class Node {

int data;

Node next;

public Node(int data) {

this.data = data;

this.next = null;

}

}

class SingularlyLinkedList {

Node head;

public SingularlyLinkedList() {

head = null;

}

public int size() {

Node current = head;

int count = 0;

while (current != null) {

count++;

current = current.next;

}

return count;

}

// يمكنك إضافة طرق أخرى مثل الإدخال أو الحذف هنا

}

**الشرح:**

* في هذه الشيفرة، يحتوي كائن Node على البيانات (data) والإشارة إلى العقدة التالية (next).
* فئة SingularlyLinkedList تحتوي على المتغير head الذي يشير إلى أول عقدة في القائمة.
* تقوم طريقة size() بالانتقال عبر القائمة المرتبطة ابتداءً من head وتقوم بحساب عدد العقد عن طريق زيادة المتغير count في كل مرة يتم فيها المرور بعقدة جديدة.
* عند الوصول إلى نهاية القائمة (current == null)، يتم إرجاع قيمة count التي تمثل عدد العقد في القائمة.

باستخدام هذا الأسلوب، يمكنك حساب حجم القائمة المرتبطة دون الحاجة للاحتفاظ بمتغير size في الكائن.