

در ادامه [پست قبل](#) ، در این پست به بررسی کلاس Triangle جهت رسم مثلث و کلاس Diamond جهت رسم لوزی می‌پردازیم.

```
using System.Drawing;

namespace PWS.ObjectOrientedPaint.Models
{
    /// <summary>
    /// Triangle
    /// </summary>
    public class Triangle : Shape
    {
        #region Constructors (2)

        /// <summary>
        /// Initializes a new instance of the <see cref="Triangle" /> class.
        /// </summary>
        /// <param name="startPoint">The start point.</param>
        /// <param name="endPoint">The end point.</param>
        /// <param name="zIndex">Index of the z.</param>
        /// <param name="foreColor">Color of the fore.</param>
        /// <param name="thickness">The thickness.</param>
        /// <param name="isFill">if set to <c>true</c> [is fill].</param>
        /// <param name="backgroundColor">Color of the background.</param>
        public Triangle(PointF startPoint, PointF endPoint, int zIndex, Color foreColor, byte
thickness, bool isFill, Color backgroundColor)
            : base(startPoint, endPoint, zIndex, foreColor, thickness, isFill, backgroundColor)
        {
            ShapeType = ShapeType.Triangle;
        }

        /// <summary>
        /// Initializes a new instance of the <see cref="Triangle" /> class.
        /// </summary>
        public Triangle()
        {
            ShapeType = ShapeType.Triangle;
        }

        #endregion Constructors

        #region Methods (1)

        // Public Methods (1)

        /// <summary>
        /// Draws the specified g.
        /// </summary>
        /// <param name="g">The g.</param>
        public override void Draw(Graphics g)
        {
            var points = new PointF[3];
            points[0] = new PointF(X + Width / 2, Y);
            points[1] = new PointF(X + Width, Y + Height);
            points[2] = new PointF(X, Y + Height);
            if (IsFill)
                g.FillPolygon(BackgroundBrush, points);
            g.DrawPolygon(new Pen(ForeColor, Thickness), points);
            base.Draw(g);
        }

        #endregion Methods
    }
}
```

همانگونه که مشاهده می‌کنید کلاس مثلث از کلاس Shape ارث برده و تشکیل شده از یک سازنده و بازنویسی (override) متد Draw می‌باشد، البته متد HasPointInShape در کلاس پایه قاعدتا باید بازنویسی شود، برای تشخیص وجود نقطه در شکل مثلث،

(اگر دوستان فرمولش می‌دونن ممنون می‌شم در اختیار بذارن). در متد Draw سه نقطه مثلث در نظر گرفته شده که بر طبق آن با استفاده از متدهای رسم منحنی اقدام به رسم مثلث توپر یا تو خالی نموده‌ایم.

کلاس لوزی نیز دقیقاً مانند کلاس مثلث عمل می‌کند.

```
using System.Drawing;

namespace PWS.ObjectOrientedPaint.Models
{
    /// <summary>
    /// Diamond
    /// </summary>
    public class Diamond : Shape
    {
        #region Constructors (2)

        /// <summary>
        /// Initializes a new instance of the <see cref="Diamond" /> class.
        /// </summary>
        /// <param name="startPoint">The start point.</param>
        /// <param name="endPoint">The end point.</param>
        /// <param name="zIndex">Index of the z.</param>
        /// <param name="foreColor">Color of the fore.</param>
        /// <param name="thickness">The thickness.</param>
        /// <param name="isFill">if set to <c>true</c> [is fill].</param>
        /// <param name="backgroundColor">Color of the background.</param>
        public Diamond(PointF startPoint, PointF endPoint, int zIndex, Color foreColor, byte thickness,
            bool isFill, Color backgroundColor)
            : base(startPoint, endPoint, zIndex, foreColor, thickness, isFill, backgroundColor)
        {
            ShapeType = ShapeType.Diamond;
        }

        /// <summary>
        /// Initializes a new instance of the <see cref="Diamond" /> class.
        /// </summary>
        public Diamond()
        {
            ShapeType = ShapeType.Diamond;
        }

        #endregion Constructors

        #region Methods (1)

        // Public Methods (1)

        /// <summary>
        /// Draws the specified g.
        /// </summary>
        /// <param name="g">The g.</param>
        public override void Draw(Graphics g)
        {
            var points = new PointF[4];
            points[0] = new PointF(X + Width / 2, Y);
            points[1] = new PointF(X + Width, Y + Height / 2);
            points[2] = new PointF(X + Width / 2, Y + Height);
            points[3] = new PointF(X, Y + Height / 2);
            if (IsFill)
                g.FillPolygon(BackgroundBrush, points);
            g.DrawPolygon(new Pen(ForeColor, Thickness), points);
            base.Draw(g);
        }

        #endregion Methods
    }
}
```

این کلاس نیز از کلاس Shape ارث برده و دارای یک سازنده بوده و متد Draw را از نو بازنویسی می‌کند، این متد نیز با استفاده از چهار نقطه و استفاده از متد رسم منحنی در دات نت اقدام به طراحی لوزی توپر یا تو خالی می‌کند، متد HasPointInShape در کلاس پایه قاعدتاً باید بازنویسی شود، برای تشخیص وجود نقطه در شکل لوزی، برای رسم لوزی توپر نیز خصوصیت BackgroundBrush استفاده کرده و شی توپر را رسم می‌کند.

مباحث رسم مستطیل و مربع، دایره و بیضی در پست‌های بعد بررسی خواهند شد.

[پیاده سازی پروژه نقاشی \(Paint\) به صورت شی گرا #1](#)

[پیاده سازی پروژه نقاشی \(Paint\) به صورت شی گرا #2](#)

[پیاده سازی پروژه نقاشی \(Paint\) به صورت شی گرا #3](#)

موفق و موید باشید.