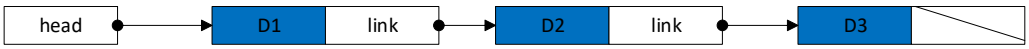
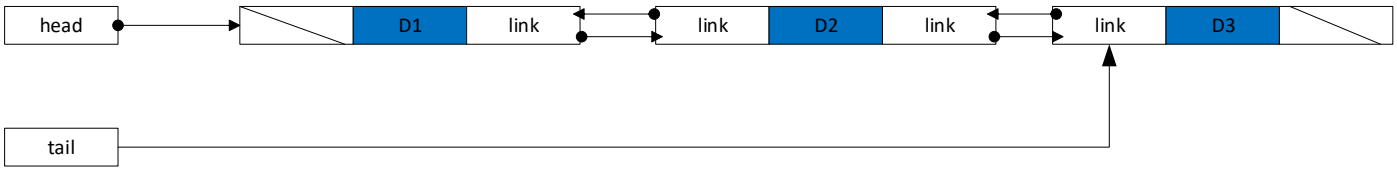
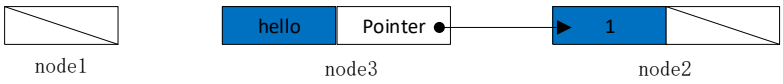
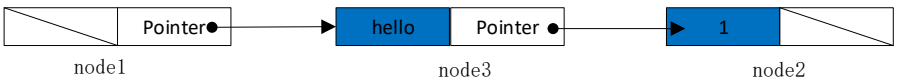


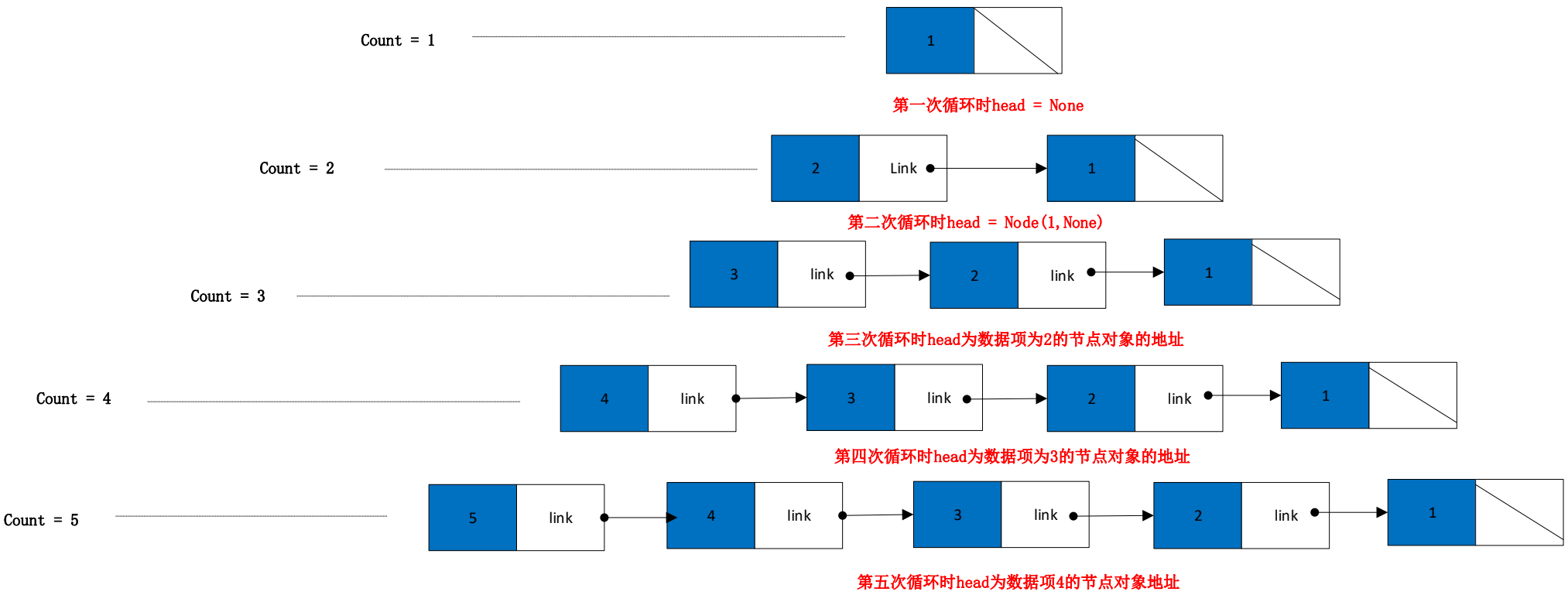
含有3个节点的单向链表的结构



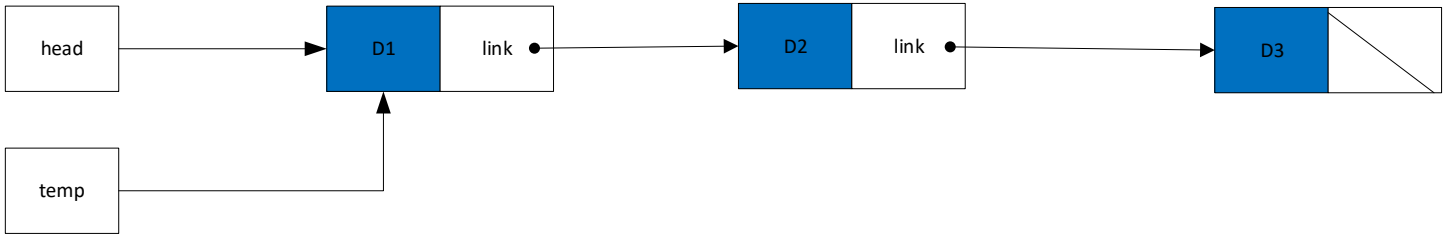
含有3个节点的双向链表的结构


hello
Pointer
1
node1
node3
node2

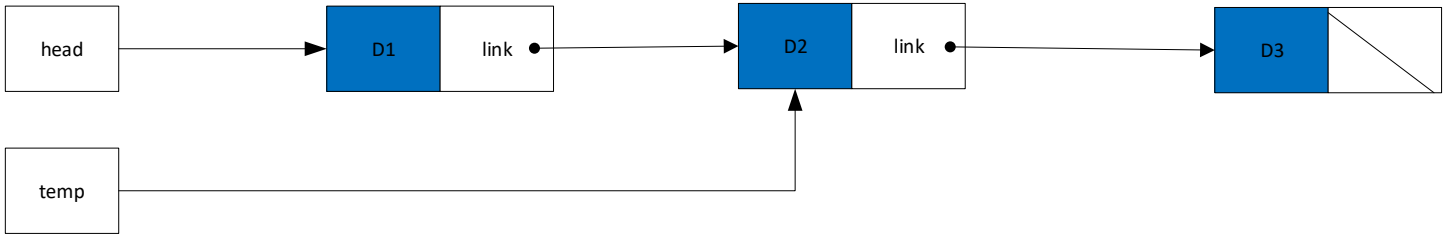

Pointer
hello
Pointer
1
node1
node3
node2


Count = 1
1
第一次循环时head = None
Count = 2
2
Link
1
第二次循环时head = Node(1,None)
Count = 3
3
link
2
link
1
第三次循环时head为数据项为2的节点对象的地址
Count = 4
4
link
3
link
2
link
1
第四次循环时head为数据项为3的节点对象的地址
Count = 5
5
link
4
link
3
link
2
link
1
第五次循环时head为数据项4的节点对象地址

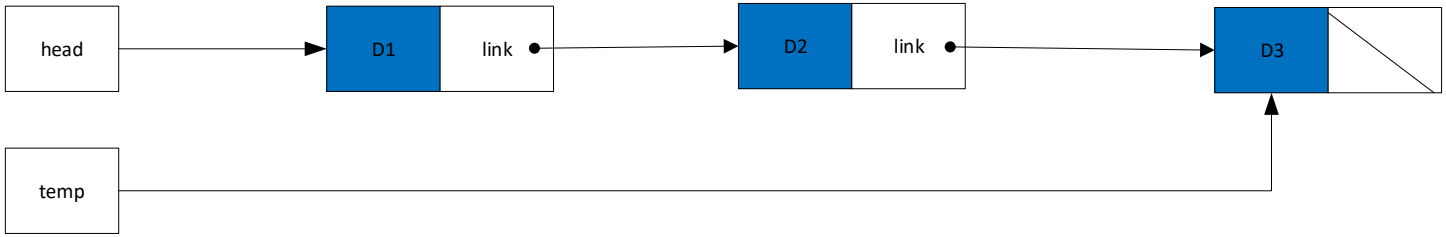
第一轮：访问节点D1



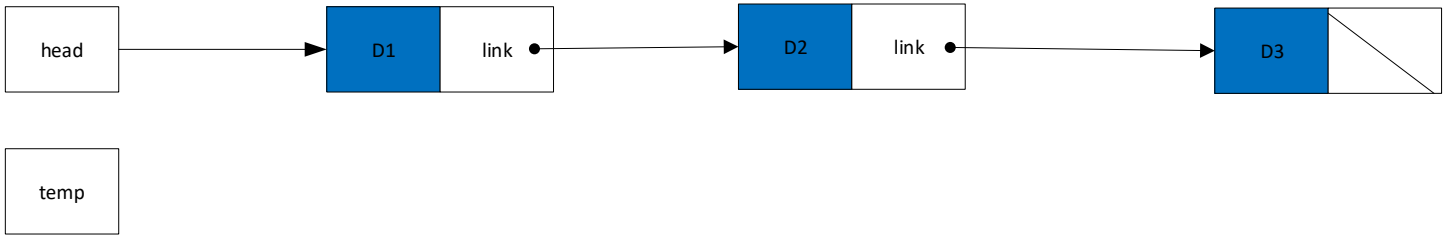
第二轮：访问节点D2



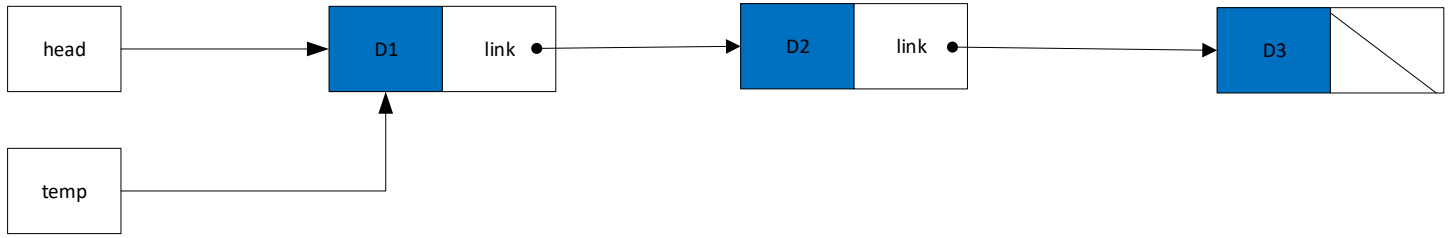
第三轮：访问节点D3



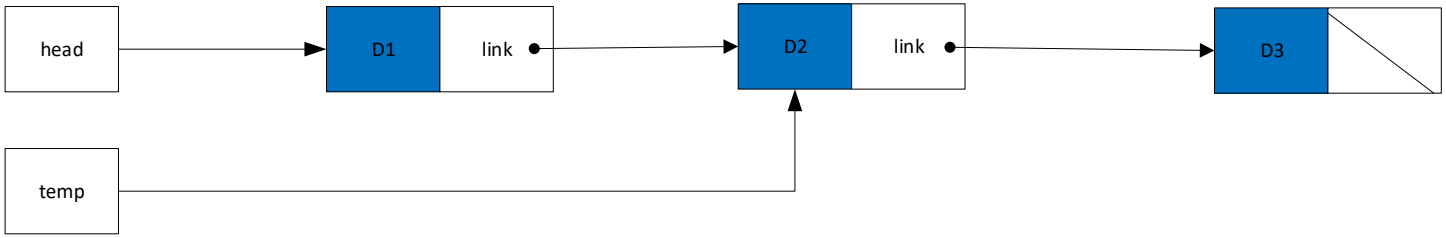
第四轮：temp为None
循环结束



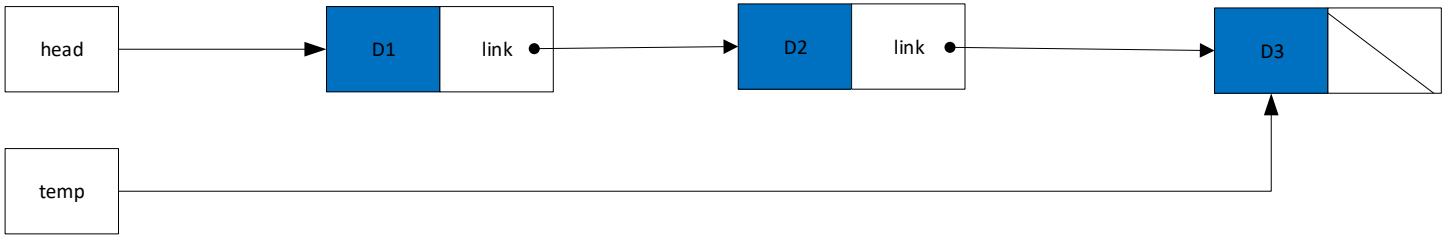
temp.next!=None



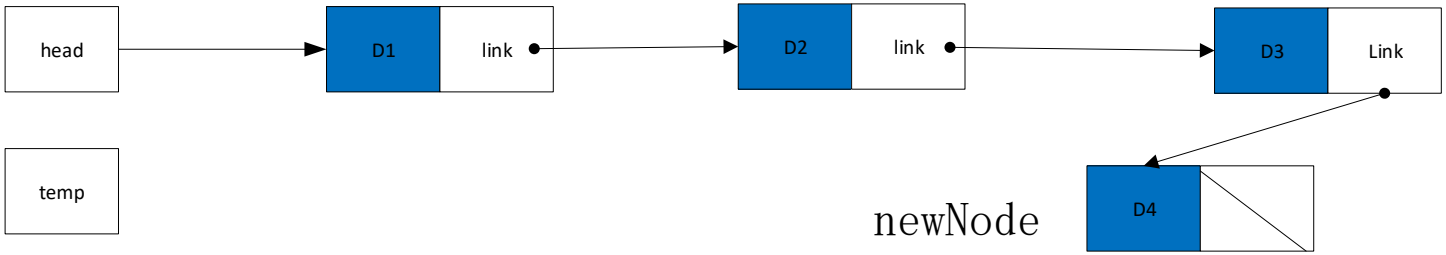
temp.next!=None



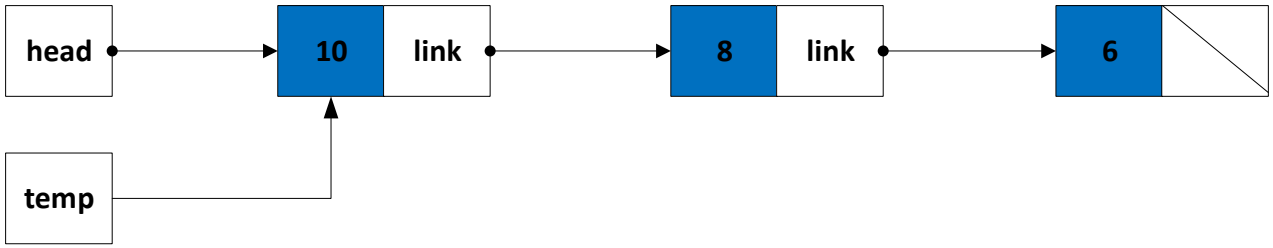
temp.next== None



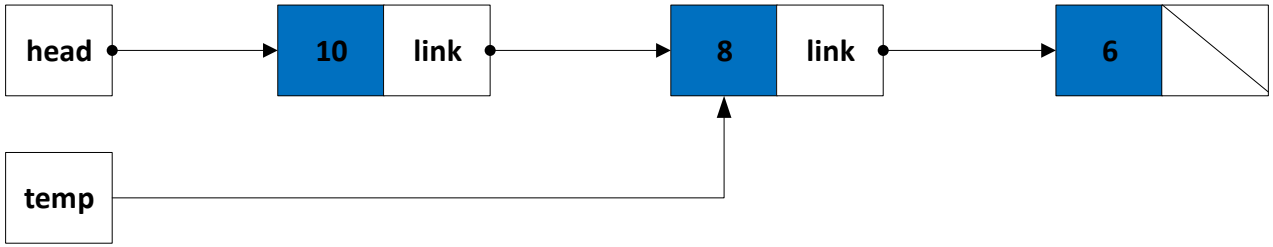
temp.next = newNode
(插入新节点)



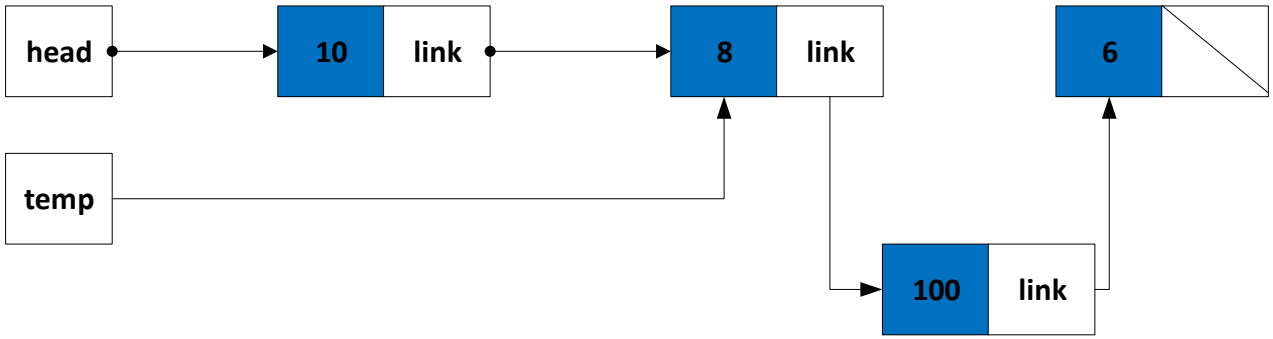
index==2 > 1且
temp.next!= None,
temp = temp.next
index -1



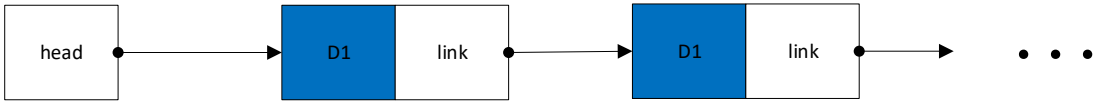
index == 1
停止循环



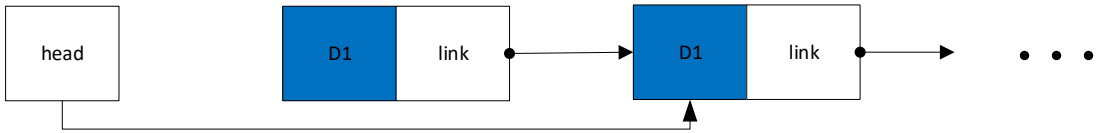
temp.next=Node(100,temp.next)



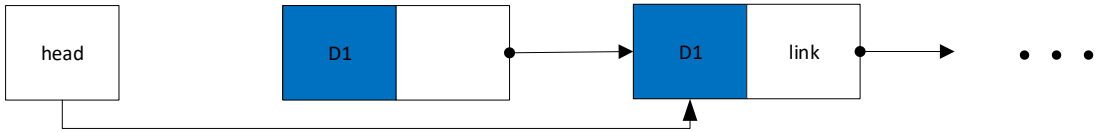
head初始状态



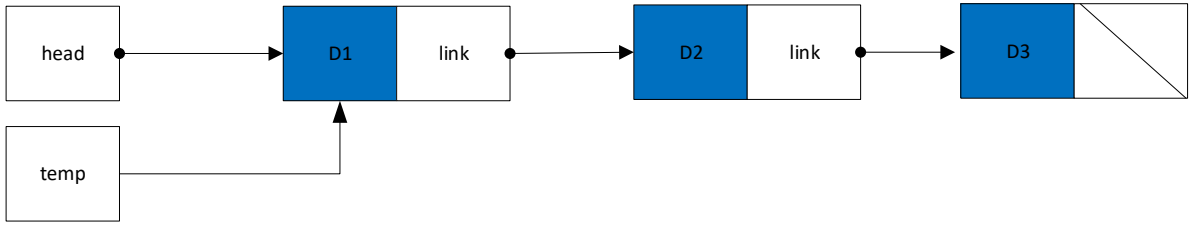
head = head.next
(删除第一个节点)



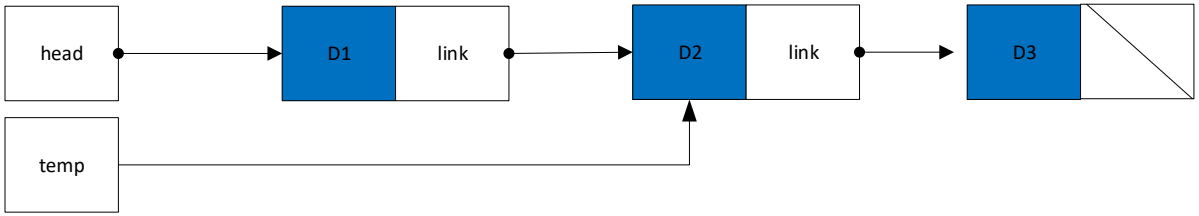
垃圾收集将节点1返回给系统堆



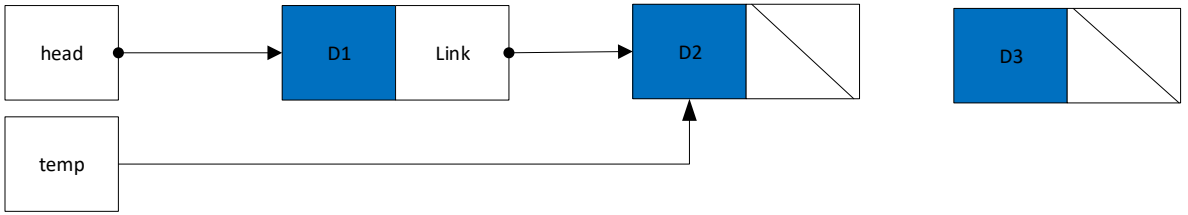
temp.next != None



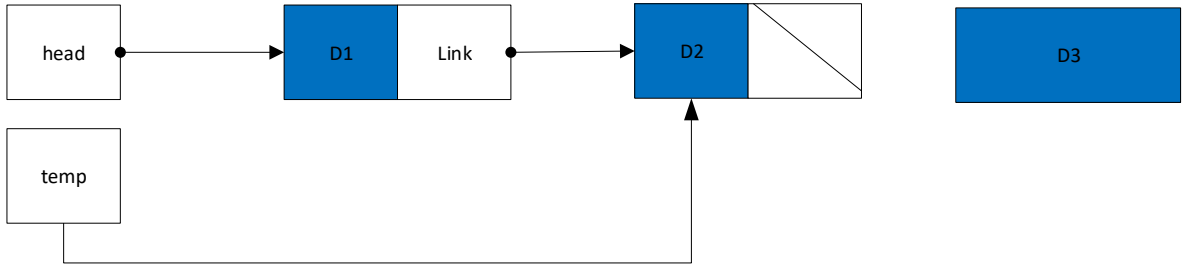
temp.next.next == None
停止循环



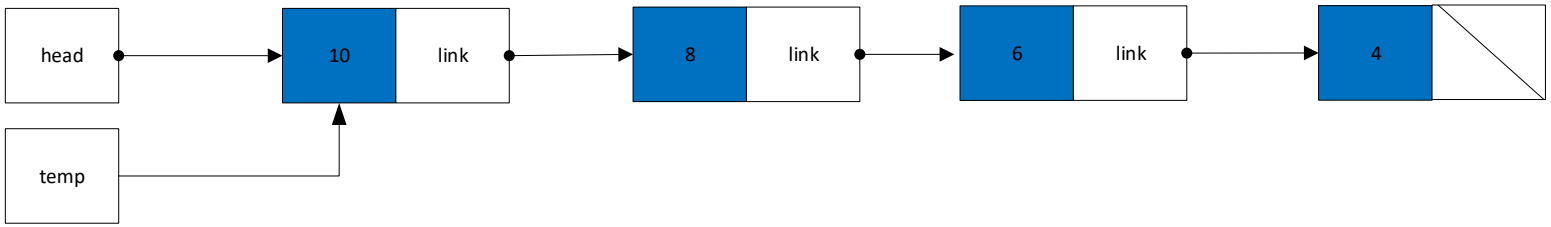
temp.next== None
删除最后一个节点



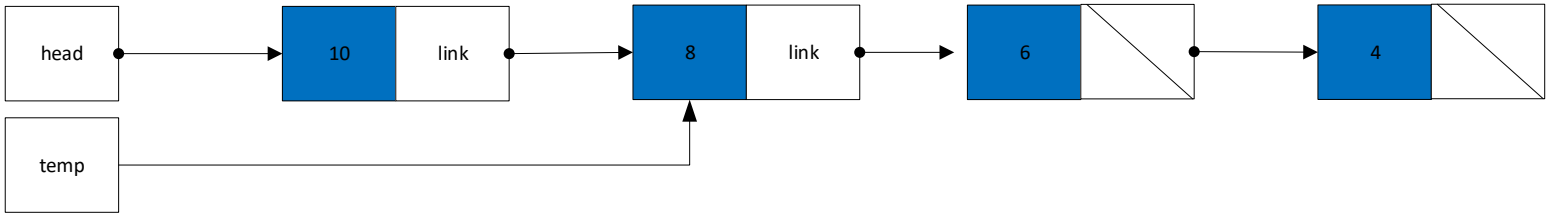
垃圾收集将节点返回
给系统堆



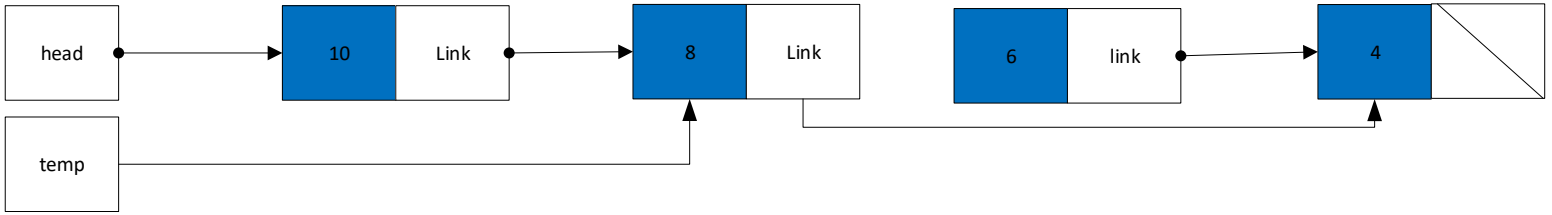
index == 2>1 且
temp.next!= None,
temp = temp.next
index - 1



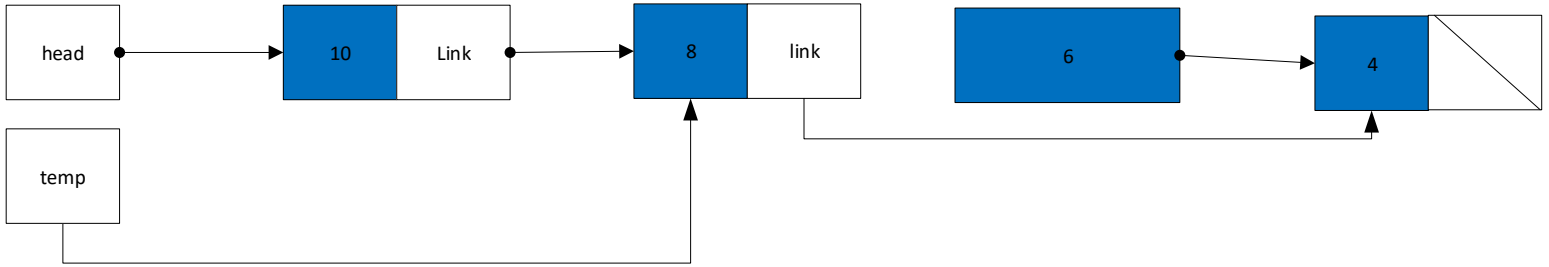
Index == 1
停止循环

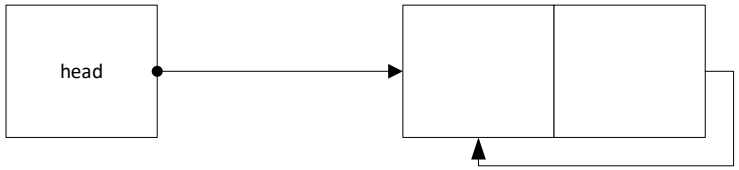


temp.next==
temp.next.next
删除节点

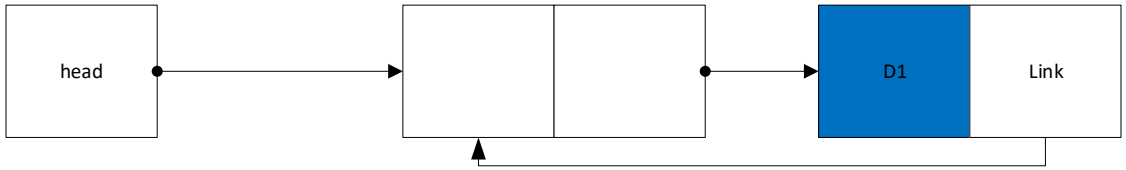


垃圾收集将节点返回
给系统堆

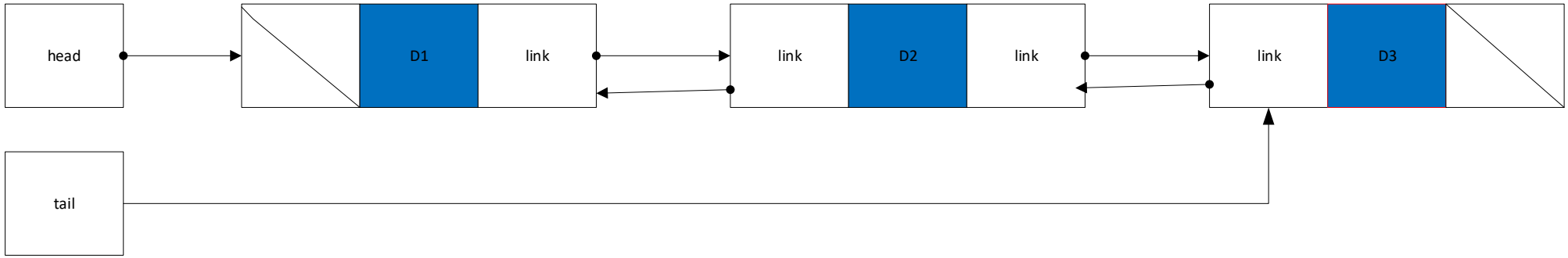




带有哑头节点的一个空的循环列表结构

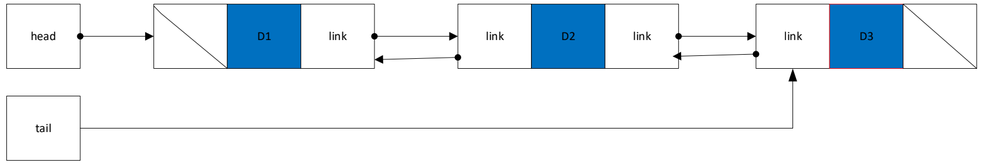


在插入了第一个节点后的循环列表结构

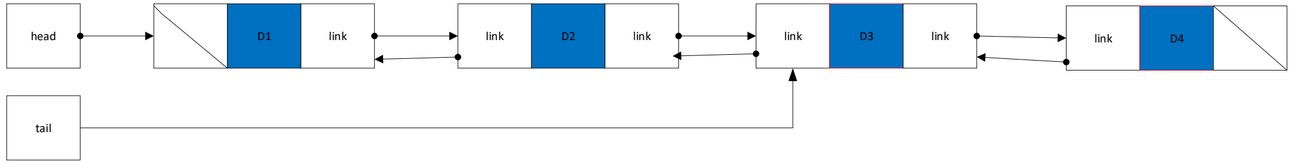


含有3个节点的双向链表的结构

插入之前的
双链表结构



Tail.next =
TwoWayNode(data,tail)



Tail=tail.next