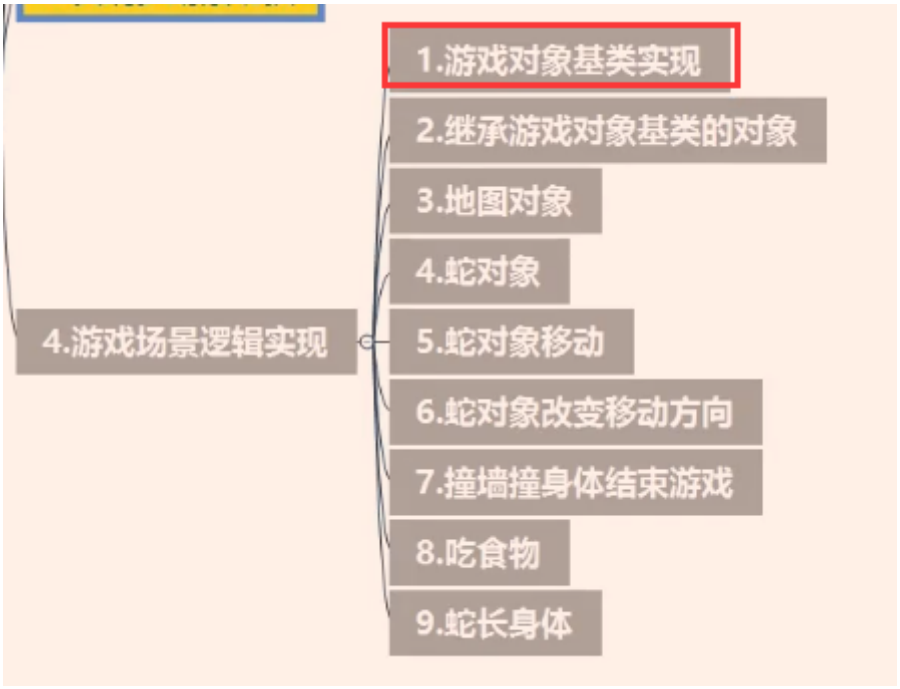
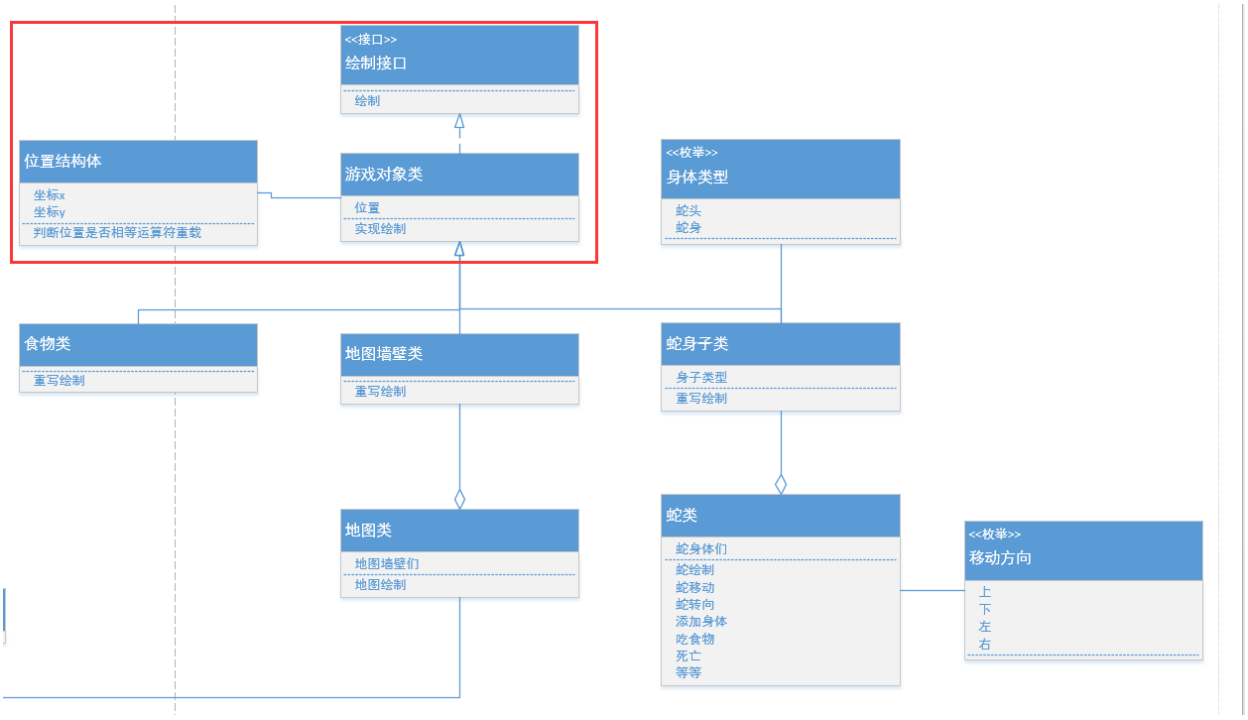
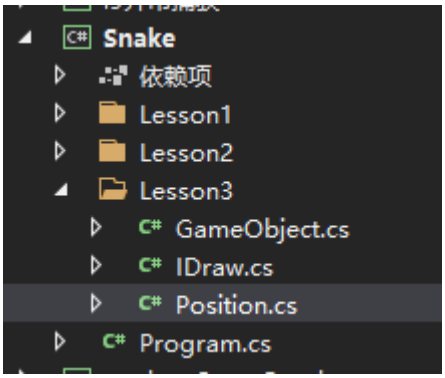
任务58：游戏对象基类 知识点







绘制接口：

```
namespace Snake.Lesson3
{
    0 个引用
    interface IDraw
    {
        /// <summary>
        /// 绘制的接口
        /// </summary>
        0 个引用
        void Draw();
    }
}
```

游戏对象类，因为游戏会有几个对象，但是他们都有公共的特征，比如位置、都要绘制，所以我们可以在抽象一个游戏类，然后让具体的游戏类来继承即可：

```
namespace Snake.Lesson3
{
    0 个引用
    abstract class GameObject:IDraw
    {

        //游戏对象的位置
        public Position pos;  每个游戏类都会位置

        //可以继承接口后，把接口中的行为 编程 抽象行为
        //让子类去实现，因为是抽象行为，所以子类中是必须去实现
        1 个引用
        public abstract void Draw(); 具体子类去实现绘制
    }
}
```

位置类：（封装）

```
namespace Snake.Lesson3
{
    5 个引用
    struct Position
    {
        public int x;
        public int y;
        0 个引用
        public Position(int x, int y)
        {
            this.x = x;
            this.y = y;
        }
        //贪食蛇中，肯定是存在 两个位置进行比较的，
        //既然我们这里写了struct来表示位置，那么是不能直接比较的
        //所以我们可以使用运算符重载，来实现自己的比较需求


        0 个引用
        public static bool operator==(Position p1,Position p2)
        {
            return p1.x == p2.x && p1.y == p2.y;
        }

        0 个引用
        public static bool operator!=(Position p1,Position p2)
        {
            return p1.x != p2.x && p1.y != p2.y;
        }

    }
}
```

位置比较，因为是结构体，所以直接运算符重载