```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SplashKitSDK;

namespace ShapeDrawing
{
    public class MyLine : Shape
    {
        //Set attributes
        private double _angle;
        private int _length;
        private static int _lineCount = 0;
        //Set constructors and methods
        public MyLine(Color color, int length) : base(color)
        {
            this.Color = color; //Change this line allowing changing line colors
            _length = length;
            double[] angles = { 0, 10, 20, 30, 40 };
            _angle = angles[_lineCount % angles.Length];
            _lineCount++;
        }
        public MyLine() : this(Color.White, 200) { }
        public int Length
        {
            get => _length;
            set => _length = value;
        }
        public override void Draw()
        {
            double radians = DegreesToRadians(_angle);
            double endX = X + _length * Math.Cos(radians);
            double endY = Y + _length * Math.Sin(radians);
            SplashKit.DrawLine(Color, X, Y, endX, endY);
            if (Selected)
            {
                this.DrawOutline();
            }
            //Add this new line
            SplashKit.DrawLine(Color, X, Y, endX, endY);
        }
        private double DegreesToRadians(double degrees)
        {
            return degrees * (Math.PI / 180.0);
        }
        public override bool IsAt(Point2D point)
        {
            double radians = DegreesToRadians(_angle);
            double endX = X + _length * Math.Cos(radians);

            double endY = Y + _length * Math.Sin(radians);
            return SplashKit.PointOnLine(point, SplashKit.LineFrom(X, Y, endX, ↩
              endY));
        }
        public override void DrawOutline()
        {
            double radians = DegreesToRadians(_angle);
            double endX = X + _length * Math.Cos(radians);
            double endY = Y + _length * Math.Sin(radians);
            SplashKit.DrawCircle(Color.Black, X, Y, 3);
            SplashKit.DrawCircle(Color.Black, endX, endY, 3);
        }
        //Step 8 of task 5.3C
        public override void SaveTo(StreamWriter writer)
        {
            writer.WriteLine("Line");
            base.SaveTo(writer);
            writer.WriteLine(Length);
        }
        //Step 15 of task 5.3C
        public override void LoadFrom(StreamReader reader)
        {
            base.LoadFrom(reader);
            Length = reader.ReadInteger();
        }
    }
}
```