```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PointOnLine.BL;
using PointOnLine.UI;

namespace PointOnLine
{
    class Program
    {
        static MyLine option_1()
        {
            MenuUI.Header();

            MyPoint Begin = PointUI.TakeInputOfPoint("Begin");
            MyPoint End = PointUI.TakeInputOfPoint("End");
            MyLine Line = new MyLine(Begin, End);
            MenuUI.InputString();
            return Line;
        }
        static void option_2(MyLine Line)
        {
            MenuUI.Header();
            MenuUI.DisplayMsg("......Update Begin Point.....");
            Line.setBegin(PointUI.TakeInputOfPoint("Begin"));
            MenuUI.InputString();
        }
        static void option_3(MyLine Line)
        {
            MenuUI.Header();
            MenuUI.DisplayMsg("......Update End Point.....");
            Line.setBegin(PointUI.TakeInputOfPoint("End"));
            MenuUI.InputString();
        }
        static void option_4(MyLine Line)
        {
            MenuUI.Header();
            PointUI.DisplayBeginPoint(Line.getBegin());
            MenuUI.InputString();
        }
        static void option_5(MyLine Line)
        {
            MenuUI.Header();
            PointUI.DisplayBeginPoint(Line.getEnd());
            MenuUI.InputString();
        }
        static void option_6(MyLine Line)
        {
            MenuUI.Header();
            LineUI.PrintLenght(Line.getLength());
            MenuUI.InputString();
        }
        static void option_7(MyLine Line)
        {
            MenuUI.Header();
            LineUI.PrintGradient(Line.getGradient());
            MenuUI.InputString();
        }
        static void option_8(MyLine Line)
        {
            MenuUI.Header();
            LineUI.DistanceOfBeginFromOrigin(Line.getBegin().DistanceFromZero());
            MenuUI.InputString();
        }
```

```
        static void option_9(MyLine Line)
        {
            MenuUI.Header();
            LineUI.DistanceOfEndFromOrigin(Line.getEnd().DistanceFromZero());
            MenuUI.InputString();
        }

        static void Main(string[] args)
        {

            MyLine Line = null;
            string option;
            do
            {
                MenuUI.Header();
                option = MenuUI.Menu();
                switch (option)
                {
                    case "1":
                        Line = option_1();
                        break;
                    case "2":
                        option_2(Line);
                        break;
                    case "3":
                        option_3(Line);
                        break;
                    case "4":
                        option_4(Line);
                        break;
                    case "5":
                        option_5(Line);
                        break;
                    case "6":
                        option_6(Line);
                        break;
                    case "7":
                        option_7(Line);
                        break;
                    case "8":
                        option_8(Line);
                        break;
                    case "9":
                        option_9(Line);
                        break;
                    case "10":
                        MenuUI.Header();
                        MenuUI.EndiingMsg();
                        break;
                    default:
                        MenuUI.Header();
                        MenuUI.DisplayMsg("........Wrong Opiton.....");
                        break;
                }
                MenuUI.InputString();
            } while (option != "10");
        }
    }
}
```