عنوان: **پیاده سازی پروژه نقاشی (Paint) به صورت شی گرا #2**
نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۰:۳۵
آدرس: www.dotnettips.info
برچسب‌ها: OOP, C#, Project, Paint

در ادامه مطلب پیاده سازی پروژه نقاشی (Paint) به صورت شی گرا #1 به تشریح مابقی کلاس‌های برنامه می‌پردازیم.

با توجه به تجزیه و تحلیل انجام شده تمامی اشیا از کلاس پایه به نام Shape ارث بری دارند حال به توضیح کدهای این کلاس می‌پردازیم. (به دلیل اینکه توضیحات این کلاس در دو پست نوشته خواهد شد برای این کلاس‌ها از partial class استفاده شده است)

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Net;

namespace PWS.ObjectOrientedPaint.Models
{
    /// <summary>
    /// Shape (Base Class)
    /// </summary>
    public abstract partial class Shape
    {
        #region Fields (1)

        private Brush _backgroundBrush;

        #endregion Fields

        #region Properties (16)

        /// <summary>
        /// Gets or sets the brush.
        /// </summary>
        /// <value>
        /// The brush.
        /// </value>
        public Brush BackgroundBrush
        {
            get { return _backgroundBrush ?? (_backgroundBrush = new SolidBrush(BackgroundColor)); }
            private set
            {
                _backgroundBrush = value ?? new SolidBrush(BackgroundColor);
            }
        }

        /// <summary>
        /// Gets or sets the color of the background.
        /// </summary>
        /// <value>
        /// The color of the background.
        /// </value>
        public Color BackgroundColor { get; set; }

        /// <summary>
        /// Gets or sets the end point.
        /// </summary>
        /// <value>
        /// The end point.
        /// </value>
        public PointF EndPoint { get; set; }

        /// <summary>
        /// Gets or sets the color of the fore.
        /// </summary>
        /// <value>
        /// The color of the fore.
        /// </value>
        public Color ForeColor { get; set; }

        /// <summary>
        /// Gets or sets the height.
        /// </summary>
        /// <value>
```

```
        /// The height.
        /// </value>
        public float Height
        {
            get
            {
                return Math.Abs(StartPoint.Y - EndPoint.Y);
            }
            set
            {
                if (value > 0)
                    EndPoint = new PointF(EndPoint.X, StartPoint.Y + value);
            }
        }

        /// <summary>
        /// Gets or sets a value indicating whether this instance is fill.
        /// </summary>
        /// <value>
        ///   <c>true</c> if this instance is fill; otherwise, <c>false</c>.
        /// </value>
        public bool IsFill { get; set; }

        /// <summary>
        /// Gets or sets a value indicating whether this instance is selected.
        /// </summary>
        /// <value>
        /// <c>true</c> if this instance is selected; otherwise, <c>false</c>.
        /// </value>
        public bool IsSelected { get; set; }

        /// <summary>
        /// Gets or sets my pen.
        /// </summary>
        /// <value>
        /// My pen.
        /// </value>
        public Pen Pen
        {
            get
            {
                return new Pen(ForeColor, Thickness);
            }
        }

        /// <summary>
        /// Gets or sets the type of the shape.
        /// </summary>
        /// <value>
        /// The type of the shape.
        /// </value>
        public ShapeType ShapeType { get; protected set; }

        /// <summary>
        /// Gets the size.
        /// </summary>
        /// <value>
        /// The size.
        /// </value>
        public SizeF Size
        {
            get
            {
                return new SizeF(Width, Height);
            }
        }

        /// <summary>
        /// Gets or sets the start point.
        /// </summary>
        /// <value>
        /// The start point.
        /// </value>
        public PointF StartPoint { get; set; }

        /// <summary>
        /// Gets or sets the thickness.
        /// </summary>
        /// <value>
        /// The thickness.
        /// </value>
```

```
        public byte Thickness { get; set; }

        /// <summary>
        /// Gets or sets the width.
        /// </summary>
        /// <value>
        /// The width.
        /// </value>
        public float Width
        {
            get
            {
                return Math.Abs(StartPoint.X - EndPoint.X);
            }
            set
            {
                if (value > 0)
                    EndPoint = new PointF(StartPoint.X + value, EndPoint.Y);
            }
        }

        /// <summary>
        /// Gets or sets the X.
        /// </summary>
        /// <value>
        /// The X.
        /// </value>
        public float X
        {
            get
            {
                return StartPoint.X;
            }
            set
            {
                if (value > 0)
                    StartPoint = new PointF(value, StartPoint.Y);
            }
        }

        /// <summary>
        /// Gets or sets the Y.
        /// </summary>
        /// <value>
        /// The Y.
        /// </value>
        public float Y
        {
            get
            {
                return StartPoint.Y;
            }
            set
            {
                if (value > 0)
                    StartPoint = new PointF(StartPoint.X, value);
            }
        }

        /// <summary>
        /// Gets or sets the index of the Z.
        /// </summary>
        /// <value>
        /// The index of the Z.
        /// </value>
        public int Zindex { get; set; }

        #endregion Properties
        }
}
```

ابتدا به تشریح خصوصیات کلاس می‌پردازیم:

خصوصیات:

BackgroundColor : در صورتی که شی مورد نظر به صورت توپررسم شود، این خاصیت رنگ پس زمینه شی را مشخص می‌کند.
BackgroundBrush : خاصیتی است که با توجه به خاصیت BackgroundColor یک الگوی پر کردن زمینه شی می‌سازد.
StartPoint : نقطه شروع شی را در خود نگهداری می‌کند.
EndPoint : نقطه انتهای شی را در خود نگهداری می‌کند. (قبلا گفته شد که هر شی را در صورتی که در یک مستطیل فرض کنیم یک نقطه شروع و یک نقطه پایان دارد)
ForeColor : رنگ قلم ترسیم شی مورد نظر را تعیین می‌کند.
Height : ارتفاع شی مورد نظر را تعیین می‌کند (این خصوصیت اختلاف عمودی StartPoint.Y و EndPoint.Y را محاسبه می‌کند و در زمان مقدار دهی EndPoint جدیدی ایجاد می‌کند).
Width : عرض شی مورد نظر را تعیین می‌کند (این خصوصیت اختلاف افقیStartPoint.X و EndPoint.X را محاسبه می‌کند و در زمان مقدار دهی EndPoint جدیدی ایجاد می‌کند).
IsFill : این خصوصیت تعیین کننده توپر و یا توخالی بودن شی است.
IsSelected : این خاصیت تعیین می‌کند که آیا شی انتخاب شده است یا خیر (در زمان انتخاب شی چهار مربع کوچک روی شی رسم می‌شود).
Pen : قلم خط ترسیم شی را مشخص می‌کند. (قلم با ضخامت دلخواه)
ShapeType : این خصوصیت نوع شی را مشخص می‌کند (این خاصیت بیشتر برای زمان پیش نمایش ترسیم شی در زمان اجراست البته به نظر خودم اضافه هست اما راه بهتری به ذهنم نرسید)
Size : با استفاده از خصوصیات Height و Width ایجاد شده و تعیین کننده Size شی است.
Thickness : ضخامت خط ترسیمی شی را مشخص می‌کند، این خاصیت در خصوصیت Pen استفاده شده است.
X : مقدار افقی نقطه شروع شی را تعیین می‌کند در واقع StartPoint.X را برمی‌گرداند (این خاصیت اضافی بوده و جهت راحتی کار استفاده شده می‌توان آن را ننوشت).
Y : مقدار عمودی نقطه شروع شی را تعیین می‌کند در واقع StartPoint.Y را برمی‌گرداند (این خاصیت اضافی بوده و جهت راحتی کار استفاده شده می‌توان آن را ننوشت).
Zindex : در زمان ترسیم اشیا ممکن است اشیا روی هم ترسیم شوند، در واقع Zindex تعیین کننده عمق شی روی بوم گرافیکی است.

در پست بعدی به توضیح متدهای این کلاس می‌پردازیم.

نظرات خوانندگان

نویسنده: بتیسا
تاریخ: ۱۳۹۱/۱۱/۱۸ ۱۰:۲۶

با سلام

از مطلب مفیدی که تهیه کردید ممنون.
می‌شود از طریق خاصیت Brush که فعلا فقط خواندنی هست، طرح‌های مختلفی برای پس زمینه اشیاع ایجاد کرد. مانند Paint.net و یا MS Paint.

اگر به صورت زیر تعریف کنیم فکر می‌کنم کمی کامل‌تر باشه!

```
private Brush _backgroundBrush;

        /// <summary>
        /// Gets or sets the brush.
        /// </summary>
        /// <value>
        /// The brush.
        /// </value>
        public Brush BackgroundBrush
        {
            get
            {
                return _backgroundBrush;
            }
            private set
            {
                _backgroundBrush = (value != null) ? value : new SolidBrush(BackgroundColor);
            }
        }
        //-------------------------------------[Methode for set brush]---------------------------------
-----
        public virtual void SetBackgroundBrushAsHatch(HatchStyle hatchStyle)
        {
            HatchBrush brush = new HatchBrush(hatchStyle, BackgroundColor);
            BackgroundBrush = brush;
        }

        public virtual void SetBackgroundBrushAsSolid()
        {
            SolidBrush brush = new SolidBrush(BackgroundColor);
            BackgroundBrush = brush;
        }

        public virtual void SetBackgroundBrushAsLinearGradient()
        {
            LinearGradientBrush brush = new LinearGradientBrush(StartPoint, EndPoint, ForeColor,
BackgroundColor);
            BackgroundBrush = brush;
        }
```

که اگر بخواهیم میتونیم باز بیشتر Customize بکنیمشون.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۱۰:۴۰

بله کاملا حق با شماست خیلی کارها می‌شه روش انجام داد (قصد آموزش یک مبحث به زبان ساده بود)==> نظر شما اعمال شد.
سعی می‌کنم در زمان ارائه پروژه نهایی همه اینها اعمال بشه