```swift
import Foundation
/// **剑指 Offer 52. 两个链表的第一个公共节点**
/// 输入两个链表，找出它们的第一个公共节点。
///
/// 输入：intersectVal = 8, listA = [4,1,8,4,5], listB = [5,0,1,8,4,5], skipA = 2, skipB = 3
/// 输出：Reference of the node with value = 8
/// 输入解释：相交节点的值为 8 （注意，如果两个列表相交则不能为 0）。
/// 从各自的表头开始算起，链表 A 为 [4,1,8,4,5]，链表 B 为 [5,0,1,8,4,5]。
/// 在 A 中，相交节点前有 2 个节点；在 B 中，相交节点前有 3 个节点。
public class ListNode {
    public var val: Int
    public var next: ListNode?

    public init(_ val: Int) {
        self.val = val
        self.next = nil
    }
}

class Solution {
    /// 找到两个链表的第一个公共节点
    func getIntersectionNode(_ headA: ListNode?, _ headB: ListNode?) ->
     ListNode? {
        /// 定义两个指针 A 和 B，分别指向链表的头节点 headA 和 headB
        var A = headA
        var B = headB
        /// 当指针 A 和 B 不相同时，循环继续
        while A !== B {
            /// 若指针 A 不为空，则将其指向下一个节点，否则指向链表 headB 的头节点
            A = A?.next ?? headB
            /// 若指针 B 不为空，则将其指向下一个节点，否则指向链表 headA 的头节点
            B = B?.next ?? headA
        }
        /// 循环终止时，返回指针 A（或 B），即为两个链表的第一个公共节点，若没有公共节点则返回
            nil
        return A
    }
}
```