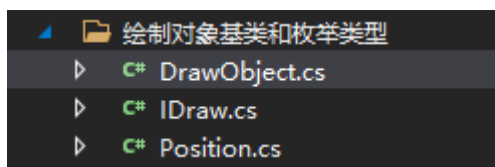
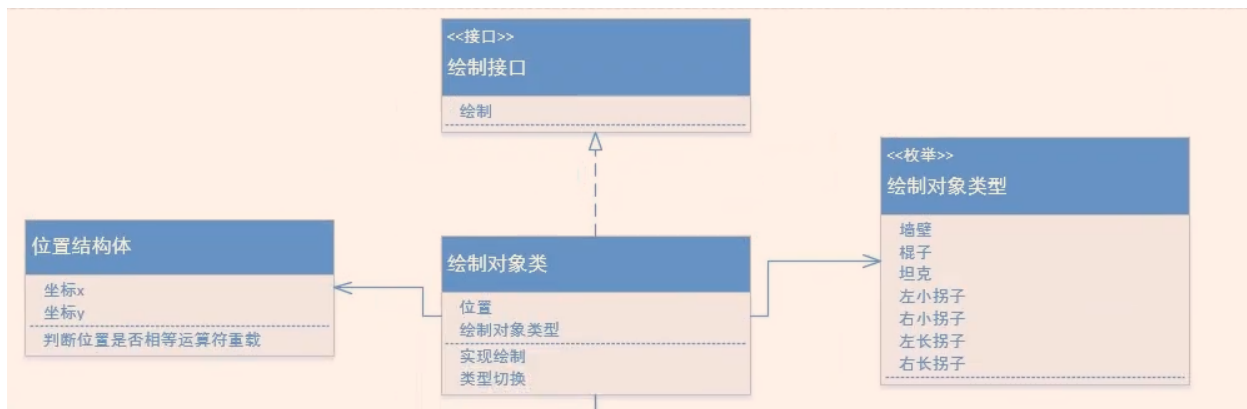


3：绘制对象基类和枚举等信息



绘制接口

```
namespace 俄罗斯方块. 绘制对象基类和枚举类型
{
    1 个引用
    interface IDraw
    {
        1 个引用
        void Draw();
    }
}
```

位置结构体：

```

namespace 俄罗斯方块
{
    9 个引用
    struct Position//结构体
    {
        public int x;
        public int y;
        1 个引用
        public Position(int x,int y)
        {
            this.x = x;
            this.y = y;
        }

        //还需要比较
        0 个引用
        public static bool operator==(Position v1,Position v2)
        {
            return v1.x == v2.x && v1.y == v2.y;
        }
        0 个引用
        public static bool operator!=(Position v1,Position v2)
        {
            return v1.x != v2.x && v1.y != v2.y;
        }

        0 个引用
        public static Position operator+(Position v1,Position v2)
        {
            return new Position(v1.x + v2.x, v1.y + v2.y);
        }
    }
}

```

绘制对象类（最小单位）和方块的枚举类型

```

namespace 俄罗斯方块. 绘制对象基类和枚举类型
{
    12 个引用
    enum E_DrawType
    {
        /// <summary>
        /// 墙壁
        /// </summary>
        Wall,
        /// <summary>
        /// 正方形
        /// </summary>
        Cube,
        /// <summary>
        /// 直线
        /// </summary>
        Line,
        /// <summary>
        /// 坦克形
        /// </summary>
        Tank,
        /// <summary>
        /// 左梯子形
        /// </summary>
        Left_Ladder,
        /// <summary>
        /// 右梯子形
        /// </summary>
        Right_Ladder,
        /// <summary>
        /// 左长梯子形

```

```

    /// </summary>
    Left_Long_Ladder,
    /// <summary>
    /// 右长梯子形
    /// </summary>
    Right_Long_Ladder,
}

3 个引用
class DrawObject : IDraw //表示游戏里面的最小单位(一个小方块对象)
{
    /// <summary>
    /// 根据不同类型来改变 改变绘制方块的颜色
    /// </summary>
    public Position pos;
    public E_DrawType type;

    4 个引用
    public DrawObject(E_DrawType type)
    {
        this.type = type;
    }

    0 个引用
    public DrawObject(E_DrawType type, int x, int y) : this(type)
    {
        pos = new Position(x, y);
    }

    1 个引用
    public void Draw()
    {
        Console.SetCursorPosition(pos.x, pos.y);
        switch (type)
        {
            case E_DrawType.Wall:
                Console.ForegroundColor = ConsoleColor.Red;
                break;
            case E_DrawType.Cube:
                Console.ForegroundColor = ConsoleColor.Blue;

                break;
            case E_DrawType.Line:
                Console.ForegroundColor = ConsoleColor.Green;

                break;
            case E_DrawType.Tank:
                Console.ForegroundColor = ConsoleColor.Cyan;
                break;
            case E_DrawType.Left_Ladder:
            case E_DrawType.Right_Ladder:
                Console.ForegroundColor = ConsoleColor.Magenta;
                break;
            case E_DrawType.Left_Long_Ladder:
            case E_DrawType.Right_Long_Ladder:
                Console.ForegroundColor = ConsoleColor.DarkGray;
                break;
        }

        Console.Write("■");
    }

    /// <summary>
    /// 切换方块类型, 用于板砖下落到底切换为墙类型
    /// </summary>
    /// <param name="type"></param>
    0 个引用
    public void ChangeType(E_DrawType type)
    {
        this.type = type;
    }
}

```