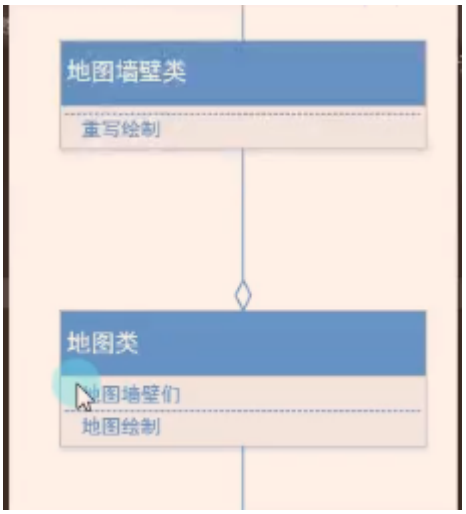
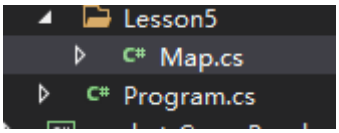
任务60：地图对象 知识点



绘制四周墙壁：



```
namespace Snake.Lesson5
{
    3 个引用
    class Map:IDraw
    {
        private Wall[] walls;

        1 个引用
        public Map()
        {
            walls = new Wall[Game.w + (Game.h - 3) * 2];

            int index = 0;//记录墙体数组的下标
            //分配x轴墙体位置
            for (int i = 0; i < Game.w; i+=2)
            {
                walls[index++] = new Wall(i, 0);
            }
            for (int i = 0; i < Game.w; i+=2)
            {
                walls[index++] = new Wall(i, Game.h - 2);
            }
            //分配y轴墙体位置
            for (int i = 1; i < Game.h-2; i++)
            {
                walls[index++] = new Wall(0, i);
            }
            for (int i = 1; i < Game.h - 2; i++)
            {
                walls[index++] = new Wall(Game.w-2, i);
            }


        }

        /// <summary>
        /// 将分配好的墙体画出来（每个墙体都是一个wall对象）
        /// </summary>
        2 个引用
        public void Draw()
        {
            for (int i = 0; i < walls.Length; i++)
            {
                walls[i].Draw();
            }
        }
    }
}
```

地图类直接继承绘制接口，进行一个封装的绘制，因为墙体类都在这里保

聚合，携带前面创建的wall墙体类数组，因为这里是地图类，肯定是需要墙体的

构造墙体的个数

画x轴

画y轴

绘制的直接遍历所有的墙体类，执行墙体的绘制方法

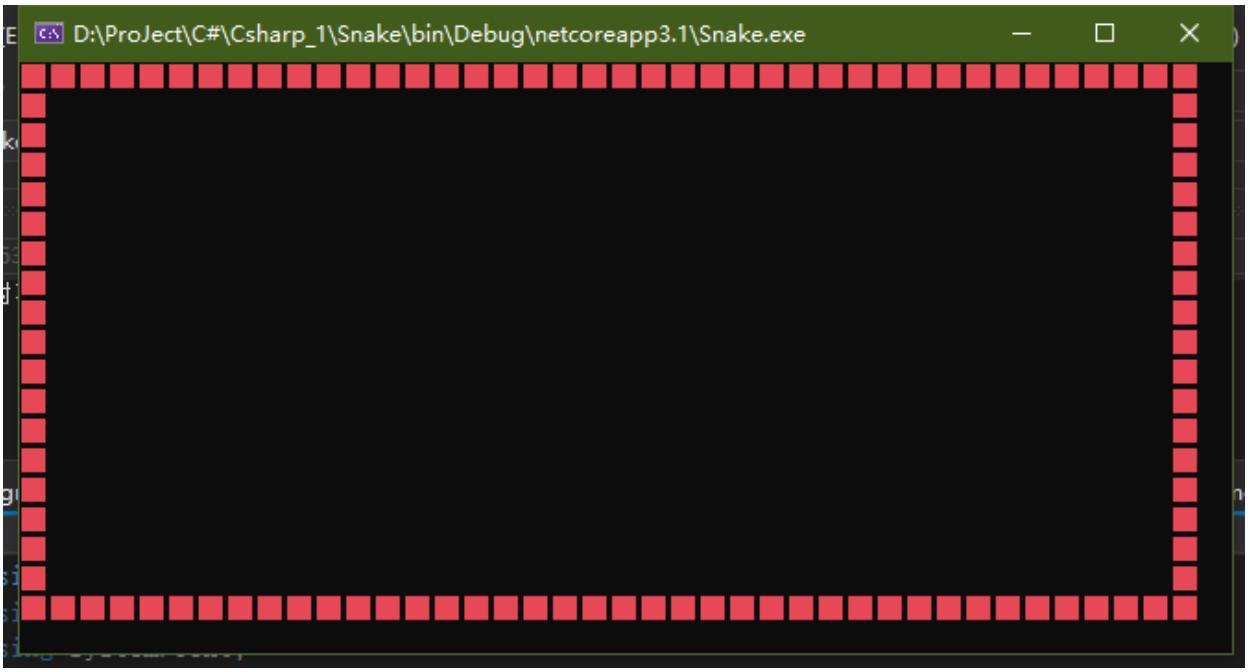
```
namespace Snake.Lesson2
{
    2 个引用
    class GameScene : ISceneUpdate
    {
        private Map map; 地图是在游戏场景的

        1 个引用
        public GameScene()   进入游戏场景就构造
        {                    出地图 符合逻辑
            map = new Map();
        }

        2 个引用
        public void upData()
        {
            map.Draw(); 然后就绘制地图出
        }
    }
}
```

这里的Draw就会执行前面的遍历数组调用绘制函数：

```
namespace Snake.Lesson4
{
    7 个引用
    class Wall : GameObject
    {
        4 个引用
        public Wall(int x,int y)
        {
            pos = new Position(x, y);
        }
        3 个引用
        public override void Draw()
        {
            Console.SetCursorPosition(pos.x, pos.y);
            Console.ForegroundColor = ConsoleColor.Red;
            Console.Write("■");
        }
    }
}
```



总结：地图类Map是需要墙体的，所以这里就会包含一个墙体数组，主要就是对墙体的位置分配和绘制出来，因为墙体本身就实现了IDraw接口，就具有绘制功能，所以我们可以在这里封装一下，也继承IDraw方法，直接调用地图的draw方法，就直接遍历墙体数组，全部绘制一边。代码结构清晰