющую, какая из точек лежит снаружи, а какая – внутри прямоугольника.

Код:

```
public class MyRect
{
    public Point TopLeft { get; set; }
    public double Width { get; set; }
    public double Height { get; set; }

    public MyRect(Point topLeft, double width, double height)
    {
        TopLeft = topLeft;
        Width = width;
        Height = height;
    }

    // Метод, который проверяет, лежит ли точка внутри прямоугольника.
    public bool Contains(Point point)
    {
        return point.X >= TopLeft.X && point.X <= TopLeft.X + Width &&
            point.Y >= TopLeft.Y && point.Y <= TopLeft.Y + Height;
    }
}

public partial class MainWindow : Window
{
    private readonly List<MyRect> rectangles = [];
    private readonly List<Point> points = [];

    public MainWindow()
    {
        InitializeComponent();
    }

    private void Canvas_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)
    {
        Point mousePosition = e.GetPosition(canvas);

        Rectangle rectangle = new()
        {
            Width = 50,
            Height = 30,
            Fill = Brushes.White,
            Stroke = Brushes.Black,
            StrokeThickness = 1,
        };

        double left = mousePosition.X - rectangle.Width / 2;
        double top = mousePosition.Y - rectangle.Height / 2;

        Canvas.SetLeft(rectangle, left);
        Canvas.SetTop(rectangle, top);
    }
}
```

```

        rectangles.Add(new MyRect(new Point(left, top), rectangle.Width,
rectangle.Height));
        canvas.Children.Add(rectangle);
    }

    private void Canvas_MouseRightButtonDown(object sender, MouseButtonEventArgs e)
    {
        Point mousePosition = e.GetPosition(canvas);

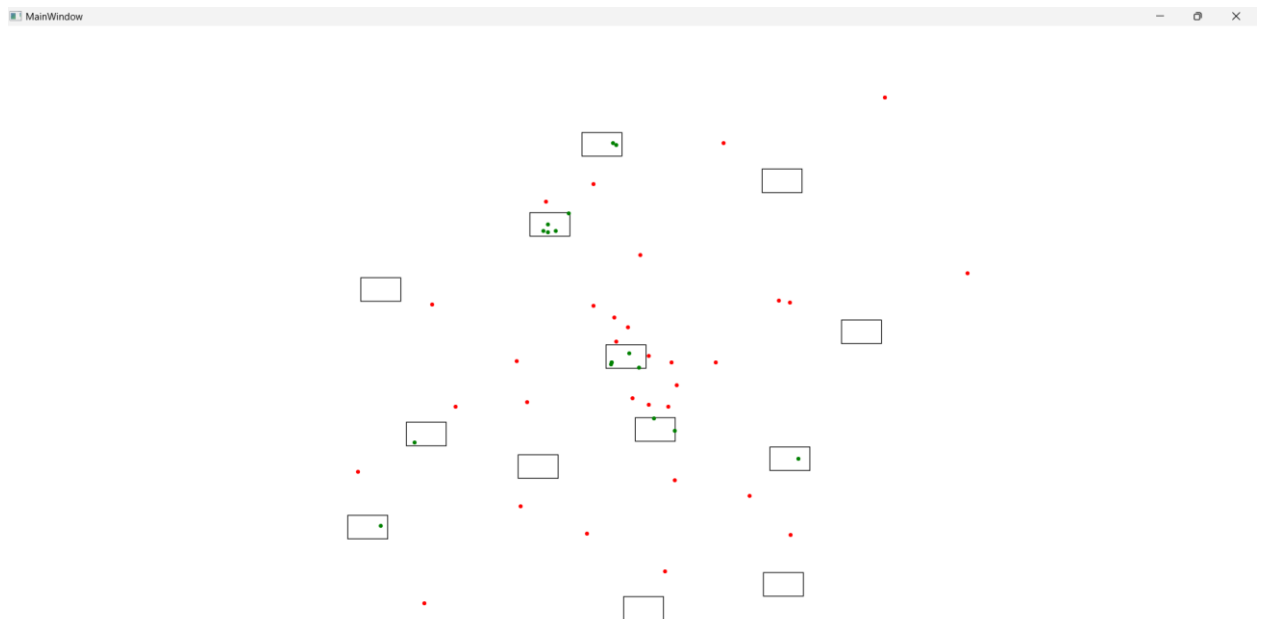
        Ellipse point = new()
        {
            Width = 5,
            Height = 5,
            Fill = Brushes.Red,
        };

        if (rectangles.Any(rect => rect.Contains(mousePosition)))
        {
            point.Fill = Brushes.Green;
        }

        Canvas.SetLeft(point, mousePosition.X - point.Width / 2);
        Canvas.SetTop(point, mousePosition.Y - point.Height / 2);

        points.Add(mousePosition);
        canvas.Children.Add(point);
    }
}

```



Задание 2

H-фрактал.

```

public partial class MainWindow : Window
{
    private readonly double sqrt2 = Math.Sqrt(2);

    public MainWindow()
    {
        InitializeComponent();
    }

    private void canvas_MouseDown(object sender, MouseButtonEventArgs e)
    {
        var point = e.GetPosition(canvas);

        var modal = new ModalWindow();
        if (modal.ShowDialog() == true)
        {
            var depth = (int)modal.Slider;

            DrawHilbertFractal(depth, point);
        }
    }

    private async void DrawHilbertFractal(int depth, Point point)
    {
        double length = 200;

        var start = new Point(point.X - length / 2, point.Y);
        var end = new Point(point.X + length / 2, point.Y);

        Line line = new()
        {
            X1 = start.X,
            X2 = end.X,
            Y1 = start.Y,
            Y2 = end.Y,

            Stroke = Brushes.Black,
            StrokeThickness = 1,
        };

        DrawHorizontal(line, depth);
    }

    private void DrawVertical(Line line, int depth)
    {
        if (depth == 0)
        {
            return;
        }

        canvas.Children.Add(line);

        var length = (line.Y2 - line.Y1) / sqrt2;
        var line1 = new Line()
        {
            X1 = line.X1 - length / 2,
            X2 = line.X1 + length / 2,

            Y1 = line.Y1,
            Y2 = line.Y1,
        };
    }
}

```

```

        Stroke = Brushes.Black,
        StrokeThickness = 1,
    };

    var line2 = new Line()
    {
        X1 = line.X2 - length / 2,
        X2 = line.X2 + length / 2,

        Y1 = line.Y2,
        Y2 = line.Y2,

        Stroke = Brushes.Black,
        StrokeThickness = 1,
    };

    DrawHorizontal(line1, depth - 1);
    DrawHorizontal(line2, depth - 1);
}

private void DrawHorizontal(Line line, int depth)
{
    if (depth == 0)
    {
        return;
    }

    canvas.Children.Add(line);

    var length = (line.X2 - line.X1) / sqrt2;

    var line1 = new Line()
    {
        X1 = line.X1,
        X2 = line.X1,

        Y1 = line.Y1 - length / 2,
        Y2 = line.Y1 + length / 2,

        Stroke = Brushes.Black,
        StrokeThickness = 1,
    };

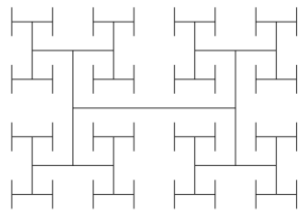
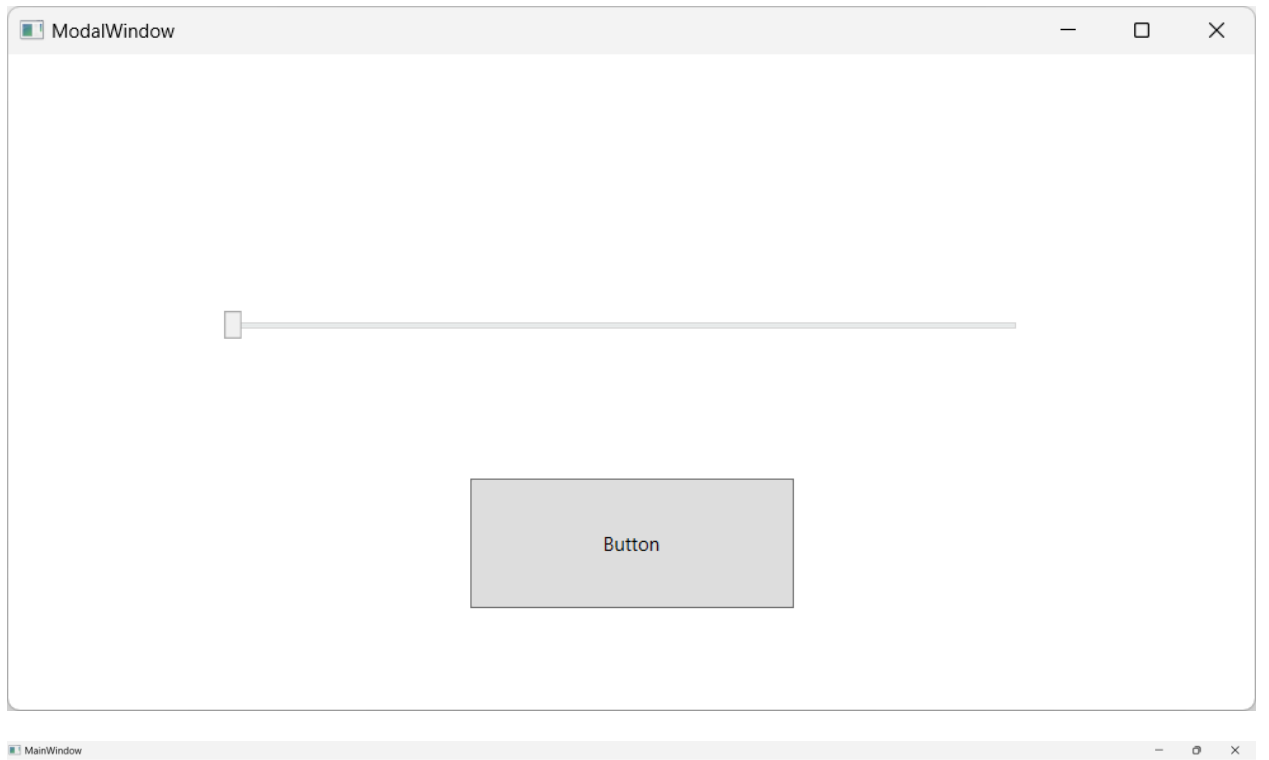
    var line2 = new Line()
    {
        X1 = line.X2,
        X2 = line.X2,

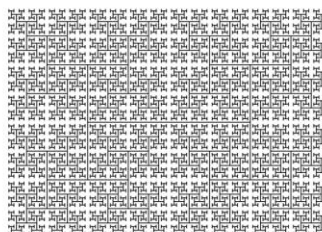
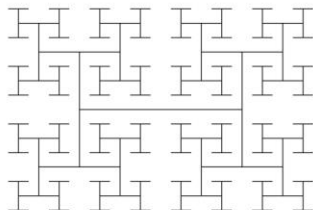
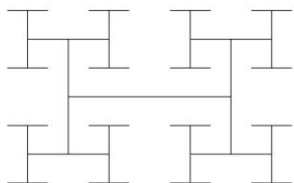
        Y1 = line.Y2 - length / 2,
        Y2 = line.Y2 + length / 2,

        Stroke = Brushes.Black,
        StrokeThickness = 1,
    };

    DrawVertical(line1, depth - 1);
    DrawVertical(line2, depth - 1);
}
}

```





Вывод: освоил возможности языка программирования C# в построении графических приложений.