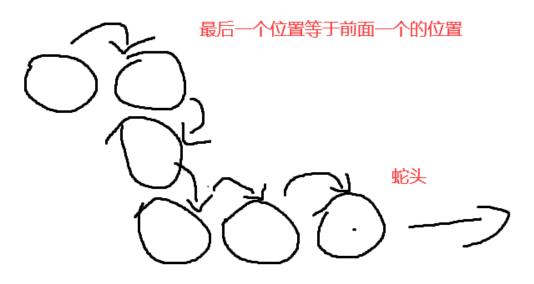
任务66: 长身体 知识点

9.蛇长身体

长身体的核心逻辑就是:

让最后一个身体等于前面一个的身体, 让蛇头的位置自己单独在变化:



但是一定要注意的问题是,在这个位置的类型不能是引用类型,必须是值类型,否者会导致蛇头位置变了,后面的一个位置跟着变,就会导致蛇头和身体位置一致触发撞自己身体的问题。所以这里的位置必须是struct,的而不是class:

```
namespace Snake. Lesson3
10 소리田
 struct Position
    public int x;
    public int y;
    3 个引用
    //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。
    4 个引用
    public static bool operator==(Position p1, Position p2)
        return p1. x == p2. x && p1. y == p2. y;
    public static bool operator!=(Position p1, Position p2)
       return p1. x != p2. x && p1. y != p2. y;
```

```
public void Move()
 //移动前先清理掉最后一个位置
 SnakeBody body = bodys[nowNum - 1];
 Console. SetCursorPosition(body. pos. x, body. pos. y);
 Console. Write (" ");
 //蛇长身体的核心逻辑 新到的位置是上-
 for (int i = nowNum-1; i > 0; i--)
     bodys[i].pos = bodys[i-1].pos;//最后一个位置等于前面的位置
 switch (dir)
                         蛇头自己单独在最后在修改xy
     case E_MoveDir.Up:
        bodys[0]. pos. y--;
        break;
     case E_MoveDir.Down:
        bodys[0]. pos. y++;
        break;
     case E_MoveDir.Left:
        bodys[0]. pos. x -= 2;
        break:
     case E_MoveDir.Right:
        bodys[0]. pos. x += 2;
        break;
```